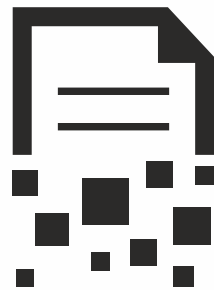


 **ACL 2013**

4 – 9 August | Sofia, Bulgaria

**51<sup>st</sup>**  
**ANNUAL MEETING OF THE  
ASSOCIATION FOR  
COMPUTATIONAL LINGUISTICS**



---

**Proceedings of  
the Conference**

---

**VOLUME 2: Short Papers**

ACL 2013

**51st Annual Meeting of the  
Association for Computational Linguistics**

**Proceedings of the Conference  
Volume 2: Short Papers**

August 4-9, 2013  
Sofia, Bulgaria

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

PLATINUM LEVEL SPONSOR



GOLD LEVEL SPONSORS



SILVER LEVEL SPONSORS



BRONZE LEVEL SPONSORS



SUPPORTER



BEST STUDENT PAPER AWARD



STUDENT VOLUNTEER



CONFERENCE BAG SPONSOR



CONFERENCE DINNER ENTERTAINMENT SPONSOR



LOCAL ORGANIZER



©2013 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

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

ISBN 978-1-937284-51-0 (Volume 2)

## Preface: General Chair

Welcome to the 51st Annual Meeting of the Association for Computational Linguistics in Sofia, Bulgaria! The first ACL meeting was held in Denver in 1963 under the name AMTCL. This makes ACL one of the longest running conferences in computer science. This year we received a record total number of 1286 submissions, which is a testament to the continued and growing importance of computational linguistics and natural language processing.

The success of an ACL conference is made possible by the dedication and hard work of many people. I thank all of them for volunteering their time and energy in service to our community.

Priscilla Rasmussen, the ACL Business Manager, and Graeme Hirst, the treasurer, did most of the groundwork in selecting Sofia as the conference site, went through several iterations of planning and shouldered a significant part of the organizational work for the conference. It was my first exposure to the logistics of organizing a large event and I was surprised at how much expertise and experience is necessary to make ACL a successful meeting.

Thanks to Svetla Koeva and her team for their work on local arrangements, including social activities (Radka Vlahova, Tsvetana Dimitrova, Svetlozara Lesseva), local sponsorship (Stoyan Mihov, Rositsa Dekova), conference handbook (Nikolay Genov, Hristina Kukova), web site (Tinko Tinchev, Emil Stoyanov, Georgi Iliev), local exhibits (Maria Todorova, Ekaterina Tarpomanova), internet, wifi and equipment (Martin Yalamov, Angel Genov, Borislav Rizov) and student volunteer management (Kalina Boncheva). Perhaps most importantly, Svetla was the liaison to the professional conference organizer AIM Group, a relationship that is crucial for the success of the conference. Doing the local arrangements is a fulltime job for an extended period of time. We are lucky that we have people in our community who are willing to provide this service without compensation.

The program co-chairs Pascale Fung and Massimo Poesio selected a strong set of papers for the main conference and invited three great keynote speakers, Harald Baayen, Chantal Prat and Lars Rasmussen. Putting together the program of the top conference in our field is a difficult job and I thank Pascale and Massimo for taking on this important responsibility.

Thanks are also due to the other key members of the ACL organizing committees: Aoife Cahill and Qun Liu (workshop co-chairs); Johan Bos and Keith Hall (tutorial co-chairs); Miriam Butt and Sarmad Hussain (demo co-chairs); Steven Bethard, Preslav Nakov and Feiyu Xu (faculty advisors to the student research workshop); Anik Dey, Eva Vecchi, Sebastian Krause and Ivelina Nikolova (co-chairs of the student research workshop); Leo Wanner (mentoring chair); and Anisava Miltenova, Ivan Derzhanski and Anna Korhonen (publicity co-chairs).

I am particularly indebted to Roberto Navigli, Jing-Shin Chang and Stefano Faralli for producing the proceedings of the conference, a bigger job than usual because of the large number of submissions and the resulting large number of acceptances.

The ACL conference and the ACL organization benefit greatly from the financial support of our sponsors. We thank the platinum level sponsor, Baidu; the three gold level sponsors; the three silver level sponsors; and six bronze level sponsors. Three other sponsors took advantage of more creative options to assist us: Facebook sponsored the Student Volunteers; IBM sponsored the Best Student Paper Award; and SDL sponsored the conference bags. We are grateful for the financial support from these organizations.

Finally, I would like to express my appreciation to the area chairs, workshop organizers, tutorial presenters and reviewers for their participation and contribution.

Of course, the ACL conference is primarily held for the people who attend the conference, including the

authors. I would like to thank all of you for your participation and wish you a productive and enjoyable meeting in Sofia!

ACL 2013 General Chair  
Hinrich Schuetze, University of Munich

## Preface: Programme Committee Co-Chairs

Welcome to the 2013 Conference of the Association for Computational Linguistics! Our community continues to grow, and this year's conference has set a new record for paper submissions. We received 1286 submissions, which is 12% more than the previous record; we are particularly pleased to see a striking increase in the number of short papers submitted - 624, which is 21.8% higher than the previous record set in 2011.

Another encouraging trend in recent years is the increasing number of aspects of language processing, and forms of language, of interest to our community. In order to reflect this greater diversity, this year's conference has a much larger number of tracks than previous conferences, 26. Consequently, many more area chairs and reviewers were recruited than in the past, thus involving an even greater subset of the community in the selection of the program. We feel this, too, is a very positive development. We thank the area chairs and reviewers for their hard work.

A key innovation introduced this year is the presentation at the conference of sixteen papers accepted by the new ACL journal, Transactions of the Association for Computational Linguistics (TACL). We have otherwise maintained most of the innovations introduced in recent years, including accepting papers accompanied by supplemental materials such as corpora or software.

Another new practice this year is the presence of an industrial keynote speaker in addition to the two traditional keynote speakers. We are delighted to have as invited speakers two scholars as distinguished as Prof. Harald Baayen of Tuebingen and Alberta and Prof. Chantel Prat from the University of Wisconsin. Prof. Baayen will talk about using eye-tracking to study the semantics of compounds, an issue of great interest for work on distributional semantics. Prof. Prat will talk about research studying language in bilinguals using methods from neuroscience. The industrial keynote speaker, Dr. Lars Rasmussen from Facebook, will talk about the new graph search algorithm recently announced by the company. Last, but not least, the recipient of this year's ACL Lifetime Achievement Award will give a plenary lecture during the final day of the conference.

The list of people to thank for their contribution to this year's program is very long. First of all we wish to thank the authors who submitted top quality work to the conference; we would not have such a strong program without them, nor without the hard work of area chairs and reviewers, who enabled us to make often very difficult choices and to provide valuable feedback to the authors. As usual, Rich Gerber and the START team gave us crucial help with an amazing speed. The general conference chair Hinrich Schuetze provided valuable guidance and kept the timetable ticking along. We thank the local arrangements committee headed by Svetla Koeva, who played a key role in finalizing the program. We also thank the publication chairs, Jing-Shin Chang and Roberto Navigli, and their collaborator Stefano Faralli, who together produced this volume; and Priscilla Rasmussen, Drago Radev and Graeme Hirst, who provided enormously useful guidance and support. Finally, we wish to thank previous program chairs, and in particular John Carroll, Stephen Clark, and Jian Su, for their insight on the process.

We hope you will be as pleased as we are with the result and that you'll enjoy the conference in Sofia this Summer.

ACL 2013 Program Co-Chairs

Pascale Fung, Hong-Kong University of Science and Technology

Massimo Poesio, University of Essex





# Organizing Committee

## **General Chair:**

Hinrich Schuetze, University of Munich

## **Program Co-Chairs:**

Pascale Fung, The Hong Kong University of Science and Technology  
Massimo Poesio, University of Essex

## **Local Chair:**

Svetla Koeva, Bulgarian Academy of Sciences

## **Workshop Co-Chairs:**

Aoife Cahill, Educational Testing Service  
Qun Liu, Dublin City University & Chinese Academy of Sciences

## **Tutorial Co-Chairs:**

Johan Bos, University of Groningen  
Keith Hall, Google

## **Demo Co-Chairs:**

Miriam Butt, University of Konstanz  
Sarmad Hussain, Al-Khawarizmi Institute of Computer Science

## **Publication Chairs:**

Roberto Navigli, Sapienza University of Rome (Chair)  
Jing-Shin Chang, National Chi Nan University (Co-Chair)  
Stefano Faralli, Sapienza University of Rome

## **Faculty Advisors (Student Research Workshop):**

Steven Bethard, University of Colorado Boulder & KU Leuven  
Preslav I. Nakov, Qatar Computing Research Institute  
Feiyu Xu, DFKI, German Research Center for Artificial Intelligence

## **Student Chairs (Student Research Workshop):**

Anik Dey, The Hong Kong University of Science & Technology  
Eva Vecchi, Università di Trento

Sebastian Krause, DFKI, German Research Center for Artificial Intelligence  
Ivelina Nikolova, Bulgarian Academy of Sciences

**Mentoring Chair:**

Leo Wanner, Universitat Pompeu Fabra

**Publicity Co-Chairs:**

Anisava Miltenova, Bulgarian Academy of Sciences  
Ivan Derzhanski, Bulgarian Academy of Sciences  
Anna Korhonen, University of Cambridge

**Business Manager:**

Priscilla Rasmussen, ACL

**Area Chairs:**

Frank Keller, University of Edinburgh  
Roger Levy, UC San Diego  
Amanda Stent, AT&T  
David Suendermann, DHBW, Stuttgart, Germany  
Andrew Kehler, UC San Diego  
Becky Passonneau, Columbia  
Hang Li, Huawei Technologies  
Nancy Ide, Vassar  
Piek Vossen, Freie Universitat Amsterdam  
Philipp Cimiano, University of Bielefeld  
Sabine Schulte im Walde, University of Stuttgart  
Dekang Lin, Google  
Chiori Hori, NICT, Japan  
Keh-Yih Su, Behavior Design Corporation  
Roland Kuhn, NRC  
Dekai Wu, HKUST  
Benjamin Snyder, University of Wisconsin-Madison  
Thamar Solorio, University of Texas-Dallas  
Ehud Reiter, University of Aberdeen  
Massimiliano Ciaramita, Google  
Ken Church, IBM  
Carlo Strapparava, FBK  
Tomaz Erjavec, Jožef Stefan Institute  
Adam Przepiorkowski, Polish Academy of Sciences  
Patrick Pantel, Microsoft  
Owen Rambow, Columbia  
Chris Dyer, CMU  
Jason Eisner, Johns Hopkins  
Jennifer Chu-Carroll, IBM  
Bernardo Magnini, FBK  
Lluis Marquez, Universitat Politecnica de Catalunya

Alessandro Moschitti, University of Trento  
Claire Cardie, Cornell  
Rada Mihalcea, University of North Texas  
Dilek Hakkani-Tur, Microsoft  
Walter Daelemans, University of Antwerp  
Dan Roth, University of Illinois Urbana Champaign  
Alex Koller, University of Potsdam  
Ani Nenkova, University of Pennsylvania  
Jamie Henderson, XRCE  
Sadao Kurohashi, University of Kyoto  
Yuji Matsumoto, Nara Institute of S&T  
Heng Ji, CUNY  
Marie-Francine Moens, KU Leuven  
Hwee Tou Ng, NU Singapore

### **Program Committee:**

Abend Omri, Abney Steven, Abu-Jbara Amjad, Agarwal Apoorv, Agirre Eneko, Aguado-de-Cea Guadalupe, Ahrenberg Lars, Akkaya Cem, Alfonseca Enrique, Alishahi Afra, Allauzen Alexander, Altun Yasemin, Androutsopoulos Ion, Araki Masahiro, Artiles Javier, Artzi Yoav, Asahara Masayuki, Asher Nicholas, Atserias Batalla Jordi, Attardi Giuseppe, Ayan Necip Fazil

Baker Collin, Baldridge Jason, Baldwin Timothy, Banchs Rafael E., Banea Carmen, Bangalore Srinivas, Baroni Marco, Barrault Loïc, Barreiro Anabela, Basili Roberto, Bateman John, Bechet Frederic, Beigman Klebanov Beata, Bel Núria, Benajiba Yassine, Bender Emily M., Bendersky Michael, Benotti Luciana, Bergler Sabine, Besacier Laurent, Bethard Steven, Bicknell Clinton, Biemann Chris, Bikel Dan, Birch Alexandra, Bisazza Arianna, Blache Philippe, Bloodgood Michael, Bod Rens, Boitet Christian, Bojar Ondrej, Bond Francis, Bontcheva Kalina, Bordino Ilaria, Bosch Sonja, Boschee Elizabeth, Botha Jan, Bouma Gosse, Boye Johan, Boyer Kristy, Bracewell David, Branco António, Brants Thorsten, Brew Chris, Briscoe Ted, Bu Fan, Buitelaar Paul, Bunesco Razvan, Busemann Stephan, Byrne Bill, Byron Donna

Cabrio Elena, Cahill Aoife, Cahill Lynne, Callison-Burch Chris, Calzolari Nicoletta, Campbell Nick, Cancedda Nicola, Cao Hailong, Caragea Cornelia, Carberry Sandra, Cardenosa Jesus, Cardie Claire, Carl Michael, Carpuat Marine, Carreras Xavier, Carroll John, Casacuberta Francisco, Caselli Tommaso, Cassidy Steve, Cassidy Taylor, Celikyilmaz Asli, Cerisara Christophe, Chambers Nate, Chang Jason, Chang Kai-Wei, Chang Ming-Wei, Chang Jing-Shin, Chelba Ciprian, Chen Wenliang, Chen Zheng, Chen Wenliang, Chen John, Chen Boxing, Chen David, Cheng Pu-Jen, Cherry Colin, Chiang David, Choi Yejin, Choi Key-Sun, Christodoulopoulos Christos, Chrupala Grzegorz, Chu-Carroll Jennifer, Clark Stephen, Clark Peter, Cohn Trevor, Collier Nigel, Conroy John, Cook Paul, Coppola Bonaventura, Corazza Anna, Core Mark, Costa-jussà Marta R., Cristea Dan, Croce Danilo, Culotta Aron, da Cunha Iria

Daelemans Walter, Dagan Ido, Daille Beatrice, Danescu-Niculescu-Mizil Cristian, Dang Hoa Trang, Danlos Laurence, Das Dipanjan, de Gispert Adrià, De La Clergerie Eric, de Marneffe Marie-Catherine, de Melo Gerard, Declerck Thierry, Delmonte Rodolfo, Demberg Vera, DeNero John, Deng Hongbo, Denis Pascal, Deoras Anoop, DeVault David, Di Eugenio Barbara, Di Fabrizio Giuseppe, Diab Mona, Diaz de Ilarraza Arantza, Diligenti Michelangelo, Dinarelli Marco, Dipper Stefanie, Do Quang, Downey Doug, Dragut Eduard, Dreyer Markus, Du Jinhua, Duh Kevin, Dymetman Marc

Eberle Kurt, Eguchi Koji, Eisele Andreas, Elhadad Michael, Erk Katrin, Esuli Andrea, Evert Stefan

Fader Anthony, Fan James, Fang Hui, Favre Benoit, Fazly Afsaneh, Federico Marcello, Feldman Anna, Feldman Naomi, Fellbaum Christiane, Feng Junlan, Fernandez Raquel, Filippova Katja, Finch Andrew, Fišer Darja, Fleck Margaret, Forcada Mikel, Foster Jennifer, Foster George, Frank Stella, Frank Stefan L., Frank Anette, Fraser Alexander

Gabrilovich Evgeniy, Gaizauskas Robert, Galley Michel, Gamon Michael, Ganitkevitch Juri, Gao Jianfeng, Gardent Claire, Garrido Guillermo, Gatt Albert, Gavrilidou Maria, Georgila Kallirroi, Gesmundo Andrea, Gildea Daniel, Gill Alastair, Gillenwater Jennifer, Gillick Daniel, Girju Roxana, Giuliano Claudio, Gliozzo Alfio, Goh Chooi-Ling, Goldberg Yoav, Goldwasser Dan, Goldwater Sharon, Gonzalo Julio, Grau Brigitte, Green Nancy, Greene Stephan, Grefenstette Gregory, Grishman Ralph, Guo Jiafeng, Gupta Rahul, Gurevych Iryna, Gustafson Joakim, Guthrie Louise, Gutiérrez Yoan

Habash Nizar, Hachey Ben, Haddow Barry, Hahn Udo, Hall David, Harabagiu Sanda, Hardmeier Christian, Hashimoto Chikara, Hayashi Katsuhiko, He Xiaodong, He Zhongjun, Heid Uli, Heinz Jeffrey, Henderson John, Hendrickx Iris, Hermjakob Ulf, Hirst Graeme, Hoang Hieu, Hockenmaier Julia, Hoffart Johannes, Hopkins Mark, Horak Ales, Hori Chiori, Hoste Veronique, Hovy Eduard, Hsieh Shu-Kai, Hsu Wen-Lian, Huang Xuanjing, Huang Minlie, Huang Liang, Huang Chu-Ren, Huang Xuanjing, Huang Liang, Huang Fei, Hwang Mei-Yuh

Iglesias Gonzalo, Ikbal Shajith, Ilisei Iustina, Inkpen Diana, Isabelle Pierre, Isahara Hitoshi, Ittycheriah Abe

Jaeger T. Florian, Jagarlamudi Jagadeesh, Jiampojarn Sittichai, Jiang Xing, Jiang Wenbin, Jiang Jing, Johansson Richard, Johnson Mark, Johnson Howard, Jurgens David

Kageura Kyo, Kan Min-Yen, Kanoulas Evangelos, Kanzaki Kyoko, Kawahara Daisuke, Keizer Simon, Kelleher John, Kempe Andre, Keshtkar Fazel, Khadivi Shahram, Kilgarriff Adam, King Tracy Holloway, Kit Chunyu, Knight Kevin, Koehn Philipp, Koeling Rob, Kolomiyets Oleksandr, Komatani Kazunori, Kondrak Grzegorz, Kong Fang, Kopp Stefan, Koppel Moshe, Kordoni Valia, Kozareva Zornitsa, Kozhevnikov Mikhail, Kraemer Emiel, Kremer Gerhard, Kudo Taku, Kuhlmann Marco, Kuhn Roland, Kumar Shankar, Kundu Gourab, Kurland Oren

Lam Wai, Lamar Michael, Lambert Patrik, Langlais Phillippe, Lapalme Guy, Lapata Mirella, Laws Florian, Leacock Claudia, Lee Yoong Keok, Lee Lin-shan, Lee Gary Geunbae, Lee Yoong Keok, Lee Sungjin, Lee John, Lefevre Fabrice, Lemon Oliver, Lenci Alessandro, Leong Ben, Leusch Gregor, Levenberg Abby, Levy Roger, Li Linlin, Li Fangtao, Li Yan, Li Haibo, Li Wenjie, Li Shoushan, Li Qi, Li Haizhou, Li Tao, Liao Shasha, Lin Dekang, Lin Ziheng, Lin Hui, Lin Ziheng, Lin Thomas, Litvak Marina, Liu Yang, Liu Bing, Liu Qun, Liu Ting, Liu Fei, Liu Zhiyuan, Liu Yiqun, Liu Chang, Liu Zhiyuan, Liu Jingjing, Liu Yiqun, Ljubešić Nikola, Lloret Elena, Lopez Adam, Lopez-Cozar Ramon, Louis Annie, Lu Wei, Lu Xiaofei, Lu Yue, Luca Dini, Luo Xiaoqiang, Lv Yajuan

Ma Yanjun, Macherey Wolfgang, Macherey Klaus, Madnani Nitin, Maegaard Bente, Magnini Bernardo, Maier Andreas, Manandhar Suresh, Marcu Daniel, Markantonatou Stella, Markert Katja, Marsi Erwin, Martin James H., Martinez David, Mason Rebecca, Matsubara Shigeki, Matsumoto

Yuji, Matsuzaki Takuya, Mauro Cettolo, Mauser Arne, May Jon, Mayfield James, Maynard Diana, McCarthy Diana, McClosky David, McCoy Kathy, McCrae John Philip, McNamee Paul, Meij Edgar, Mejova Yelena, Mellish Chris, Merlo Paola, Metze Florian, Metzler Donald, Meyers Adam, Mi Haitao, Mihalcea Rada, Miltsakaki Eleni, Minkov Einat, Mitchell Margaret, Miyao Yusuke, Mochihashi Daichi, Moens Marie-Francine, Mohammad Saif, Moilanen Karo, Monson Christian, Montes Manuel, Monz Christof, Moon Taesun, Moore Robert, Morante Roser, Morarescu Paul, Mueller Thomas, Munteanu Dragos, Murawaki Yugo, Muresan Smaranda, Myaeng Sung-Hyon, Mylonakis Markos

Nakagawa Tetsuji, Nakano Mikio, Nakazawa Toshiaki, Nakov Preslav, Naradowsky Jason, Naseem Tahira, Nastase Vivi, Navarro Borja, Navigli Roberto, Nazarenko Adeline, Nederhof Mark-Jan, Negri Matteo, Nenkova Ani, Neubig Graham, Neumann Guenter, Ng Vincent, Ngai Grace, Nguyen ThuyLinh, Nivre Joakim, Nowson Scott

Och Franz, Odijk Jan, Oflazer Kemal, Oh Jong-Hoon, Okazaki Naoaki, Oltramari Alessandro, Orasan Constantin, Osborne Miles, Osenova Petya, Ott Myle, Ovesdotter Alm Cecilia

Padó Sebastian, Palmer Martha, Palmer Alexis, Pang Bo, Pantel Patrick, Paraboni Ivandre, Pardo Thiago, Paris Cecile, Paroubek Patrick, Patwardhan Siddharth, Paul Michael, Paulik Matthias, Pearl Lisa, Pedersen Ted, Pedersen Bolette, Pedersen Ted, Peñas Anselmo, Penn Gerald, Perez-Rosas Veronica, Peters Wim, Petrov Slav, Petrovic Sasa, Piasecki Maciej, Pighin Daniele, Pinkal Manfred, Piperidis Stelios, Piskorski Jakub, Pitler Emily, Plank Barbara, Ponzetto Simone Paolo, Popescu Octavian, Popescu-Belis Andrei, Popović Maja, Potts Christopher, Pradhan Sameer, Prager John, Prasad Rashmi, Prószyński Gábor, Pulman Stephen, Punyakanok Vasin, Purver Matthew, Pustejovsky James

Qazvinian Vahed, Qian Xian, Qu Shaolin, Quarteroni Silvia, Quattoni Ariadna, Quirk Chris

Raaijmakers Stephan, Rahman Altaf, Rambow Owen, Rao Delip, Rappoport Ari, Ravi Sujith, Rayner Manny, Recasens Marta, Regneri Michaela, Reichart Roi, Reitter David, Resnik Philip, Riccardi Giuseppe, Riedel Sebastian, Riesa Jason, Rieser Verena, Riezler Stefan, Rigau German, Ringgaard Michael, Ritter Alan, Roark Brian, Rodriguez Horacio, Rohde Hannah, Rosenberg Andrew, Rosso Paolo, Rozovskaya Alla, Rus Vasile, Rusu Delia

Sagae Kenji, Sahakian Sam, Saint-Dizier Patrick, Samdani Rajhans, Sammons Mark, Sangal Rajeev, Saraclar Murat, Sarkar Anoop, Sassano Manabu, Satta Giorgio, Saurí Roser, Scaiano Martin, Schlangen David, Schmid Helmut, Schneider Nathan, Schulte im Walde Sabine, Schwenk Holger, Segond Frederique, Seki Yohei, Sekine Satoshi, Senellart Jean, Setiawan Hendra, Severyn Aliaksei, Shanker Vijay, Sharma Dipti, Sharoff Serge, Shi Shuming, Shi Xiaodong, Shi Shuming, Shutova Ekaterina, Si Xiance, Sidner Candace, Silva Mario J., Sima'an Khalil, Simard Michel, Skantze Gabriel, Small Kevin, Smith Noah A., Smith Nathaniel, Smrz Pavel, Smrz Pavel, Šnajder Jan, Snyder Benjamin, Søggaard Anders, Solorio Tamar, Somasundaran Swapna, Song Yangqiu, Spotovský Valentin, Sporleder Caroline, Sprugnoli Rachele, Srikumar Vivek, Stede Manfred, Steedman Mark, Steinberger Ralf, Stevenson Mark, Stone Matthew, Stoyanov Veselin, Strube Michael, Strzalkowski Tomek, Stymne Sara, Su Keh-Yih, Su Jian, Sun Ang, Surdeanu Mihai, Suzuki Hisami, Schwartz Roy, Szpakowicz Stan, Szpektor Idan

Täckström Oscar, Takamura Hiroya, Talukdar Partha, Tatu Marta, Taylor Sarah, Tenbrink Thora, Thater Stefan, Tiedemann Jörg, Tillmann Christoph, Titov Ivan, Toivonen Hannu, Tokunaga Takenobu,

Tonelli Sara, Toutanova Kristina, Tsarfaty Reut, Tsochantaridis Ioannis, Tsujii Jun'ichi, Tsukada Hajime, Tsuruoka Yoshimasa, Tufis Dan, Tur Gokhan, Turney Peter, Tymoshenko Kateryna

Uchimoto Kiyotaka, Udupa Raghavendra, Uryupina Olga, Utiyama Masao

Valitutti Alessandro, van den Bosch Antal, van der Plas Lonneke, Van Durme Benjamin, van Genabith Josef, Van Huyssteen Gerhard, van Noord Gertjan, Vandeghinste Vincent, Veale Tony, Velardi Paola, Verhagen Marc, Vetulani Zygmunt, Viethen Jette, Vieu Laure, Vilar David, Villavicencio Aline, Virpioja Sami, Voorhees Ellen, Vossen Piek, Vulić Ivan

Walker Marilyn, Wan Stephen, Wan Xiaojun, Wang Lu, Wang Chi, Wang Jun, Wang Haifeng, Wang Mengqiu, Wang Quan, Wang Wen, Ward Nigel, Washtell Justin, Watanabe Taro, Webber Bonnie, Wei Furu, Welty Chris, Wen Zhen, Wen Ji-Rong, Wen Zhen, Wicentowski Rich, Widdows Dominic, Wiebe Jan, Williams Jason, Wilson Theresa, Wintner Shuly, Wong Kam-Fai, Woodsend Kristian, Wooters Chuck, Wu Xianchao

Xiao Tong, Xiong Deyi, Xu Wei, Xu Jun, Xue Nianwen, Xue Xiaobing

Yan Rui, Yang Muyun, Yang Bishan, Yangarber Roman, Yano Tae, Yao Limin, Yates Alexander, Yatskar Mark, Yih Wen-tau, Yli-Jyrä Anssi, Yu Bei, Yvon François

Zabokrtsky Zdenek, Zanzotto Fabio Massimo, Zens Richard, Zettlemoyer Luke, Zeyrek Deniz, Zhang Yue, Zhang Min, Zhang Ruiqiang, Zhang Hao, Zhang Yue, Zhang Hui, Zhang Yi, Zhang Joy Ying, Zhanyi Liu, Zhao Hai, Zhao Tiejun, Zhao Jun, Zhao Shiqi, Zheng Jing, Zhou Guodong, Zhou Ming, Zhou Ke, Zhou Guodong, Zhou Ming, Zhou Guodong, Zhu Jingbo, Zhu Xiaodan, Zock Michael, Zukerman Ingrid, Zweigenbaum Pierre.

## Invited Talk

### When parsing makes things worse: An eye-tracking study of English compounds

Harald Baayen

Seminar für Sprachwissenschaft, Eberhard Karls University, Tuebingen

### Abstract

Compounds differ in the degree to which they are semantically compositional (compare, e.g., "carwash", "handbag", "beefcake" and "humbug"). Since even relatively transparent compounds such as "carwash" may leave the uninitiated reader with uncertainty about the intended meaning (soap for washing cars? a place where you can get your car washed?), an efficient way of retrieving the meaning of a compound is to use the compound's form as an access key for its meaning.

However, in psychology, the view has become popular that at the earliest stage of lexical processing in reading, a morpho-orthographic decomposition into morphemes would necessarily take place. Theorists ascribing to obligatory decomposition appear to have some hash coding scheme in mind, with the constituents providing entry points to a form of table look-up (e.g., Taft & Forster, 1976).

Leaving aside the question of whether such a hash coding scheme would be computationally efficient as well as the question how the putative morpho-orthographic representations would be learned, my presentation focuses on the details of lexical processing as revealed by an eye-tracking study of the reading of English compounds in sentences.

A careful examination of the eye-tracking record with generalized additive modeling (Wood, 2006), combined with computational modeling using naive discrimination learning (Baayen, Milin, Filipovic, Hendrix, & Marelli, 2011) revealed that how far the eye moved into the compound is co-determined by the compound's lexical distributional properties, including the cosine similarity of the compound and its head in document vector space (as measured with latent semantic analysis, Landauer & Dumais, 1997). This indicates that compound processing is initiated already while the eye is fixating on the preceding word, and that even before the eye has landed on the compound, processes discriminating the meaning of the compound from the meaning of its head have already come into play.

Once the eye lands on the compound, two very different reading signatures emerge, which critically depend on the letter trigrams spanning the morpheme boundary (e.g., "ndb" and "dba" in "handbag"). From a discrimination learning perspective, these boundary trigrams provide the crucial (and only) orthographic cues for the compound's (idiosyncratic) meaning. If the boundary trigrams are sufficiently strongly associated with the compound's meaning, and if the eye lands early enough in the word, a single fixation suffices. Within 240 ms (of which 80 ms involve planning the next saccade) the compound's meaning is discriminated well enough to proceed to the next word.

However, when the boundary trigrams are only weakly associated with the compound's meaning, multiple fixations become necessary. In this case, without the availability of the critical orthographic cues, the eye-tracking record bears witness to the cognitive system engaging not only bottom-up processes from form to meaning, but also top-down guessing processes that are informed by the a-priori probability of the head and the cosine similarities of the compound and its constituents in semantic vector space.

These results challenge theories positing obligatory decomposition with hash coding, as hash coding predicts insensitivity to semantic transparency, contrary to fact. Our results also challenge theories positing blind look-up based on compounds' orthographic forms. Although this might be computationally efficient, the eye can't help seeing parts of the whole. In summary, reality is much more complex, with deep pre-arrival parafoveal processing followed by either efficient discrimination driven by the boundary

trigrams (within 140 ms), or by an inefficient decompositional process (requiring an additional 200 ms) that seeks to make sense of the conjunction of head and modifier.

### **References**

Baayen, R. H., Kuperman, V., Shaoul, C., Milin, P., Kliegl, R. & Ramscar, M. (submitted), Decomposition makes things worse. A discrimination learning approach to the time course of understanding compounds in reading.

Baayen, R. H., Milin, P., Filipovic Durdjevic, D., Hendrix, P. & Marelli, M. (2011), An amorphous model for morphological processing in visual comprehension based on naive discriminative learning, *Psychological Review*, 118, 3, 438-481.

Landauer, T.K. & Dumais, S.T. (1997), A Solution to Plato's Problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge, *Psychological Review*, 104, 2, 211-240.

Taft, M. & Forster, K. I. (1976), Lexical Storage and Retrieval of Polymorphemic and Polysyllabic Words, *Journal of Verbal Learning and Verbal Behavior*, 15, 607-620.

Wood, S. N. (2006), *Generalized Additive Models*, Chapman & Hall/CRC, New York.



## **Invited Talk**

### **The Natural Language Interface of Graph Search**

**Lars Rasmussen**

Facebook Inc

### **Abstract**

The backbone of the Facebook social network service is an enormous graph representing hundreds of types of nodes and thousands of types of edges. Among these nodes are over 1 billion users and 250 billion photos. The edges connecting these nodes have exceeded 1 trillion and continue to grow at an incredible rate. Retrieving information from such a graph has been a formidable and exciting task. Now it is possible for you to find, in an aggregated manner, restaurants in a city that your friends have visited, or photos of people who have attended college with you, and explore many other nuanced connections between the nodes and edges in our graph given that such information is visible to you.

Graph Search Beta, launched early this year, is a personalized semantic search engine that allows users to express their intent in natural language. It seeks answers through the traversal of relevant graph edges and ranks results by various signals extracted from our data. You can find “tv shows liked by people who study linguistics“ by issuing this query verbatim and, for the entertainment value, compare the results with “tv shows liked by people who study computer science“. Our system is built to be robust to many varied inputs, such as grammatically incorrect user queries or traditional keyword searches. Our query suggestions are always constructed in natural language, expressing the precise intention interpreted by our system. This means users would know in advance whether the system has correctly understood their intent before selecting any suggestion. The system also assists users with auto-completions, demonstrating what kinds of queries it can understand.

The development of the natural language interface encountered an array of challenging problems. The grammar structure needed to incorporate semantic information in order to translate an unstructured query into a structured semantic function, and also use syntactic information to return grammatically meaningful suggestions. The system required not only the recognition of entities in a query, but also the resolution of entities to database entries based on proximity of the entity and user nodes. Semantic parsing aimed to rank potential semantics including those that may match the immediate purpose of the query along with other refinements of the original intent. The ambiguous nature of natural language led us to consider how to interpret certain queries in the most sensible way. The need for speed demanded state-of-the-art parsing algorithms tailored for our system. In this talk, I will introduce the audience to Graph Search Beta, share our experience in developing the technical components of the natural language interface, and bring up topics that may be of interesting research value to the NLP community.

## **Invited Talk**

### **Individual Differences in Language and Executive Processes: How the Brain Keeps Track of Variables**

**Chantel S. Prat**

University of Washington

#### **Abstract**

Language comprehension is a complex cognitive process which requires tracking and integrating multiple variables. Thus, it is not surprising that language abilities (e.g., reading comprehension) vary widely even in the college population, and that language and general cognitive abilities (e.g., working memory capacity) co-vary. Although it has been widely accepted that improvements in general cognitive abilities enable (or give rise to) increased linguistic skills, the fact that individuals who develop bilingually outperform monolinguals in tests of executive functioning provides evidence of a situation in which a particular language experience gives rise to improvements in general cognitive processes. In this talk, I will describe two converging lines of research investigating individual differences in working memory capacity and reading ability in monolinguals and improved executive functioning in bilinguals. Results from these investigations suggest that the functioning of the fronto-striatal loops can explain the relation between language and non-linguistic executive functioning in both populations. I then discuss evidence suggesting that this system may function to track and route “variables” into prefrontal control structures.

## Table of Contents

|   |    |
|---|----|
| <i>Translating Dialectal Arabic to English</i>  |    |
| Hassan Sajjad, Kareem Darwish and Yonatan Belinkov .....  | 1  |
| <i>Exact Maximum Inference for the Fertility Hidden Markov Model</i>  |    |
| Chris Quirk .....   | 7  |
| <i>A Tale about PRO and Monsters</i>  |    |
| Preslav Nakov, Francisco Guzmán and Stephan Vogel .....   | 12 |
| <i>Supervised Model Learning with Feature Grouping based on a Discrete Constraint</i>   |    |
| Jun Suzuki and Masaaki Nagata .....   | 18 |
| <i>Exploiting Topic based Twitter Sentiment for Stock Prediction</i>  |    |
| Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li and Xiaotie Deng .....  | 24 |
| <i>Learning Entity Representation for Entity Disambiguation</i>   |    |
| Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang and Houfeng Wang .....   | 30 |
| <i>Natural Language Models for Predicting Programming Comments</i>  |    |
| Dana Movshovitz-Attias and William W. Cohen .....   | 35 |
| <i>Paraphrasing Adaptation for Web Search Ranking</i>   |    |
| Chenguang Wang, Nan Duan, Ming Zhou and Ming Zhang .....  | 41 |
| <i>Semantic Parsing as Machine Translation</i>  |    |
| Jacob Andreas, Andreas Vlachos and Stephen Clark .....  | 47 |
| <i>A relatedness benchmark to test the role of determiners in compositional distributional semantics</i>  |    |
| Raffaella Bernardi, Georgiana Dinu, Marco Marelli and Marco Baroni .....  | 53 |
| <i>An Empirical Study on Uncertainty Identification in Social Media Context</i>   |    |
| Zhongyu Wei, Junwen Chen, Wei Gao, Binyang Li, Lanjun Zhou, Yulan He and Kam-Fai Wong .....   | 58 |
| <i>PARMA: A Predicate Argument Aligner</i>  |    |
| Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu and Xuchen Yao .....  | 63 |
| <i>Aggregated Word Pair Features for Implicit Discourse Relation Disambiguation</i>   |    |
| Or Biran and Kathleen McKeown .....   | 69 |
| <i>Implicatures and Nested Beliefs in Approximate Decentralized-POMDPs</i>  |    |
| Adam Vogel, Christopher Potts and Dan Jurafsky .....  | 74 |
| <i>Domain-Specific Coreference Resolution with Lexicalized Features</i>   |    |
| Nathan Gilbert and Ellen Riloff .....   | 81 |
| <i>Learning to Order Natural Language Texts</i>   |    |
| Jiwei Tan, Xiaojun Wan and Jianguo Xiao .....   | 87 |
| <i>Universal Dependency Annotation for Multilingual Parsing</i>   |    |
| Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló and Jungmee Lee ..... | 92 |

|   |     |
|---|-----|
| <i>An Empirical Examination of Challenges in Chinese Parsing</i><br>Jonathan K. Kummerfeld, Daniel Tse, James R. Curran and Dan Klein .....   | 98  |
| <i>Joint Inference for Heterogeneous Dependency Parsing</i><br>Guangyou Zhou and Jun Zhao .....   | 104 |
| <i>Easy-First POS Tagging and Dependency Parsing with Beam Search</i><br>Ji Ma, Jingbo Zhu, Tong Xiao and Nan Yang .....  | 110 |
| <i>Arguments and Modifiers from the Learner’s Perspective</i><br>Leon Bergen, Edward Gibson and Timothy J. O’Donnell .....  | 115 |
| <i>Benefactive/Malefactive Event and Writer Attitude Annotation</i><br>Lingjia Deng, Yoonjung Choi and Janyce Wiebe .....   | 120 |
| <i>GuiTAR-based Pronominal Anaphora Resolution in Bengali</i><br>Apurbalal Senapati and Utpal Garain .....  | 126 |
| <i>A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art</i><br>Peter A. Rankel, John M. Conroy, Hoa Trang Dang and Ani Nenkova .....                            | 131 |
| <i>On the Predictability of Human Assessment: when Matrix Completion Meets NLP Evaluation</i><br>Guillaume Wisniewski .....   | 137 |
| <i>Automated Pyramid Scoring of Summaries using Distributional Semantics</i><br>Rebecca J. Passonneau, Emily Chen, Weiwei Guo and Dolores Perin .....   | 143 |
| <i>Are Semantically Coherent Topic Models Useful for Ad Hoc Information Retrieval?</i><br>Romain Deveaud, Eric SanJuan and Patrice Bellot .....   | 148 |
| <i>Post-Retrieval Clustering Using Third-Order Similarity Measures</i><br>Jose G. Moreno, Gaël Dias and Guillaume Cleuziou .....  | 153 |
| <i>Automatic Coupling of Answer Extraction and Information Retrieval</i><br>Xuchen Yao, Benjamin Van Durme and Peter Clark .....  | 159 |
| <i>An improved MDL-based compression algorithm for unsupervised word segmentation</i><br>Ruey-Cheng Chen .....  | 166 |
| <i>Co-regularizing character-based and word-based models for semi-supervised Chinese word segmentation</i><br>Xiaodong Zeng, Derek F. Wong, Lidia S. Chao and Isabel Trancoso .....                     | 171 |
| <i>Improving Chinese Word Segmentation on Micro-blog Using Rich Punctuations</i><br>Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang and Ni Sun .....  | 177 |
| <i>Accurate Word Segmentation using Transliteration and Language Model Projection</i><br>Masato Hagiwara and Satoshi Sekine .....   | 183 |
| <i>Broadcast News Story Segmentation Using Manifold Learning on Latent Topic Distributions</i><br>Xiaoming Lu, Lei Xie, Cheung-Chi Leung, Bin Ma and Haizhou Li .....                                   | 190 |
| <i>Is word-to-phone mapping better than phone-phone mapping for handling English words?</i><br>Naresh Kumar Elluru, Anandaswarup Vadapalli, Raghavendra Elluru, Hema Murthy and Kishore Prahallad ..... | 196 |

|  |     |
|--|-----|
| <i>Enriching Entity Translation Discovery using Selective Temporality</i><br>Gae-won You, Young-rok Cha, Jinhan Kim and Seung-won Hwang . . . . .  | 201 |
| <i>Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling</i><br>Heike Adel, Ngoc Thang Vu and Tanja Schultz . . . . .                     | 206 |
| <i>Latent Semantic Matching: Application to Cross-language Text Categorization without Alignment Information</i><br>Tsutomu Hirao, Tomoharu Iwata and Masaaki Nagata . . . . .                 | 212 |
| <i>TopicSpam: a Topic-Model based approach for spam detection</i><br>Jiwei Li, Claire Cardie and Sujian Li . . . . .   | 217 |
| <i>Semantic Neighborhoods as Hypergraphs</i><br>Chris Quirk and Pallavi Choudhury . . . . .  | 222 |
| <i>Unsupervised joke generation from big data</i><br>Saša Petrović and David Matthews . . . . .  | 228 |
| <i>Modeling of term-distance and term-occurrence information for improving n-gram language model performance</i><br>Tze Yuang Chong, Rafael E. Banchs, Eng Siong Chng and Haizhou Li . . . . . | 233 |
| <i>Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners</i><br>Keisuke Sakaguchi, Yuki Arase and Mamoru Komachi . . . . .  | 238 |
| <i>"Let Everything Turn Well in Your Wife": Generation of Adult Humor Using Lexical Constraints</i><br>Alessandro Valitutti, Hannu Toivonen, Antoine Doucet and Jukka M. Toivanen . . . . .    | 243 |
| <i>Random Walk Factoid Annotation for Collective Discourse</i><br>Ben King, Rahul Jha, Dragomir Radev and Robert Mankoff . . . . .   | 249 |
| <i>Identifying English and Hungarian Light Verb Constructions: A Contrastive Approach</i><br>Veronika Vincze, István Nagy T. and Richárd Farkas . . . . .                                      | 255 |
| <i>English-to-Russian MT evaluation campaign</i><br>Pavel Braslavski, Alexander Beloborodov, Maxim Khalilov and Serge Sharoff . . . . .  | 262 |
| <i>IndoNet: A Multilingual Lexical Knowledge Network for Indian Languages</i><br>Brijesh Bhatt, Lahari Poddar and Pushpak Bhattacharyya . . . . .  | 268 |
| <i>Building Japanese Textual Entailment Specialized Data Sets for Inference of Basic Sentence Relations</i><br>Kimi Kaneko, Yusuke Miyao and Daisuke Bekki . . . . .                           | 273 |
| <i>Building Comparable Corpora Based on Bilingual LDA Model</i><br>Zede Zhu, Miao Li, Lei Chen and Zhenxin Yang . . . . .  | 278 |
| <i>Using Lexical Expansion to Learn Inference Rules from Sparse Data</i><br>Oren Melamud, Ido Dagan, Jacob Goldberger and Idan Szpektor . . . . .  | 283 |
| <i>Mining Equivalent Relations from Linked Data</i><br>Ziqi Zhang, Anna Lisa Gentile, Isabelle Augenstein, Eva Blomqvist and Fabio Ciravegna . . . . .   | 289 |
| <i>Context-Dependent Multilingual Lexical Lookup for Under-Resourced Languages</i><br>Lian Tze Lim, Lay-Ki Soon, Tek Yong Lim, Enya Kong Tang and Bali Ranaivo-Malançon . . . . .              | 294 |

|  |     |
|--|-----|
| <i>Sorani Kurdish versus Kurmanji Kurdish: An Empirical Comparison</i><br>Kyumars Sheykh Esmaili and Shahin Salavati .....   | 300 |
| <i>Enhanced and Portable Dependency Projection Algorithms Using Interlinear Glossed Text</i><br>Ryan Georgi, Fei Xia and William D. Lewis .....                              | 306 |
| <i>Cross-lingual Projections between Languages from Different Families</i><br>Mo Yu, Tiejun Zhao, Yalong Bai, Hao Tian and Dianhai Yu .....                                  | 312 |
| <i>Using Context Vectors in Improving a Machine Translation System with Bridge Language</i><br>Samira Tofighi Zahabi, Somayeh Bakhshaei and Shahram Khadivi .....            | 318 |
| <i>Task Alternation in Parallel Sentence Retrieval for Twitter Translation</i><br>Felix Hieber, Laura Jehl and Stefan Riezler .....  | 323 |
| <i>Sign Language Lexical Recognition With Propositional Dynamic Logic</i><br>Arturo Curiel and Christophe Collet .....   | 328 |
| <i>Stacking for Statistical Machine Translation</i><br>Majid Razmara and Anoop Sarkar .....  | 334 |
| <i>Bilingual Data Cleaning for SMT using Graph-based Random Walk</i><br>Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li and Ming Zhou .....                                       | 340 |
| <i>Automatically Predicting Sentence Translation Difficulty</i><br>Abhijit Mishra, Pushpak Bhattacharyya and Michael Carl .....  | 346 |
| <i>Learning to Prune: Context-Sensitive Pruning for Syntactic MT</i><br>Wenduan Xu, Yue Zhang, Philip Williams and Philipp Koehn .....                                       | 352 |
| <i>A Novel Graph-based Compact Representation of Word Alignment</i><br>Qun Liu, Zhaopeng Tu and Shouxun Lin .....  | 358 |
| <i>Stem Translation with Affix-Based Rule Selection for Agglutinative Languages</i><br>Zhiyang Wang, Yajuan Lü, Meng Sun and Qun Liu .....                                   | 364 |
| <i>A Novel Translation Framework Based on Rhetorical Structure Theory</i><br>Mei Tu, Yu Zhou and Chengqing Zong .....  | 370 |
| <i>Improving machine translation by training against an automatic semantic frame based evaluation metric</i><br>Chi-kiu Lo, Karteek Addanki, Markus Saers and Dekai Wu ..... | 375 |
| <i>Bilingual Lexical Cohesion Trigger Model for Document-Level Machine Translation</i><br>Guosheng Ben, Deyi Xiong, Zhiyang Teng, Yajuan Lü and Qun Liu .....                | 382 |
| <i>Generalized Reordering Rules for Improved SMT</i><br>Fei Huang and Cezar Pendus .....   | 387 |
| <i>A Tightly-coupled Unsupervised Clustering and Bilingual Alignment Model for Transliteration</i><br>Tingting Li, Tiejun Zhao, Andrew Finch and Chunyue Zhang .....         | 393 |
| <i>Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?</i><br>Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang and Philipp Koehn .....         | 399 |
| <i>Learning Non-linear Features for Machine Translation Using Gradient Boosting Machines</i><br>Kristina Toutanova and Byung-Gyu Ahn .....                                   | 406 |

|  |     |
|--|-----|
| <i>Language Independent Connectivity Strength Features for Phrase Pivot Statistical Machine Translation</i><br>Ahmed El Kholly, Nizar Habash, Gregor Leusch, Evgeny Matusov and Hassan Sawaf ..... | 412 |
| <i>Semantic Roles for String to Tree Machine Translation</i><br>Marzieh Bazrafshan and Daniel Gildea .....   | 419 |
| <i>Minimum Bayes Risk based Answer Re-ranking for Question Answering</i><br>Nan Duan .....   | 424 |
| <i>Question Classification Transfer</i><br>Anne-Laure Ligozat .....  | 429 |
| <i>Latent Semantic Tensor Indexing for Community-based Question Answering</i><br>Xipeng Qiu, Le Tian and Xuanjing Huang .....  | 434 |
| <i>Measuring semantic content in distributional vectors</i><br>Aurélie Herbelot and Mohan Ganesalingam .....   | 440 |
| <i>Modeling Human Inference Process for Textual Entailment Recognition</i><br>Hen-Hsen Huang, Kai-Chun Chang and Hsin-Hsi Chen .....   | 446 |
| <i>Recognizing Partial Textual Entailment</i><br>Omer Levy, Torsten Zesch, Ido Dagan and Iryna Gurevych .....  | 451 |
| <i>Sentence Level Dialect Identification in Arabic</i><br>Heba Elfardy and Mona Diab .....   | 456 |
| <i>Leveraging Domain-Independent Information in Semantic Parsing</i><br>Dan Goldwasser and Dan Roth .....  | 462 |
| <i>A Structured Distributional Semantic Model for Event Co-reference</i><br>Kartik Goyal, Sujay Kumar Jauhar, Huiying Li, Mrinmaya Sachan, Shashank Srivastava and Eduard Hovy .....               | 467 |
| <i>Text Classification from Positive and Unlabeled Data using Misclassified Data Correction</i><br>Fumiyo Fukumoto, Yoshimi Suzuki and Suguru Matsuyoshi .....                                     | 474 |
| <i>Character-to-Character Sentiment Analysis in Shakespeare’s Plays</i><br>Eric T. Nalisnick and Henry S. Baird .....  | 479 |
| <i>A Novel Classifier Based on Quantum Computation</i><br>Ding Liu, Xiaofang Yang and Minghu Jiang .....   | 484 |
| <i>Re-embedding words</i><br>Igor Labutov and Hod Lipson .....   | 489 |
| <i>LABR: A Large Scale Arabic Book Reviews Dataset</i><br>Mohamed Aly and Amir Atiya .....   | 494 |
| <i>Generating Recommendation Dialogs by Extracting Information from User Reviews</i><br>Kevin Reschke, Adam Vogel and Dan Jurafsky .....   | 499 |
| <i>Exploring Sentiment in Social Media: Bootstrapping Subjectivity Clues from Multilingual Twitter Streams</i><br>Svitlana Volkova, Theresa Wilson and David Yarowsky .....                        | 505 |

|   |     |
|---|-----|
| <i>Joint Modeling of News Reader’s and Comment Writer’s Emotions</i><br>Huanhuan Liu, Shoushan Li, Guodong Zhou, Chu-ren Huang and Peifeng Li .....                                     | 511 |
| <i>An annotated corpus of quoted opinions in news articles</i><br>Tim O’Keefe, James R. Curran, Peter Ashwell and Irena Koprinska .....   | 516 |
| <i>Dual Training and Dual Prediction for Polarity Classification</i><br>Rui Xia, Tao Wang, Xuelei Hu, Shoushan Li and Chengqing Zong .....  | 521 |
| <i>Co-Regression for Cross-Language Review Rating Prediction</i><br>Xiaojun Wan .....   | 526 |
| <i>Extracting Definitions and Hypernym Relations relying on Syntactic Dependencies and Support Vector Machines</i><br>Guido Boella and Luigi Di Caro .....                              | 532 |
| <i>Neighbors Help: Bilingual Unsupervised WSD Using Context</i><br>Sudha Bhingardive, Samiulla Shaikh and Pushpak Bhattacharyya .....   | 538 |
| <i>Reducing Annotation Effort for Quality Estimation via Active Learning</i><br>Daniel Beck, Lucia Specia and Trevor Cohn .....   | 543 |
| <i>Reranking with Linguistic and Semantic Features for Arabic Optical Character Recognition</i><br>Nadi Tomeh, Nizar Habash, Ryan Roth, Noura Farra, Pradeep Dasigi and Mona Diab ..... | 549 |
| <i>Evolutionary Hierarchical Dirichlet Process for Timeline Summarization</i><br>Jiwei Li and Sujian Li .....   | 556 |
| <i>Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts</i><br>Gerasimos Lampouras and Ion Androutsopoulos .....                                | 561 |
| <i>Sequential Summarization: A New Application for Timely Updated Twitter Trending Topics</i><br>Dehong Gao, Wenjie Li and Renxian Zhang .....  | 567 |
| <i>A System for Summarizing Scientific Topics Starting from Keywords</i><br>Rahul Jha, Amjad Abu-Jbara and Dragomir Radev .....   | 572 |
| <i>A Unified Morpho-Syntactic Scheme of Stanford Dependencies</i><br>Reut Tsarfaty .....  | 578 |
| <i>Dependency Parser Adaptation with Subtrees from Auto-Parsed Target Domain Data</i><br>Xuezhe Ma and Fei Xia .....  | 585 |
| <i>Iterative Transformation of Annotation Guidelines for Constituency Parsing</i><br>Xiang Li, Wenbin Jiang, Yajuan Lü and Qun Liu .....  | 591 |
| <i>Nonparametric Bayesian Inference and Efficient Parsing for Tree-adjointing Grammars</i><br>Elif Yamangil and Stuart M. Shieber .....   | 597 |
| <i>Using CCG categories to improve Hindi dependency parsing</i><br>Bharat Ram Ambati, Tejaswini Deoskar and Mark Steedman .....   | 604 |
| <i>The Effect of Higher-Order Dependency Features in Discriminative Phrase-Structure Parsing</i><br>Greg Coppola and Mark Steedman .....  | 610 |



|  |     |
|--|-----|
| <i>Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers</i><br>Andre Martins, Miguel Almeida and Noah A. Smith .....  | 617 |
| <i>A Lattice-based Framework for Joint Chinese Word Segmentation, POS Tagging and Parsing</i><br>Zhiguo Wang, Chengqing Zong and Nianwen Xue .....   | 623 |
| <i>Efficient Implementation of Beam-Search Incremental Parsers</i><br>Yoav Goldberg, Kai Zhao and Liang Huang .....  | 628 |
| <i>Simpler unsupervised POS tagging with bilingual projections</i><br>Long Duong, Paul Cook, Steven Bird and Pavel Pecina .....  | 634 |
| <i>Part-of-speech tagging with antagonistic adversaries</i><br>Anders Søgaard .....  | 640 |
| <i>Temporal Signals Help Label Temporal Relations</i><br>Leon Derczynski and Robert Gaizauskas .....   | 645 |
| <i>Diverse Keyword Extraction from Conversations</i><br>Maryam Habibi and Andrei Popescu-Belis .....   | 651 |
| <i>Understanding Tables in Context Using Standard NLP Toolkits</i><br>Vidhya Govindaraju, Ce Zhang and Christopher Ré .....  | 658 |
| <i>Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction</i><br>Wei Xu, Raphael Hoffmann, Le Zhao and Ralph Grishman .....  | 665 |
| <i>Joint Apposition Extraction with Syntactic and Semantic Constraints</i><br>Will Radford and James R. Curran .....   | 671 |
| <i>Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation</i><br>Kevin Duh, Graham Neubig, Katsuhito Sudoh and Hajime Tsukada .....                                | 678 |
| <i>Mapping Source to Target Strings without Alignment by Analogical Learning: A Case Study with Transliteration</i><br>Phillippe Langlais .....  | 684 |
| <i>Scalable Modified Kneser-Ney Language Model Estimation</i><br>Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark and Philipp Koehn .....   | 690 |
| <i>Incremental Topic-Based Translation Model Adaptation for Conversational Spoken Language Translation</i><br>Sanjika Hewavitharana, Dennis Mehay, Sankaranarayanan Ananthakrishnan and Prem Natarajan | 697 |
| <i>A Lightweight and High Performance Monolingual Word Aligner</i><br>Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch and Peter Clark .....   | 702 |
| <i>A Learner Corpus-based Approach to Verb Suggestion for ESL</i><br>Yu Sawai, Mamoru Komachi and Yuji Matsumoto .....   | 708 |
| <i>Learning Semantic Textual Similarity with Structural Representations</i><br>Aliaksei Severyn, Massimo Nicosia and Alessandro Moschitti .....  | 714 |
| <i>Typesetting for Improved Readability using Lexical and Syntactic Information</i><br>Ahmed Salama, Kemal Oflazer and Susan Hagan .....   | 719 |

|  |     |
|--|-----|
| <i>Annotation of regular polysemy and underspecification</i>   |     |
| Héctor Martínez Alonso, Bolette Sandford Pedersen and Núria Bel .....                                  | 725 |
| <i>Derivational Smoothing for Syntactic Distributional Semantics</i>                                   |     |
| Sebastian Padó, Jan Šnajder and Britta Zeller .....  | 731 |
| <i>Diathesis alternation approximation for verb clustering</i>   |     |
| Lin Sun, Diana McCarthy and Anna Korhonen .....  | 736 |
| <i>Outsourcing FrameNet to the Crowd</i>   |     |
| Marco Fossati, Claudio Giuliano and Sara Tonelli .....   | 742 |
| <i>Smatch: an Evaluation Metric for Semantic Feature Structures</i>                                    |     |
| Shu Cai and Kevin Knight .....   | 748 |
| <i>Variable Bit Quantisation for LSH</i>   |     |
| Sean Moran, Victor Lavrenko and Miles Osborne .....  | 753 |
| <i>Context Vector Disambiguation for Bilingual Lexicon Extraction from Comparable Corpora</i>          |     |
| Dhouha Bouamor, Nasredine Semmar and Pierre Zweigenbaum .....  | 759 |
| <i>The Effects of Lexical Resource Quality on Preference Violation Detection</i>                       |     |
| Jesse Dunietz, Lori Levin and Jaime Carbonell .....  | 765 |
| <i>Exploiting Qualitative Information from Automatic Word Alignment for Cross-lingual NLP Tasks</i>    |     |
| José G.C. de Souza, Miquel Esplà-Gomis, Marco Turchi and Matteo Negri .....                            | 771 |
| <i>An Information Theoretic Approach to Bilingual Word Clustering</i>                                  |     |
| Manaal Faruqui and Chris Dyer .....  | 777 |
| <i>Building and Evaluating a Distributional Memory for Croatian</i>                                    |     |
| Jan Šnajder, Sebastian Padó and Željko Agić .....  | 784 |
| <i>Generalizing Image Captions for Image-Text Parallel Corpus</i>                                      |     |
| Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara Berg and Yejin Choi .....                   | 790 |
| <i>Recognizing Identical Events with Graph Kernels</i>   |     |
| Goran Glavaš and Jan Šnajder .....   | 797 |
| <i>Automatic Term Ambiguity Detection</i>  |     |
| Tyler Baldwin, Yunyao Li, Bogdan Alexe and Ioana R. Stanoi .....                                       | 804 |
| <i>Towards Accurate Distant Supervision for Relational Facts Extraction</i>                            |     |
| Xingxing Zhang, Jianwen Zhang, Junyu Zeng, Jun Yan, Zheng Chen and Zhifang Sui .....                   | 810 |
| <i>Extra-Linguistic Constraints on Stance Recognition in Ideological Debates</i>                       |     |
| Kazi Saidul Hasan and Vincent Ng .....   | 816 |
| <i>Are School-of-thought Words Characterizable?</i>  |     |
| Xiaorui Jiang, Xiaoping Sun and Hai Zhuge .....  | 822 |
| <i>Identifying Opinion Subgroups in Arabic Online Discussions</i>                                      |     |
| Amjad Abu-Jbara, Ben King, Mona Diab and Dragomir Radev .....  | 829 |
| <i>Extracting Events with Informal Temporal References in Personal Histories in Online Communities</i> |     |
| Miaomiao Wen, Zeyu Zheng, Hyeju Jang, Guang Xiang and Carolyn Penstein Rosé .....                      | 836 |

|   |     |
|---|-----|
| <i>Multimodal DBN for Predicting High-Quality Answers in cQA portals</i><br>Haifeng Hu, Bingquan Liu, Baoxun Wang, Ming Liu and Xiaolong Wang .....         | 843 |
| <i>Bi-directional Inter-dependencies of Subjective Expressions and Targets and their Value for a Joint Model</i><br>Roman Klinger and Philipp Cimiano ..... | 848 |
| <i>Identifying Sentiment Words Using an Optimization-based Model without Seed Words</i><br>Hongliang Yu, Zhi-Hong Deng and Shiyngxue Li .....               | 855 |
| <i>Detecting Turnarounds in Sentiment Analysis: Thwarting</i><br>Ankit Ramteke, Akshat Malu, Pushpak Bhattacharyya and J. Saketha Nath .....                | 860 |
| <i>Explicit and Implicit Syntactic Features for Text Classification</i><br>Matt Post and Shane Bergsma .....  | 866 |
| <i>Does Korean defeat phonotactic word segmentation?</i><br>Robert Daland and Kie Zuraw .....   | 873 |
| <i>Word surprisal predicts N400 amplitude during reading</i><br>Stefan L. Frank, Leun J. Otten, Giulia Galli and Gabriella Vigliocco .....                  | 878 |
| <i>Computerized Analysis of a Verbal Fluency Test</i><br>James O. Ryan, Serguei Pakhomov, Susan Marino, Charles Bernick and Sarah Banks .....               | 884 |
| <i>A New Set of Norms for Semantic Relatedness Measures</i><br>Sean Szumlanski, Fernando Gomez and Valerie K. Sims .....                                    | 890 |



# Conference Program

**Monday August 5, 2013**

**(7:30 - 17:00) Registration**

**(9:00 - 9:30) Opening session**

**(9:30) Invited Talk 1: Harald Baayen**

**(10:30) Coffee Break**

**Oral Presentations**

**(12:15) Lunch break**

**(16:15) Coffee Break**

**(16:45 - 18:05) SP 4a**

- 16:45 *Translating Dialectal Arabic to English*  
Hassan Sajjad, Kareem Darwish and Yonatan Belinkov
- 17:05 *Exact Maximum Inference for the Fertility Hidden Markov Model*  
Chris Quirk
- 17:25 *A Tale about PRO and Monsters*  
Preslav Nakov, Francisco Guzmán and Stephan Vogel
- 17:45 *Supervised Model Learning with Feature Grouping based on a Discrete Constraint*  
Jun Suzuki and Masaaki Nagata

**Monday August 5, 2013 (continued)**

**(16:45 - 18:05) SP 4b**

- 16:45 *Exploiting Topic based Twitter Sentiment for Stock Prediction*  
Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li and Xiaotie Deng
- 17:05 *Learning Entity Representation for Entity Disambiguation*  
Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang and Houfeng Wang
- 17:25 *Natural Language Models for Predicting Programming Comments*  
Dana Movshovitz-Attias and William W. Cohen
- 17:45 *Paraphrasing Adaptation for Web Search Ranking*  
Chenguang Wang, Nan Duan, Ming Zhou and Ming Zhang

**(16:45 - 18:05) SP 4c**

- 16:45 *Semantic Parsing as Machine Translation*  
Jacob Andreas, Andreas Vlachos and Stephen Clark
- 17:05 *A relatedness benchmark to test the role of determiners in compositional distributional semantics*  
Raffaella Bernardi, Georgiana Dinu, Marco Marelli and Marco Baroni
- 17:25 *An Empirical Study on Uncertainty Identification in Social Media Context*  
Zhongyu Wei, Junwen Chen, Wei Gao, Binyang Li, Lanjun Zhou, Yulan He and Kam-Fai Wong
- 17:45 *PARMA: A Predicate Argument Aligner*  
Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathan Weese, Tan Xu and Xuchen Yao

**Monday August 5, 2013 (continued)**

**(16:45 - 18:05) SP 4d**

- 16:45 *Aggregated Word Pair Features for Implicit Discourse Relation Disambiguation*  
Or Biran and Kathleen McKeown
- 17:05 *Implicatures and Nested Beliefs in Approximate Decentralized-POMDPs*  
Adam Vogel, Christopher Potts and Dan Jurafsky
- 17:25 *Domain-Specific Coreference Resolution with Lexicalized Features*  
Nathan Gilbert and Ellen Riloff
- 17:45 *Learning to Order Natural Language Texts*  
Jiwei Tan, Xiaojun Wan and Jianguo Xiao

**(16:45 - 18:05) SP 4e**

- 16:45 *Universal Dependency Annotation for Multilingual Parsing*  
Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló and Jungmee Lee
- 17:05 *An Empirical Examination of Challenges in Chinese Parsing*  
Jonathan K. Kummerfeld, Daniel Tse, James R. Curran and Dan Klein
- 17:25 *Joint Inference for Heterogeneous Dependency Parsing*  
Guangyou Zhou and Jun Zhao
- 17:45 *Easy-First POS Tagging and Dependency Parsing with Beam Search*  
Ji Ma, Jingbo Zhu, Tong Xiao and Nan Yang

**Monday August 5, 2013 (continued)**

**(18:30 - 19:45) Poster Session A**

**SP - Cognitive Modelling and Psycholinguistics**

*Arguments and Modifiers from the Learner's Perspective*

Leon Bergen, Edward Gibson and Timothy J. O'Donnell

**SP - Dialogue and Interactive Systems**

*Benefactive/Malefactive Event and Writer Attitude Annotation*

Lingjia Deng, Yoonjung Choi and Janyce Wiebe

**SP- Discourse, Coreference and Pragmatics**

*GuiTAR-based Pronominal Anaphora Resolution in Bengali*

Apurbalal Senapati and Utpal Garain

**SP - Evaluation Methods**

*A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art*

Peter A. Rankel, John M. Conroy, Hoa Trang Dang and Ani Nenkova

*On the Predictability of Human Assessment: when Matrix Completion Meets NLP Evaluation*

Guillaume Wisniewski

*Automated Pyramid Scoring of Summaries using Distributional Semantics*

Rebecca J. Passonneau, Emily Chen, Weiwei Guo and Dolores Perin



**Monday August 5, 2013 (continued)**

**SP - Information Retrieval**

*Are Semantically Coherent Topic Models Useful for Ad Hoc Information Retrieval?*

Romain Deveaud, Eric SanJuan and Patrice Bellot

*Post-Retrieval Clustering Using Third-Order Similarity Measures*

Jose G. Moreno, Gaël Dias and Guillaume Cleuziou

*Automatic Coupling of Answer Extraction and Information Retrieval*

Xuchen Yao, Benjamin Van Durme and Peter Clark

**SP - Word Segmentation**

*An improved MDL-based compression algorithm for unsupervised word segmentation*

Ruey-Cheng Chen

*Co-regularizing character-based and word-based models for semi-supervised Chinese word segmentation*

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao and Isabel Trancoso

*Improving Chinese Word Segmentation on Micro-blog Using Rich Punctuations*

Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang and Ni Sun

*Accurate Word Segmentation using Transliteration and Language Model Projection*

Masato Hagiwara and Satoshi Sekine

**SP - Spoken Language Processing**

*Broadcast News Story Segmentation Using Manifold Learning on Latent Topic Distributions*

Xiaoming Lu, Lei Xie, Cheung-Chi Leung, Bin Ma and Haizhou Li

*Is word-to-phone mapping better than phone-phone mapping for handling English words?*

Naresh Kumar Elluru, Anandaswarup Vadapalli, Raghavendra Elluru, Hema Murthy and Kishore Prahallad

**Monday August 5, 2013 (continued)**

**SP - Multilinguality**

*Enriching Entity Translation Discovery using Selective Temporality*

Gae-won You, Young-rok Cha, Jinhan Kim and Seung-won Hwang

*Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling*

Heike Adel, Ngoc Thang Vu and Tanja Schultz

*Latent Semantic Matching: Application to Cross-language Text Categorization without Alignment Information*

Tsutomu Hirao, Tomoharu Iwata and Masaaki Nagata

**SP - NLP Applications**

*TopicSpam: a Topic-Model based approach for spam detection*

Jiwei Li, Claire Cardie and Sujian Li

*Semantic Neighborhoods as Hypergraphs*

Chris Quirk and Pallavi Choudhury

*Unsupervised joke generation from big data*

Saša Petrović and David Matthews

*Modeling of term-distance and term-occurrence information for improving n-gram language model performance*

Tze Yuang Chong, Rafael E. Banchs, Eng Siong Chng and Haizhou Li

*Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners*

Keisuke Sakaguchi, Yuki Arase and Mamoru Komachi

**Monday August 5, 2013 (continued)**

**SP - NLP and Creativity**

*"Let Everything Turn Well in Your Wife": Generation of Adult Humor Using Lexical Constraints*

Alessandro Valitutti, Hannu Toivonen, Antoine Doucet and Jukka M. Toivanen

*Random Walk Factoid Annotation for Collective Discourse*

Ben King, Rahul Jha, Dragomir Radev and Robert Mankoff

**SP - NLP for the Languages of Central and Eastern Europe and the Balkans**

*Identifying English and Hungarian Light Verb Constructions: A Contrastive Approach*

Veronika Vincze, István Nagy T. and Richárd Farkas

*English-to-Russian MT evaluation campaign*

Pavel Braslavski, Alexander Beloborodov, Maxim Khalilov and Serge Sharoff

**SP - Language Resources**

*IndoNet: A Multilingual Lexical Knowledge Network for Indian Languages*

Brijesh Bhatt, Lahari Poddar and Pushpak Bhattacharyya

*Building Japanese Textual Entailment Specialized Data Sets for Inference of Basic Sentence Relations*

Kimi Kaneko, Yusuke Miyao and Daisuke Bekki

*Building Comparable Corpora Based on Bilingual LDA Model*

Zede Zhu, Miao Li, Lei Chen and Zhenxin Yang

**Monday August 5, 2013 (continued)**

**SP - Lexical Semantics and Ontologies**

*Using Lexical Expansion to Learn Inference Rules from Sparse Data*

Oren Melamud, Ido Dagan, Jacob Goldberger and Idan Szpektor

*Mining Equivalent Relations from Linked Data*

Ziqi Zhang, Anna Lisa Gentile, Isabelle Augenstein, Eva Blomqvist and Fabio Ciravegna

**SP - Low Resource Language Processing**

*Context-Dependent Multilingual Lexical Lookup for Under-Resourced Languages*

Lian Tze Lim, Lay-Ki Soon, Tek Yong Lim, Enya Kong Tang and Bali Ranaivo-Malançon

*Sorani Kurdish versus Kurmanji Kurdish: An Empirical Comparison*

Kyumars Sheykh Esmaili and Shahin Salavati

*Enhanced and Portable Dependency Projection Algorithms Using Interlinear Glossed Text*

Ryan Georgi, Fei Xia and William D. Lewis

*Cross-lingual Projections between Languages from Different Families*

Mo Yu, Tiejun Zhao, Yalong Bai, Hao Tian and Dianhai Yu

*Using Context Vectors in Improving a Machine Translation System with Bridge Language*

Samira Tofighi Zahabi, Somayeh Bakhshaei and Shahram Khadivi

**SP - Machine Translation: Methods, Applications and Evaluations**

*Task Alternation in Parallel Sentence Retrieval for Twitter Translation*

Felix Hieber, Laura Jehl and Stefan Riezler

*Sign Language Lexical Recognition With Propositional Dynamic Logic*

Arturo Curiel and Christophe Collet

*Stacking for Statistical Machine Translation*

Majid Razmara and Anoop Sarkar

**Monday August 5, 2013 (continued)**

*Bilingual Data Cleaning for SMT using Graph-based Random Walk*

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li and Ming Zhou

*Automatically Predicting Sentence Translation Difficulty*

Abhijit Mishra, Pushpak Bhattacharyya and Michael Carl

*Learning to Prune: Context-Sensitive Pruning for Syntactic MT*

Wenduan Xu, Yue Zhang, Philip Williams and Philipp Koehn

*A Novel Graph-based Compact Representation of Word Alignment*

Qun Liu, Zhaopeng Tu and Shouxun Lin

*Stem Translation with Affix-Based Rule Selection for Agglutinative Languages*

Zhiyang Wang, Yajuan Lü, Meng Sun and Qun Liu

*A Novel Translation Framework Based on Rhetorical Structure Theory*

Mei Tu, Yu Zhou and Chengqing Zong

*Improving machine translation by training against an automatic semantic frame based evaluation metric*

Chi-kiu Lo, Karteek Addanki, Markus Saers and Dekai Wu

**(19:45 - 21:00) Poster Session B**

**SP - Machine Translation: Statistical Models**

*Bilingual Lexical Cohesion Trigger Model for Document-Level Machine Translation*

Guosheng Ben, Deyi Xiong, Zhiyang Teng, Yajuan Lü and Qun Liu

*Generalized Reordering Rules for Improved SMT*

Fei Huang and Cezar Pendus

*A Tightly-coupled Unsupervised Clustering and Bilingual Alignment Model for Transliteration*

Tingting Li, Tiejun Zhao, Andrew Finch and Chunyue Zhang

*Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?*

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang and Philipp Koehn

**Monday August 5, 2013 (continued)**

*Learning Non-linear Features for Machine Translation Using Gradient Boosting Machines*

Kristina Toutanova and Byung-Gyu Ahn

*Language Independent Connectivity Strength Features for Phrase Pivot Statistical Machine Translation*

Ahmed El Kholly, Nizar Habash, Gregor Leusch, Evgeny Matusov and Hassan Sawaf

*Semantic Roles for String to Tree Machine Translation*

Marzieh Bazrafshan and Daniel Gildea

### **SP -Question Answering**

*Minimum Bayes Risk based Answer Re-ranking for Question Answering*

Nan Duan

*Question Classification Transfer*

Anne-Laure Ligozat

*Latent Semantic Tensor Indexing for Community-based Question Answering*

Xipeng Qiu, Le Tian and Xuanjing Huang

### **SP - Semantics**

*Measuring semantic content in distributional vectors*

Aur lie Herbelot and Mohan Ganesalingam

*Modeling Human Inference Process for Textual Entailment Recognition*

Hen-Hsen Huang, Kai-Chun Chang and Hsin-Hsi Chen

*Recognizing Partial Textual Entailment*

Omer Levy, Torsten Zesch, Ido Dagan and Iryna Gurevych

*Sentence Level Dialect Identification in Arabic*

Heba Elfardy and Mona Diab

*Leveraging Domain-Independent Information in Semantic Parsing*

Dan Goldwasser and Dan Roth

**Monday August 5, 2013 (continued)**

*A Structured Distributional Semantic Model for Event Co-reference*

Kartik Goyal, Sujay Kumar Jauhar, Huiying Li, Mrinmaya Sachan, Shashank Srivastava and Eduard Hovy

**SP - Sentiment Analysis, Opinion Mining and Text Classification**

*Text Classification from Positive and Unlabeled Data using Misclassified Data Correction*

Fumiyo Fukumoto, Yoshimi Suzuki and Suguru Matsuyoshi

*Character-to-Character Sentiment Analysis in Shakespeare's Plays*

Eric T. Nalisnick and Henry S. Baird

*A Novel Classifier Based on Quantum Computation*

Ding Liu, Xiaofang Yang and Minghu Jiang

*Re-embedding words*

Igor Labutov and Hod Lipson

*LABR: A Large Scale Arabic Book Reviews Dataset*

Mohamed Aly and Amir Atiya

*Generating Recommendation Dialogs by Extracting Information from User Reviews*

Kevin Reschke, Adam Vogel and Dan Jurafsky

*Exploring Sentiment in Social Media: Bootstrapping Subjectivity Clues from Multilingual Twitter Streams*

Svitlana Volkova, Theresa Wilson and David Yarowsky

*Joint Modeling of News Reader's and Comment Writer's Emotions*

Huanhuan Liu, Shoushan Li, Guodong Zhou, Chu-ren Huang and Peifeng Li

*An annotated corpus of quoted opinions in news articles*

Tim O'Keefe, James R. Curran, Peter Ashwell and Irena Koprinska

*Dual Training and Dual Prediction for Polarity Classification*

Rui Xia, Tao Wang, Xuelei Hu, Shoushan Li and Chengqing Zong

*Co-Regression for Cross-Language Review Rating Prediction*

Xiaojun Wan

**Monday August 5, 2013 (continued)**

**SP - Statistical and Machine Learning Methods in NLP**

*Extracting Definitions and Hypernym Relations relying on Syntactic Dependencies and Support Vector Machines*

Guido Boella and Luigi Di Caro

*Neighbors Help: Bilingual Unsupervised WSD Using Context*

Sudha Bhingardive, Samiulla Shaikh and Pushpak Bhattacharyya

*Reducing Annotation Effort for Quality Estimation via Active Learning*

Daniel Beck, Lucia Specia and Trevor Cohn

*Reranking with Linguistic and Semantic Features for Arabic Optical Character Recognition*

Nadi Tomeh, Nizar Habash, Ryan Roth, Noura Farra, Pradeep Dasigi and Mona Diab

**SP - Summarization and Generation**

*Evolutionary Hierarchical Dirichlet Process for Timeline Summarization*

Jiwei Li and Sujian Li

*Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts*

Gerasimos Lampouras and Ion Androutsopoulos

*Sequential Summarization: A New Application for Timely Updated Twitter Trending Topics*

Dehong Gao, Wenjie Li and Renxian Zhang

*A System for Summarizing Scientific Topics Starting from Keywords*

Rahul Jha, Amjad Abu-Jbara and Dragomir Radev



**Monday August 5, 2013 (continued)**

**SP - Syntax and Parsing**

*A Unified Morpho-Syntactic Scheme of Stanford Dependencies*

Reut Tsarfaty

*Dependency Parser Adaptation with Subtrees from Auto-Parsed Target Domain Data*

Xuezhe Ma and Fei Xia

*Iterative Transformation of Annotation Guidelines for Constituency Parsing*

Xiang Li, Wenbin Jiang, Yajuan Lü and Qun Liu

*Nonparametric Bayesian Inference and Efficient Parsing for Tree-adjoining Grammars*

Elif Yamangil and Stuart M. Shieber

*Using CCG categories to improve Hindi dependency parsing*

Bharat Ram Ambati, Tejaswini Deoskar and Mark Steedman

*The Effect of Higher-Order Dependency Features in Discriminative Phrase-Structure Parsing*

Greg Coppola and Mark Steedman

*Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers*

Andre Martins, Miguel Almeida and Noah A. Smith

*A Lattice-based Framework for Joint Chinese Word Segmentation, POS Tagging and Parsing*

Zhiguo Wang, Chengqing Zong and Nianwen Xue

*Efficient Implementation of Beam-Search Incremental Parsers*

Yoav Goldberg, Kai Zhao and Liang Huang

**Monday August 5, 2013 (continued)**

**SP - Tagging and Chunking**

*Simpler unsupervised POS tagging with bilingual projections*

Long Duong, Paul Cook, Steven Bird and Pavel Pecina

*Part-of-speech tagging with antagonistic adversaries*

Anders Søgaard

**SP - Text Mining and Information Extraction**

*Temporal Signals Help Label Temporal Relations*

Leon Derczynski and Robert Gaizauskas

*Diverse Keyword Extraction from Conversations*

Maryam Habibi and Andrei Popescu-Belis

*Understanding Tables in Context Using Standard NLP Toolkits*

Vidhya Govindaraju, Ce Zhang and Christopher Ré

*Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction*

Wei Xu, Raphael Hoffmann, Le Zhao and Ralph Grishman

*Joint Apposition Extraction with Syntactic and Semantic Constraints*

Will Radford and James R. Curran

**Tuesday August 6, 2013**

**(7:30 - 17:00) Registration**

**(9:00) Industrial Lecture: Lars Rasmussen (Facebook)**

**(10:00) Best Paper Award**

**(10:30) Coffee Break**

**Oral Presentations**

**(12:15) Lunch break**

**(16:15) Coffee Break**

**(16:45 - 18:05) SP 8a**

- 16:45 *Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation*  
Kevin Duh, Graham Neubig, Katsuhito Sudoh and Hajime Tsukada
- 17:05 *Mapping Source to Target Strings without Alignment by Analogical Learning: A Case Study with Transliteration*  
Phillippe Langlais
- 17:25 *Scalable Modified Kneser-Ney Language Model Estimation*  
Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark and Philipp Koehn
- 17:45 *Incremental Topic-Based Translation Model Adaptation for Conversational Spoken Language Translation*  
Sanjika Hewavitharana, Dennis Mehay, Sankaranarayanan Ananthakrishnan and Prem Natarajan

**Tuesday August 6, 2013 (continued)**

**(16:45 - 18:05) SP 8b**

- 16:45 *A Lightweight and High Performance Monolingual Word Aligner*  
Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch and Peter Clark
- 17:05 *A Learner Corpus-based Approach to Verb Suggestion for ESL*  
Yu Sawai, Mamoru Komachi and Yuji Matsumoto
- 17:25 *Learning Semantic Textual Similarity with Structural Representations*  
Aliaksei Severyn, Massimo Nicosia and Alessandro Moschitti
- 17:45 *Typesetting for Improved Readability using Lexical and Syntactic Information*  
Ahmed Salama, Kemal Oflazer and Susan Hagan

**(16:45 - 18:05) SP 8c**

- 16:45 *Annotation of regular polysemy and underspecification*  
Héctor Martínez Alonso, Bolette Sandford Pedersen and Núria Bel
- 17:05 *Derivational Smoothing for Syntactic Distributional Semantics*  
Sebastian Padó, Jan Šnajder and Britta Zeller
- 17:25 *Diathesis alternation approximation for verb clustering*  
Lin Sun, Diana McCarthy and Anna Korhonen
- 17:45 *Outsourcing FrameNet to the Crowd*  
Marco Fossati, Claudio Giuliano and Sara Tonelli

**Tuesday August 6, 2013 (continued)**

**(16:45 - 18:05) SP 8d**

- 16:45 *Smatch: an Evaluation Metric for Semantic Feature Structures*  
Shu Cai and Kevin Knight
- 17:05 *Variable Bit Quantisation for LSH*  
Sean Moran, Victor Lavrenko and Miles Osborne
- 17:25 *Context Vector Disambiguation for Bilingual Lexicon Extraction from Comparable Corpora*  
Dhouha Bouamor, Nasredine Semmar and Pierre Zweigenbaum
- 17:45 *The Effects of Lexical Resource Quality on Preference Violation Detection*  
Jesse Dunietz, Lori Levin and Jaime Carbonell

**(18:30) Banquet**

**Wednesday August 7, 2013**

**(9:30) Invited Talk 3: Chantal Prat**

**(10:30) Coffee Break**

**Oral Presentations**

**(12:15) Lunch break**

**Wednesday August 7, 2013 (continued)**

**(13:30) ACL Business Meeting**

**(15:00 -16:45) SP 10d**

15:00 *Exploiting Qualitative Information from Automatic Word Alignment for Cross-lingual NLP Tasks*

José G.C. de Souza, Miquel Esplà-Gomis, Marco Turchi and Matteo Negri

15:35 *An Information Theoretic Approach to Bilingual Word Clustering*

Manaal Faruqui and Chris Dyer

15:55 *Building and Evaluating a Distributional Memory for Croatian*

Jan Šnajder, Sebastian Padó and Željko Agić

16:15 *Generalizing Image Captions for Image-Text Parallel Corpus*

Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara Berg and Yejin Choi

**(16:15) Coffee Break**

**(16:45 - 18:05) SP 11a**

16:45 *Recognizing Identical Events with Graph Kernels*

Goran Glavaš and Jan Šnajder

17:05 *Automatic Term Ambiguity Detection*

Tyler Baldwin, Yunyao Li, Bogdan Alexe and Ioana R. Stanoi

17:25 *Towards Accurate Distant Supervision for Relational Facts Extraction*

Xingxing Zhang, Jianwen Zhang, Junyu Zeng, Jun Yan, Zheng Chen and Zhifang Sui

17:45 *Extra-Linguistic Constraints on Stance Recognition in Ideological Debates*

Kazi Saidul Hasan and Vincent Ng

**Wednesday August 7, 2013 (continued)**

**(16:45 - 18:05) SP 11b**

- 16:45 *Are School-of-thought Words Characterizable?*  
Xiaorui Jiang, Xiaoping Sun and Hai Zhuge
- 17:05 *Identifying Opinion Subgroups in Arabic Online Discussions*  
Amjad Abu-Jbara, Ben King, Mona Diab and Dragomir Radev
- 17:25 *Extracting Events with Informal Temporal References in Personal Histories in Online Communities*  
Miaomiao Wen, Zeyu Zheng, Hyeju Jang, Guang Xiang and Carolyn Penstein Rosé
- 17:45 *Multimodal DBN for Predicting High-Quality Answers in cQA portals*  
Haifeng Hu, Bingquan Liu, Baoxun Wang, Ming Liu and Xiaolong Wang

**(16:45 - 18:05) SP 11c**

- 16:45 *Bi-directional Inter-dependencies of Subjective Expressions and Targets and their Value for a Joint Model*  
Roman Klinger and Philipp Cimiano
- 17:05 *Identifying Sentiment Words Using an Optimization-based Model without Seed Words*  
Hongliang Yu, Zhi-Hong Deng and Shiyinxue Li
- 17:25 *Detecting Turnarounds in Sentiment Analysis: Thwarting*  
Ankit Ramteke, Akshat Malu, Pushpak Bhattacharyya and J. Saketha Nath
- 17:45 *Explicit and Implicit Syntactic Features for Text Classification*  
Matt Post and Shane Bergsma

**Wednesday August 7, 2013 (continued)**

**(16:45 - 18:05) SP 11d**

- 16:45 *Does Korean defeat phonotactic word segmentation?*  
Robert Daland and Kie Zuraw
- 17:05 *Word surprisal predicts N400 amplitude during reading*  
Stefan L. Frank, Leun J. Otten, Giulia Galli and Gabriella Vigliocco
- 17:25 *Computerized Analysis of a Verbal Fluency Test*  
James O. Ryan, Serguei Pakhomov, Susan Marino, Charles Bernick and Sarah Banks
- 17:45 *A New Set of Norms for Semantic Relatedness Measures*  
Sean Szumlanski, Fernando Gomez and Valerie K. Sims

**(18:30) Lifetime Achievement Award Session**

**(19:15) Closing Session**

**(19:30) End**



# Translating Dialectal Arabic to English

**Hassan Sajjad, Kareem Darwish**

Qatar Computing Research Institute

Qatar Foundation

{hsajjad, kdarwish}@qf.org.qa

**Yonatan Belinkov**

CSAIL

Massachusetts Institute of Technology

belinkov@mit.edu

## Abstract

We present a dialectal Egyptian Arabic to English statistical machine translation system that leverages dialectal to Modern Standard Arabic (MSA) adaptation. In contrast to previous work, we first narrow down the gap between Egyptian and MSA by applying an automatic character-level transformational model that changes Egyptian to  $EG'$ , which looks similar to MSA. The transformations include morphological, phonological and spelling changes. The transformation reduces the out-of-vocabulary (OOV) words from 5.2% to 2.6% and gives a gain of 1.87 BLEU points. Further, adapting large MSA/English parallel data increases the lexical coverage, reduces OOVs to 0.7% and leads to an absolute BLEU improvement of 2.73 points.

## 1 Introduction

Modern Standard Arabic (MSA) is the lingua franca for the Arab world. Arabic speakers generally use dialects in daily interactions. There are 6 dominant dialects, namely Egyptian, Moroccan, Levantine, Iraqi, Gulf, and Yemeni<sup>1</sup>. The dialects may differ in vocabulary, morphology, syntax, and spelling from MSA and most lack spelling conventions.

Different dialects often make different lexical choices to express concepts. For example, the concept corresponding to “Oryd” أريد (“I want”) is expressed as “EAwz” عاوز in Egyptian, “Abgy” ابغي in Gulf, “Aby” ابي in Iraqi, and “bdy” بدني in Levantine<sup>2</sup>. Often, words have different or opposite meanings in different dialects.

<sup>1</sup>[http://en.wikipedia.org/wiki/Varieties\\_of\\_Arabic](http://en.wikipedia.org/wiki/Varieties_of_Arabic)

<sup>2</sup>All transliterations follow the Buckwalter scheme

Arabic dialects may differ morphologically from MSA. For example, Egyptian Arabic uses a negation construct similar to the French “ne pas” negation construct. The Egyptian word “mlEbt\$” ملعبتش (or alternatively spelled مالعبتش) (“I did not play”) is composed of “m+lEbt+\$”.

The pronunciations of letters often differ from one dialect to another. For example, the letter “q” ق is typically pronounced in MSA as an unvoiced uvular stop (as the “q” in “quote”), but as a glottal stop in Egyptian and Levantine (like “A” in “Alpine”) and a voiced velar stop in the Gulf (like “g” in “gavel”). Differing pronunciations often reflect on spelling.

Social media platforms allowed people to express themselves more freely in writing. Although MSA is used in formal writing, dialects are increasingly being used on social media sites. Some notable trends on social platforms include (Darwish et al., 2012):

- Mixed language texts where bilingual (or multilingual) users code switch between Arabic and English (or Arabic and French). In the example “wSlny mrsy” وصلني مرسي (“got it thank you”), “thank you” is the transliterated French word “merci”.
- The use of phonetic transcription to match dialectal pronunciation. For example, “Sdq” صدق (“truth”) is often written as “Sj” صج in Gulf dialect.
- Creative spellings, spelling mistakes, and word elongations are ubiquitous in social texts.
- The use of new words like “lol” لول (“LOL”).
- The attachment of new meanings to words such as using “THn” طحن to mean “very” while it means “grinding” in MSA.

The Egyptian dialect has the largest number of speakers and is the most commonly understood dialect in the Arab world. In this work, we focused on translating dialectal Egyptian to English us-

ing Egyptian to MSA adaptation. Unlike previous work, we first narrowed the gap between Egyptian and MSA using character-level transformations and word n-gram models that handle spelling mistakes, phonological variations, and morphological transformations. Later, we applied an adaptation method to incorporate MSA/English parallel data.

The contributions of this paper are as follows:

- We trained an Egyptian/MSA transformation model to make Egyptian look similar to MSA. We publicly released the training data.
- We built a phrasal Machine Translation (MT) system on adapted Egyptian/English parallel data, which outperformed a non-adapted baseline by 1.87 BLEU points.
- We used phrase-table merging (Nakov and Ng, 2009) to utilize MSA/English parallel data with the available in-domain parallel data.

## 2 Previous Work

Our work is related to research on MT from a resource poor language (to other languages) by pivoting on a closely related resource rich language. This can be done by either translating between the related languages using word-level translation, character level transformations, and language specific rules (Durrani et al., 2010; Hajič et al., 2000; Nakov and Tiedemann, 2012), or by concatenating the parallel data for both languages (Nakov and Ng, 2009). These translation methods generally require parallel data, for which hardly any exists between dialects and MSA. Instead of translating between a dialect and MSA, we tried to narrow down the lexical, morphological and phonetic gap between them using a character-level conversion model, which we trained on a small set of parallel dialect/MSA word pairs.

In the context of Arabic dialects<sup>3</sup>, most previous work focused on converting dialects to MSA and vice versa to improve the processing of dialects (Sawaf, 2010; Chiang et al., 2006; Mohamed et al., 2012; Utiyama and Isahara, 2008). Sawaf (2010) proposed a dialect to MSA normalization that used character-level rules and morphological analysis. Salloum and Habash (2011) also used a rule-based method to generate MSA paraphrases of dialectal out-of-vocabulary (OOV) and low frequency words. Instead of rules, we automatically

<sup>3</sup>Due to space limitations, we restrict discussion to work on dialects only.

learnt character mappings from dialect/MSA word pairs.

Zbib et al. (2012) explored several methods for dialect/English MT. Their best Egyptian/English system was trained on dialect/English parallel data. They used two language models built from the English GigaWord corpus and from a large web crawl. Their best system outperformed manually translating Egyptian to MSA then translating using an MSA/English system. In contrast, we showed that training on in-domain dialectal data irrespective of its small size is better than training on large MSA/English data. Our LM experiments also affirmed the importance of in-domain English LMs. We also showed that a conversion does not imply a straight forward usage of MSA resources and there is a need for adaptation which we fulfilled using phrase-table merging (Nakov and Ng, 2009).

### 2.1 Baseline

We constructed baselines that were based on the following training data:

- An Egyptian/English parallel corpus consisting of  $\approx 38k$  sentences, which is part of the LDC2012T09 corpus (Zbib et al., 2012). We randomly divided it into 32k sentences for training, 2k for development and 4k for testing. We henceforth refer to this corpus as *EG* and the English part of it as *EG<sub>en</sub>*. We did not have access to the training/test splits of Zbib et al. (2012) to directly compare to their results.

- An MSA/English parallel corpus consisting of 200k sentences from LDC<sup>4</sup>. We refer to this corpus as the *AR* corpus.

For language modeling, we used either *EG<sub>en</sub>* or the English side of the *AR* corpus plus the English side of NIST12 training data and English GigaWord v5. We refer to this corpus as *GW*.

We tokenized Egyptian and Arabic according to the ATB tokenization scheme using the MADA+TOKAN morphological analyzer and tokenizer v3.1 (Roth et al., 2008). Word elongations were already fixed in the corpus. We word-aligned the parallel data using GIZA++ (Och and Ney, 2003), and symmetrized the alignments using grow-diag-final-and heuristic (Koehn et al., 2003). We trained a phrasal MT system (Koehn et al., 2003). We built five-gram LMs using KenLM

<sup>4</sup>Arabic News (LDC2004T17), eTIRR (LDC2004E72), and parallel corpora the GALE program

|           | Train     | LM                       | BLEU         | OOV |
|-----------|-----------|--------------------------|--------------|-----|
| <i>B1</i> | <i>AR</i> | <i>GW</i>                | 7.48         | 6.7 |
| <i>B2</i> | <i>EG</i> | <i>GW</i>                | 12.82        | 5.2 |
| <i>B3</i> | <i>EG</i> | <i>EG<sub>en</sub></i>   | 13.94        | 5.2 |
| <i>B4</i> | <i>EG</i> | <i>EG<sub>en</sub>GW</i> | <b>14.23</b> | 5.2 |

Table 1: Baseline results using the *EG* and *AR* training sets with *GW* and *EG<sub>en</sub>* corpora for LM training

with modified Kneser-Ney smoothing (Heafield, 2011). In case of more than one LM, we tuned their weights on a development set using Minimum Error Rate Training (Och and Ney, 2003).

We built several baseline systems as follows:

- *B1* used *AR* for training a translation model and *GW* for LM.
- *B2-B4* systems used identical training data, namely *EG*, with the *GW*, *EG<sub>en</sub>*, or both for *B2*, *B3*, and *B4* respectively for language modeling.

Table 1 reports the baseline results. The system trained on *AR* (*B1*) performed poorly compared to the one trained on *EG* (*B2*) with a 6.75 BLEU points difference. This highlights the difference between MSA and Egyptian. Using *EG* data for training both the translation and language models was effective. *B4* used two LMs and yielded the best results. For later comparison, we only use the *B4* baseline.

### 3 Proposed Methods

#### 3.1 Egyptian to *EG'* Conversion

As mentioned previously, dialects differ from MSA in vocabulary, morphology, and phonology. Dialectal spelling often follows dialectal pronunciation, and dialects lack standard spelling conventions. To address the vocabulary problem, we used the *EG* corpus for training.

To address the spelling and morphological differences, we trained a character-level mapping model to generate MSA words from dialectal ones using character transformations. To train the model, we extracted the most frequent words from a dialectal Egyptian corpus, which had 12,527 news comments (containing 327k words) from Al-Youm Al-Sabe news site (Zaidan and Callison-Burch, 2011) and translated them to their equivalent MSA words. We hired a professional translator, who generated one or more translations of the most frequent 5,581 words into MSA. Out of these word pairs, 4,162 involved character-level transformations due to phonological, morphologi-

cal, or spelling changes. We aligned the translated pairs at character level using GIZA++ and Moses in the manner described in Section 2.1. As in the baseline of Kahki et al. (2011), given a source word, we produced all of its possible segmentations along with their associated character-level mappings. We restricted individual source character sequences to be 3 characters at most. We retained all mapping sequences leading to valid words in a large lexicon. We built the lexicon from a set of 234,638 Aljazeera articles<sup>5</sup> that span a 10 year period and contain 254M tokens. Spelling mistakes in Aljazeera articles were very infrequent. We sorted the candidates by the product of the constituent mapping probabilities and kept the top 10 candidates. Then we used a trigram LM that we built from the aforementioned Aljazeera articles to pick the most likely candidate in context. We simply multiplied the character-level transformation probability with the LM probability – giving them equal weight. Since Egyptian has a “ne pas” like negation construct that involves putting a “م” and “ش” at the beginning and end of verbs, we handled words that had negation by removing these two letters, then applying our character transformation, and lastly adding the negation article “لا” لا before the verb. We converted the *EG* train, tune, and test parts. We refer to the converted corpus as *EG'*.

As an example, our system transformed “بس اللي بيحصلهم ميعجبش حد” (“what is happening to them does not please anyone”) to “بس الذي يحصل لهم لا يعجب حد”. Transforming “Ally” اللي to “Al\*y” الذي involved a spelling correction. The transformation of “byHSlhm” بيحصلهم to “yHSl lhm” يحصل لهم involved a morphological change and word splitting. Changing “myEjbs\$” ميعجبش to “la yEjb” لا يعجب involved morphologically transforming a negation construct.

#### 3.2 Combining *AR* and *EG'*

The aforementioned conversion generated a language that is close, but not identical, to MSA. In order to maximize the gain using both parallel corpora, we used the phrase merging technique described in Nakov and Ng (2009) to merge the phrase tables generated from the *AR* and *EG'* corpora. If a phrase occurred in both phrase tables, we

<sup>5</sup><http://www.aljazeera.net>

adopted one of the following three solutions:

- Only added the phrase with its translations and their probabilities from the *AR* phrase table. This assumed *AR* alignments to be more reliable.
  - Only added the phrase with its translations and their probabilities from the *EG'* phrase table. This assumed *EG'* alignments to be more reliable.
  - Added translations of the phrase from both phrase tables and left the choice to the decoder.
- We added three additional features to the new phrase table to avail the information about the origin of phrases (as in Nakov and Ng (2009)).

### 3.3 Evaluation and Discussion

We performed the following experiments:

- *S0* involved translating the *EG'* test using *AR*.
- *S1* and *S2* trained on the *EG'* with *EG<sub>en</sub>* and both *EG<sub>en</sub>* and *GW* for LM training respectively.
- *S\** used phrase merging technique. All systems trained on both *EG'* and *AR* corpora. We built separate phrase tables from the two corpora and merged them. When merging, we preferred *AR* or *EG'* for *S<sub>AR</sub>* and *S<sub>EG'</sub>* respectively. For *S<sub>ALL</sub>*, we kept phrases from both phrase tables.

Table 2 summarizes results of using *EG'* and phrase table merging. *S0* was slightly better than *B1*, but lagged considerably behind training using *EG* or *EG'*. *S1*, which used only *EG'* for training showed an improvement of 1.67 BLEU points from the best baseline system (*B4*). Using both language models (*S2*) led to slight improvement. Phrase merging that preferred phrases learnt from *EG'* data over *AR* data performed the best with a BLEU score of 16.96.

|                        | Train                      | LM                       | BLEU         | OOV |
|------------------------|----------------------------|--------------------------|--------------|-----|
| <i>B4</i>              | <i>EG</i>                  | <i>EG<sub>en</sub>GW</i> | 14.23        | 5.2 |
| <i>S0</i>              | <i>AR</i>                  | <i>EG<sub>en</sub></i>   | 8.61         | 2.0 |
| <i>S1</i>              | <i>EG'</i>                 | <i>EG<sub>en</sub></i>   | 15.90        | 2.6 |
| <i>S2</i>              | <i>EG'</i>                 | <i>EG<sub>en</sub>GW</i> | 16.10        | 2.6 |
| <i>S<sub>AR</sub></i>  | <i>PT<sub>AR</sub></i>     | <i>EG<sub>en</sub>GW</i> | 16.14        | 0.7 |
| <i>S<sub>EG'</sub></i> | <i>PT<sub>EG'</sub></i>    | <i>EG<sub>en</sub>GW</i> | <b>16.96</b> | 0.7 |
| <i>S<sub>ALL</sub></i> | <i>PT<sub>EG',AR</sub></i> | <i>EG<sub>en</sub>GW</i> | 16.73        | 0.7 |

Table 2: Summary of results using different combinations of *EG'*/English and MSA/English training data

We analyzed 100 test sentences that led to the greatest absolute change in BLEU score, whether positive or negative, between training with *EG* and *EG'*. The largest difference in BLEU was 0.69 in favor of *EG'*. Translating the Egyp-

tian sentence “wbyHtrmwA AlnAs AltAnyp” وبيحترموا الناس الثانية (OOV) produced “ويحترموا” (BLEU = 0.31). Conversion changed “wbyHtrmwA” to “wyHtrmwA” and “AltAnyp” الثانية to “AlvAnyp” الثانية, leading to “and they respect other people” (BLEU = 1). Training with *EG'* outperformed *EG* for 63 of the sentences. Conversion improved MT, because it reduced OOVs, enabled MADA+TOKAN to successfully analyze words, and reduced spelling mistakes.

In further analysis, we examined 1% of the sentences with the largest difference in BLEU score. Out of these, more than 70% were cases where the *EG'* model achieved a higher BLEU score. For each observed conversion error, we identified its linguistic character, i.e. whether it is lexical, syntactic, morphological or other. We found that in more than half of the cases ( $\approx 57\%$ ) using morphological information could have improved the conversion. Consider the following example, where (1) is the original *EG* sentence and its *EG/EN* translation, and (2) is the converted *EG'* sentence and its *EG'/EN* translation:

1. لان دي حسب رغبتك  
lAn dy Hsb rgbtk  
because this is according to your desire
2. لأن هذه حسب رغبته  
lOn h\*h Hsb rgbth  
because this is according to his desire

In this case, “rgbtk” رغبتك (“your wish”) was converted to “rgbth” رغبته (“his wish”) leading to an unwanted change in the translation. This could be avoided, for instance, by running a morphological analyzer on the original and converted word, and making sure their morphological features (in this case, the person of the possessive) correspond. In a similar case, the phrase “mEndy\$ AEdA” معنديش اعداء was converted to “Endy OEdA” عندي اعداء, thereby changing the translation from “I don’t have enemies” to “I have enemies”. Here, again, a morphological analyzer could verify the retaining of negation after conversion.

In another sentence, “knty” كنتي (“you (fm.) were”) was correctly converted to the MSA “knt” كنت, which is used for feminine and masculine forms. However, the induced ambiguity ended up hurting translation.

Aside from morphological mistakes, conversion often changed words completely. In one sentence, the word “lbAnh” لبانه (“chewing gum”) was wrongly converted to “lOnh” لأنه (“because it”), resulting in a wrong translation. Perhaps a morphological analyzer, or just a part-of-speech tagger, could enforce (or probabilistically encourage) a match in parts of speech.

The conversion also faces some other challenges. Consider the following example:

1. هوا احنا عملنا اتيه  
hwA AHnA EmlnA Ayyyh  
he is we did we What ? ?
2. هو نحن عملنا ايه  
hw nHn EmlnA Ayh  
he we did we do ? ?

While the first two words “hwA AHnA” هوا احنا were correctly converted to “hw nHn” هو نحن, the final word “Ayyyh” اتيه (“what”) was shortened but remained dialectal “Ayh” ايه rather than MSA “mA/mA\*A” ماذا. There is a syntactic challenge in this sentence, since the Egyptian word order in interrogative sentences is normally different from the MSA word order: the interrogative particle appears at the end of the sentence instead of at the beginning. Addressing this problem might have improved translation.

The above analysis suggests that incorporating deeper linguistic information in the conversion procedure could improve translation quality. In particular, using a morphological analyzer seems like a promising possibility. One approach could be to run a morphological analyzer for dialectal Arabic (e.g. MADA-ARZ (Habash et al., 2013)) on the original *EG* sentence and another analyzer for MSA (such as MADA) on the converted *EG'* sentence, and then to compare the morphological features. Discrepancies should be probabilistically incorporated in the conversion. Exploring this approach is left for future work.

## 4 Conclusion

We presented an Egyptian to English MT system. In contrast to previous work, we used an automatic conversion method to map Egyptian close to MSA. The converted Egyptian *EG'* had fewer OOV words and spelling mistakes and improved language handling. The MT system built on the

adapted parallel data showed an improvement of 1.87 BLEU points over our best baseline. Using phrase table merging that combined *AR* and *EG'* training data in a way that preferred adapted dialectal data yielded an extra 0.86 BLEU points. We will make the training data for our conversion system publicly available.

For future work, we want to expand our work to other dialects, while utilizing dialectal morphological analysis to improve conversion. Also, we believe that improving English language modeling to match the genre of the translated sentences can have significant positive impact on translation quality.

## References

- David Chiang, Mona T. Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic dialects. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.
- Kareem Darwish, Walid Magdy, and Ahmed Mourad. 2012. Language processing for Arabic microblog retrieval. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, Maui, Hawaii, USA.
- Nadir Durrani, Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2010. Hindi-to-Urdu machine translation through transliteration. In *Proceedings of the 48th Annual Conference of the Association for Computational Linguistics*, Uppsala, Sweden.
- Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, , and Nadi Tomeh. 2013. Morphological analysis and disambiguation for dialectal Arabic. In *Proceedings of the Main Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, US.
- Jan Hajič, Jan Hric, and Vladislav Kuboň. 2000. Machine translation of very close languages. In *Proceedings of the sixth conference on Applied natural language processing*, Seattle, Washington.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK.
- Ali El Kahki, Kareem Darwish, Ahmed Saad El Din, Mohamed Abd El-Wahab, Ahmed Hefny, and Waleed Ammar. 2011. Improved transliteration mining using graph reinforcement. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, UK.

- Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Edmonton, Canada.
- Emad Mohamed, Behrang Mohit, and Kemal Oflazer. 2012. Transforming standard Arabic to colloquial Arabic. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Short Paper*, Jeju Island, Korea.
- Preslav Nakov and Hwee Tou Ng. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore.
- Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Short Paper*, Jeju Island, Korea.
- Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).
- Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio.
- Wael Salloum and Nizar Habash. 2011. Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, Edinburgh, Scotland.
- Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, Denver, Colorado.
- Masao Utiyama and Hitoshi Isahara. 2008. A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic. In *Proceedings of the 6th International Conference on Informatics and Systems*, Cairo University, Egypt.
- Omar F. Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, Portland, Oregon.
- Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of Arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada.

# Exact Maximum Inference for the Fertility Hidden Markov Model

Chris Quirk

Microsoft Research

One Microsoft Way

Redmond, WA 98052, USA

chrisq@microsoft.com

## Abstract

The notion of fertility in word alignment (the number of words emitted by a single state) is useful but difficult to model. Initial attempts at modeling fertility used heuristic search methods. Recent approaches instead use more principled approximate inference techniques such as Gibbs sampling for parameter estimation. Yet in practice we also need the single best alignment, which is difficult to find using Gibbs. Building on recent advances in dual decomposition, this paper introduces an exact algorithm for finding the single best alignment with a fertility HMM. Finding the best alignment appears important, as this model leads to a substantial improvement in alignment quality.

## 1 Introduction

Word-based translation models intended to model the translation process have found new uses identifying word correspondences in sentence pairs. These word alignments are a crucial training component in most machine translation systems. Furthermore, they are useful in other NLP applications, such as entailment identification.

The simplest models may use lexical information alone. The seminal Model 1 (Brown et al., 1993) has proved very powerful, performing nearly as well as more complicated models in some phrasal systems (Koehn et al., 2003). With minor improvements to initialization (Moore, 2004) (which may be important (Toutanova and Galley, 2011)), it can be quite competitive. Subsequent IBM models include more detailed information about context. Models

2 and 3 incorporate a positional model based on the absolute position of the word; Models 4 and 5 use a relative position model instead (an English word tends to align to a French word that is nearby the French word aligned to the previous English word). Models 3, 4, and 5 all incorporate a notion of “fertility”: the number of French words that align to any English word.

Although these latter models covered a broad range of phenomena, estimation techniques and MAP inference were challenging. The authors originally recommended heuristic procedures based on local search for both. Such methods work reasonably well, but can be computationally inefficient and have few guarantees. Thus, many researchers have switched to the HMM model (Vogel et al., 1996) and variants with more parameters (He, 2007). This captures the positional information in the IBM models in a framework that admits exact parameter estimation inference, though the objective function is not concave: local maxima are a concern.

Modeling fertility is challenging in the HMM framework as it violates the Markov assumption. Where the HMM jump model considers only the prior state, fertility requires looking across the whole state space. Therefore, the standard forward-backward and Viterbi algorithms do not apply. Recent work (Zhao and Gildea, 2010) described an extension to the HMM with a fertility model, using MCMC techniques for parameter estimation. However, they do not have an efficient means of MAP inference, which is necessary in many applications such as machine translation.

This paper introduces a method for exact MAP inference with the fertility HMM using dual decomposition. The resulting model leads to substantial improvements in alignment quality.

## 2 HMM alignment

Let us briefly review the HMM translation model as a starting point. We are given a sequence of English words  $\mathbf{e} = e_1, \dots, e_I$ . This model produces distributions over French word sequences  $\mathbf{f} = f_1, \dots, f_J$  and word alignment vectors  $\mathbf{a} = a_1, \dots, a_J$ , where  $a_j \in [0..J]$  indicates the English word generating the  $j$ th French word, 0 representing a special NULL state to handle systematically unaligned words.

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(J|I) \prod_{j=1}^J p(a_j|a_{j-1}) p(f_j|e_{a_j})$$

The generative story begins by predicting the number of words in the French sentence (hence the number of elements in the alignment vector). Then for each French word position, first the alignment variable (English word index used to generate the current French word) is selected based on only the prior alignment variable. Next the French word is predicted based on its aligned English word.

Following prior work (Zhao and Gildea, 2010), we augment the standard HMM with a fertility distribution.

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(J|I) \prod_{i=1}^I p(\phi_i|e_i) \prod_{j=1}^J p(a_j|a_{j-1}) p(f_j|e_{a_j}) \quad (1)$$

where  $\phi_i = \sum_{j=1}^J \delta(i, a_j)$  indicates the number of times that state  $j$  is visited. This deficient model wastes some probability mass on inconsistent configurations where the number of times that a state  $i$  is visited does not match its fertility  $\phi_i$ . Following in the footsteps of older, richer, and wiser colleagues (Brown et al., 1993), we forge ahead unconcerned by this complication.

### 2.1 Parameter estimation

Of greater concern is the exponential complexity of inference in this model. For the standard HMM, there is a dynamic programming algorithm to compute the posterior probability over word alignments  $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f})$ . These are the sufficient statistics gathered in the E step of EM.

The structure of the fertility model violates the Markov assumptions used in this dynamic programming method. However, we may empirically

estimate the posterior distribution using Markov chain Monte Carlo methods such as Gibbs sampling (Zhao and Gildea, 2010). In this case, we make some initial estimate of the  $\mathbf{a}$  vector, potentially randomly. We then repeatedly re-sample each element of that vector conditioned on all other positions according to the distribution  $\Pr(a_j|\mathbf{a}_{-j}, \mathbf{e}, \mathbf{f})$ . Given a complete assignment of the alignment for all words except the current, computing the complete probability including transition, emission, and jump, is straightforward. This estimate comes with a computational cost: we must cycle through all positions of the vector repeatedly to gather a good estimate. In practice, a small number of samples will suffice.

### 2.2 MAP inference with dual decomposition

Dual decomposition, also known as Lagrangian relaxation, is a method for solving complex combinatorial optimization problems (Rush and Collins, 2012). These complex problems are separated into distinct components with tractable MAP inference procedures. The subproblems are repeatedly solved with some communication over consistency until a consistent and globally optimal solution is found.

Here we are interested in the problem of finding the most likely alignment of a sentence pair  $\mathbf{e}, \mathbf{f}$ . Thus, we need to solve the combinatorial optimization problem  $\arg \max_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ . Let us rewrite the objective function as follows:

$$h(\mathbf{a}) = \sum_{i=1}^I \left( \log p(\phi_i|e_i) + \sum_{j, a_j=i} \frac{\log p(f_j|e_i)}{2} \right) + \sum_{j=1}^J \left( \log p(a_j|a_{j-1}) + \frac{\log p(f_j|e_{a_j})}{2} \right)$$

Because  $\mathbf{f}$  is fixed, the  $p(J|I)$  term is constant and may be omitted. Note how we've split the optimization into two portions. The first captures fertility as well as some component of the translation distribution, and the second captures the jump distribution and the remainder of the translation distribution.

Our dual decomposition method follows this segmentation. Define  $\mathbf{y}_{\mathbf{a}}$  as  $y_{\mathbf{a}}(i, j) = 1$  if  $a_j = i$ , and 0 otherwise. Let  $\mathbf{z} \in \{0, 1\}^{I \times J}$  be a binary



```

 $u^{(0)}(i, j) := 0 \quad \forall i \in 1..I, j \in 1..J$ 
for  $k = 1$  to  $K$ 
   $\mathbf{a}^{(k)} := \arg \max_{\mathbf{a}} \left( f(\mathbf{a}) + \sum_{i,j} u^{(k-1)}(i, j) y_{\mathbf{a}}(i, j) \right)$ 
   $\mathbf{z}^{(k)} := \arg \max_{\mathbf{z}} \left( g(\mathbf{z}) - \sum_{i,j} u^{(k-1)}(i, j) z(i, j) \right)$ 
  if  $\mathbf{y}_{\mathbf{a}} = \mathbf{z}$ 
    return  $\mathbf{a}^{(k)}$ 
  end if
   $u^{(k)}(i, j) := u^{(k-1)}(i, j) + \delta_k \left( y_{\mathbf{a}^{(k)}}(i, j) - z^{(k)}(i, j) \right)$ 
end for
return  $\mathbf{a}^{(K)}$ 

```

Figure 1: The dual decomposition algorithm for the fertility HMM, where  $\delta_k$  is the step size at the  $k$ th iteration for  $1 \leq k \leq K$ , and  $K$  is the max number of iterations.

matrix. Define the functions  $f$  and  $g$  as

$$f(\mathbf{a}) = \sum_{j=1}^J \left( \log p(a_j | a_{j-1}) + \frac{1}{2} \log p(f_j | e_{a_j}) \right)$$

$$g(\mathbf{z}) = \sum_{i=1}^I \left( \log p(\phi(\mathbf{z}_i) | e_i) + \sum_{j=1}^J \frac{z(i, j)}{2} \log p(f_j | e_i) \right)$$

Then we want to find

$$\arg \max_{\mathbf{a}, \mathbf{z}} f(\mathbf{a}) + g(\mathbf{z})$$

subject to the constraints  $y_{\mathbf{a}}(i, j) = z(i, j) \forall i, j$ . Note how this recovers the original objective function when matching variables are found.

We use the dual decomposition algorithm from Rush and Collins (2012), reproduced here in Figure 1. Note how the langrangian adds one additional term word, scaled by a value indicating whether that word is aligned in the current position. Because it is only added for those words that are aligned, we can merge this with the  $\log p(f_j | e_{a_j})$  terms in both  $f$  and  $g$ . Therefore, we can solve  $\arg \max_{\mathbf{a}} \left( f(\mathbf{a}) + \sum_{i,j} u^{(k-1)}(i, j) y_{\mathbf{a}}(i, j) \right)$  using the standard Viterbi algorithm.

The  $g$  function, on the other hand, does not have a commonly used decomposition structure. Luckily we can factor this maximization into pieces that allow for efficient computation. Note that  $g$  sums over arbitrary binary matrices. Unlike the HMM, where each French word must have exactly one English generator, this maximization allows each

```

 $z(i, j) := 0 \quad \forall (i, j) \in [1..I] \times [1..J]$ 
 $v := 0$ 
for  $i = 1$  to  $I$ 
  for  $j = 1$  to  $J$ 
     $x(j) := (\log p(f_j | e_i), j)$ 
  end for
  sort  $x$  in descending order by first component
   $max := \log p(\phi = 0 | e_i), arg := 0, sum := 0$ 
  for  $f = 1$  to  $J$ 
     $sum := sum + x[f, 1]$ 
    if  $sum + \log p(\phi = f | e_i) > max$ 
       $max := sum + \log p(\phi = f | e_i)$ 
       $arg := f$ 
    end if
  end for
   $v := v + max$ 
  for  $f = 1$  to  $arg$ 
     $z(i, x[f, 2]) := 1$ 
  end for
end for
return  $\mathbf{z}, v$ 

```

Figure 2: Algorithm for finding the arg max and max of  $g$ , the fertility-related component of the dual decomposition objective.

French word to have zero or many generators. Because assignments that are in accordance between this model and the HMM will meet the HMM's constraints, the overall dual decomposition algorithm will return valid assignments, even though individual selections for this model may fail to meet the requirements.

As the scoring function  $g$  can be decomposed into a sum of scores for each row  $\sum_i g_i$  (i.e., there are no interactions between distinct rows of the matrix) we can maximize each row independently:

$$\max_{\mathbf{z}} \sum_{i=1}^I g_i(\mathbf{z}_i) = \sum_{i=1}^I \max_{\mathbf{z}} g_i(\mathbf{z}_i)$$

Within each row, we seek the best of all  $2^J$  possible configurations. These configurations may be grouped into equivalence classes based on the number of non-zero entries. In each class, the max assignment is the one using words with the highest log probabilities; the total score of this assignment is the sum those log probabilities and the log probability of that fertility. Sorting the scores of each cell in the row in descending order by log probability allows for linear time computation of the max for each row. The algorithm described in Figure 2 finds this maximal assignment in  $O(IJ \log J)$  time, generally faster than the  $O(I^2 J)$  time used by Viterbi.

We note in passing that this maximizer is picking from an unconstrained set of binary matri-

ces. Since each English word may generate as many French words as it likes, regardless of all other words in the sentence, the underlying matrix have many more or many fewer non-zero entries than there are French words. A straightforward extension to the algorithm of Figure 2 returns only  $\mathbf{z}$  matrices with exactly  $J$  nonzero entries. Rather than maximizing each row totally independently, we keep track of the best configurations for each number of words generated in each row, and then pick the best combination that sums to  $J$ : another straightforward exercise in dynamic programming. This refinement does not change the correctness of the dual decomposition algorithm; rather it speeds the convergence.

### 3 Fertility distribution parameters

Original IBM models used a categorical distribution of fertility, one such distribution for each English word. This gives EM a great amount of freedom in parameter estimation, with no smoothing or parameter tying of even rare words. Prior work addressed this by using the single parameter Poisson distribution, forcing infrequent words to share a global parameter estimated from the fertility of all words in the corpus (Zhao and Gildea, 2010).

We explore instead a feature-rich approach to address this issue. Prior work has explored feature-rich approaches to modeling the translation distribution (Berg-Kirkpatrick et al., 2010); we use the same technique, but only for the fertility model. The fertility distribution is modeled as a log-linear distribution of  $F$ , a binary feature set:  $p(\phi|e) \propto \exp(\theta \cdot F(e, \phi))$ . We include a simple set of features:

- A binary indicator for each fertility  $\phi$ . This feature is present for all words, acting as smoothing.
- A binary indicator for each word id and fertility, if the word occurs more than 10 times.
- A binary indicator for each word length (in letters) and fertility.
- A binary indicator for each four letter word prefix and fertility.

Together these produce a distribution that can learn a reasonable distribution not only for common words, but also for rare words. Including word length information aids in for languages with compounding: long words in one language may correspond to multiple words in the other.

| Algorithm     | AER (G→E) | AER (E→G) |
|---------------|-----------|-----------|
| HMM           | 24.0      | 21.8      |
| FHMM Viterbi  | 19.7      | 19.6      |
| FHMM Dual-dec | 18.0      | 17.4      |

Table 1: Experimental results over the 120 evaluation sentences. Alignment error rates in both directions are provided here.

## 4 Evaluation

We explore the impact of this improved MAP inference procedure on a task in German-English word alignment. For training data we use the news commentary data from the WMT 2012 translation task.<sup>1</sup> 120 of the training sentences were manually annotated with word alignments.

The results in Table 1 compare several different algorithms on this same data. The first line is a baseline HMM using exact posterior computation and inference with the standard dynamic programming algorithms. The next line shows the fertility HMM with approximate posterior computation from Gibbs sampling but with final alignment selected by the Viterbi algorithm. Clearly fertility modeling is improving alignment quality. The prior work compared Viterbi with a form of local search (sampling repeatedly and keeping the max), finding little difference between the two (Zhao and Gildea, 2010). Here, however, the difference between a dual decomposition and Viterbi is significant: their results were likely due to search error.

## 5 Conclusions and future work

We have introduced a dual decomposition approach to alignment inference that substantially reduces alignment error. Unfortunately the algorithm is rather slow to converge: after 40 iterations of the dual decomposition, still only 55 percent of the test sentences have converged. We are exploring improvements to the simple sub-gradient method applied here in hopes of finding faster convergence, fast enough to make this algorithm practical. Alternate parameter estimation techniques appear promising given the improvements of dual decomposition over sampling. Once the performance issues of this algorithm are improved, exploring hard EM or some variant thereof might lead to more substantial improvements.

<sup>1</sup>[www.statmt.org/wmt12/translation-task.html](http://www.statmt.org/wmt12/translation-task.html)

## References

- Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June. Association for Computational Linguistics.
- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.
- Xiaodong He. 2007. Using word-dependent transition models in HMM-based word alignment for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 80–87, Prague, Czech Republic, June. Association for Computational Linguistics.
- Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.
- Robert C. Moore. 2004. Improving ibm word alignment model 1. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 518–525, Barcelona, Spain, July.
- Alexander M Rush and Michael Collins. 2012. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. *Journal of Artificial Intelligence Research*, 45:305–362.
- Kristina Toutanova and Michel Galley. 2011. Why initialization matters for ibm model 1: Multiple optima and non-strict convexity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 461–466, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*.
- Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA, October. Association for Computational Linguistics.

# A Tale about PRO and Monsters

Preslav Nakov, Francisco Guzmán and Stephan Vogel

Qatar Computing Research Institute, Qatar Foundation

Tornado Tower, floor 10, PO box 5825

Doha, Qatar

{pnakov, fherrera, svogel}@qf.org.qa

## Abstract

While experimenting with tuning on long sentences, we made an unexpected discovery: that PRO falls victim to *monsters* – overly long negative examples with very low BLEU+1 scores, which are unsuitable for learning and can cause testing BLEU to drop by several points absolute. We propose several effective ways to address the problem, using length- and BLEU+1-based cut-offs, outlier filters, stochastic sampling, and random acceptance. The best of these fixes not only slay and protect against monsters, but also yield higher stability for PRO as well as improved test-time BLEU scores. Thus, we recommend them to anybody using PRO, monster-believer or not.

## 1 Once Upon a Time...

For years, the standard way to do statistical machine translation parameter tuning has been to use minimum error-rate training, or MERT (Och, 2003). However, as researchers started using models with thousands of parameters, new scalable optimization algorithms such as MIRA (Watanabe et al., 2007; Chiang et al., 2008) and PRO (Hopkins and May, 2011) have emerged. As these algorithms are relatively new, they are still not quite well understood, and studying their properties is an active area of research.

For example, Nakov et al. (2012) have pointed out that PRO tends to generate translations that are consistently shorter than desired. They have blamed this on inadequate smoothing in PRO’s optimization objective, namely sentence-level BLEU+1, and they have addressed the problem using more sensible smoothing. We wondered whether the issue could be partially relieved simply by tuning on longer sentences, for which the effect of smoothing would naturally be smaller.

To our surprise, tuning on the longer 50% of the tuning sentences had a disastrous effect on PRO, causing an absolute drop of three BLEU points on testing; at the same time, MERT and MIRA did not have such a problem. While investigating the reasons, we discovered hundreds of monsters creeping under PRO’s surface...

Our tale continues as follows. We first explain what monsters are in Section 2, then we present a theory about how they can be slayed in Section 3, we put this theory to test in practice in Section 4, and we discuss some related efforts in Section 5. Finally, we present the moral of our tale, and we hint at some planned future battles in Section 6.

## 2 Monsters, Inc.

PRO uses pairwise ranking optimization, where the learning task is to classify pairs of hypotheses into correctly or incorrectly ordered (Hopkins and May, 2011). It searches for a vector of weights  $w$  such that higher evaluation metric scores correspond to higher model scores and vice versa. More formally, PRO looks for weights  $w$  such that  $g(i, j) > g(i, j') \Leftrightarrow h_w(i, j) > h_w(i, j')$ , where  $g$  is a local scoring function (typically, sentence-level BLEU+1) and  $h_w$  are the model scores for a given input sentence  $i$  and two candidate hypotheses  $j$  and  $j'$  that were obtained using  $w$ . If  $g(i, j) > g(i, j')$ , we will refer to  $j$  and  $j'$  as the positive and the negative example in the pair.

Learning good parameter values requires negative examples that are comparable to the positive ones. Instead, tuning on long sentences quickly introduces *monsters*, i.e., corrupted negative examples that are unsuitable for learning: they are (i) much longer than the respective positive examples and the references, and (ii) have very low BLEU+1 scores compared to the positive examples and in absolute terms. The low BLEU+1 means that PRO effectively has to learn from positive examples only.

| iter. | Avg. Lengths |              |      | Avg. BLEU+1 |             |
|-------|--------------|--------------|------|-------------|-------------|
|       | pos          | neg          | ref. | pos         | neg         |
| 1     | 45.2         | 44.6         | 46.5 | 52.5        | 37.6        |
| 2     | 46.4         | 70.5         | 53.2 | 52.8        | 14.5        |
| 3     | 46.4         | <b>261.0</b> | 53.4 | 52.4        | <b>2.19</b> |
| 4     | 46.4         | <b>250.0</b> | 53.0 | 52.0        | <b>2.30</b> |
| 5     | 46.3         | <b>248.0</b> | 53.0 | 52.1        | <b>2.34</b> |
| ...   | ...          | ...          | ...  | ...         | ...         |
| 25    | 47.9         | <b>229.0</b> | 52.5 | 52.2        | <b>2.81</b> |

Table 1: PRO iterations, tuning on long sentences.

Table 1 shows an optimization run of PRO when tuning on long sentences. We can see monsters after iterations in which positive examples are on average longer than negative ones (e.g., iter. 1). As a result, PRO learns to generate longer sentences, but it overshoots too much (iter. 2), which gives rise to monsters. Ideally, the learning algorithm should be able to recover from overshooting. However, once monsters are encountered, they quickly start dominating, with no chance for PRO to recover since it accumulates  $n$ -best lists, and thus also monsters, over iterations. As a result, PRO keeps jumping up and down and converges to random values, as Figure 1 shows.

By default, PRO’s parameters are averaged over iterations, and thus the final result is quite mediocre, but selecting the highest tuning score does not solve the problem either: for example, on Figure 1, PRO never achieves a BLEU better than that for the default initialization parameters.

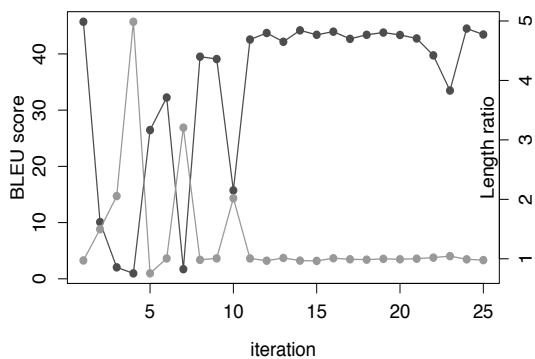


Figure 1: PRO tuning results on long sentences across iterations. The dark-gray line shows the tuning BLEU (left axis), the light-gray one is the hypothesis/reference length ratio (right axis).

Figure 2 shows the translations after iterations 1, 3 and 4; the last two are monsters. The monster at iteration 3 is potentially useful, but that at iteration 4 is clearly unsuitable as a negative example.

| Optimizer                | Objective                | BLEU         |
|--------------------------|--------------------------|--------------|
| PRO                      | sent-BLEU+1              | <b>44.57</b> |
| MERT                     | corpus-BLEU              | 47.53        |
| MIRA                     | pseudo-doc-BLEU          | 47.80        |
| PRO ( $\neq$ objective)  | pseudo-doc-BLEU          | <b>21.35</b> |
| MIRA ( $\neq$ objective) | sent-BLEU+1              | 47.59        |
| PRO, PC-smooth, ground   | <i>fixed</i> sent-BLEU+1 | <b>45.71</b> |

Table 2: PRO vs. MERT vs. MIRA.

We also checked whether other popular optimizers yield very low BLEU scores at test time when tuned on long sentences. Lines 2-3 in Table 2 show that this is not the case for MERT and MIRA. Since they optimize objectives that are different from PRO’s,<sup>1</sup> we further experimented with plugging MIRA’s objective into PRO and PRO’s objective into MIRA. The resulting MIRA scores were not much different from before, while PRO’s score dropped even further; we also found monsters. Next, we applied the length fix for PRO proposed in (Nakov et al., 2012); this helped a bit, but still left PRO two BLEU points behind MERT<sup>2</sup> and MIRA, and the monsters did not go away. We can conclude that the monster problem is PRO-specific, cannot be blamed on the objective function, and is different from the length bias.

Note also that monsters are not specific to a dataset or language pair. We found them when tuning on the top-50% of WMT10 and testing on WMT11 for Spanish-English; this yielded a drop in BLEU from 29.63 (MERT) to 27.12 (PRO).

\*\*REF\*\*:  
but we have to close ranks with each other and realize that in unity there is strength while in division there is weakness .  
-----  
\*\*IT1\*\*:  
but we are that we add our ranks to some of us and that we know that in the strength and weakness in  
  
\*\*IT3\*\*:  
, we are the but of the that that the , and , of ranks the the on the the our the our the some of we can include , and , of to the of we know the the our in of the of some people , force of that that the in of the that that the the weakness Union the the , and  
  
\*\*IT4\*\*:  
namely Dr Heba Handossah and Dr Mona been pushed aside because a larger story EU Ambassador to Egypt Ian Burg highlighted 've dragged us backwards and dragged our speaking , never balme your defaulting a December 7th 1941 in Pearl Harbor ) we can include ranks will be joined by all 've dragged us backwards and dragged our \$ 3.8 billion in tourism income proceeds Chamber are divided among themselves : some 've dragged us backwards and dragged our were exaggerated . Al @-@ Hakim namely Dr Heba Handossah and Dr Mona December 7th 1941 in Pearl Harbor ) cases might be known to us December 7th 1941 in Pearl Harbor ) platform depends on combating all liberal policies Track and Field Federation shortened strength as well face several challenges , namely Dr Heba Handossah and Dr Mona platform depends on combating all liberal policies the report forecast that the weak structure

Figure 2: Example reference translation and hypothesis translations after iterations 1, 3 and 4. The last two hypotheses are monsters.

<sup>1</sup>See (Cherry and Foster, 2012) for details on objectives.

<sup>2</sup>Also, using PRO to initialize MERT, as implemented in Moses, yields 46.52 BLEU and monsters, but using MERT to initialize PRO yields 47.55 and no monsters.

### 3 Slaying Monsters: Theory

Below we explain what monsters are and where they come from. Then, we propose various monster slaying techniques to be applied during PRO’s selection and acceptance steps.

#### 3.1 What is PRO?

PRO is a batch optimizer that iterates between (i) *translation*: using the current parameter values, generate  $k$ -best translations, and (ii) *optimization*: using the translations from all previous iterations, find new parameter values. The optimization step has four substeps:

1. **Sampling:** For each sentence, sample uniformly at random  $\Gamma = 5000$  pairs from the set of all candidate translations for that sentence from all previous iterations.
2. **Selection:** From these sampled pairs, select those for which the absolute difference between their BLEU+1 scores is higher than  $\alpha = 0.05$  (note: this is 5 BLEU+1 points).
3. **Acceptance:** For each sentence, accept the  $\Xi = 50$  selected pairs with the highest absolute difference in their BLEU+1 scores.
4. **Learning:** Assemble the accepted pairs for all sentences into a single set and use it to train a ranker to prefer the higher-scoring sentence in each pair.

We believe that monsters are nurtured by PRO’s selection and acceptance policies. PRO’s selection step filters pairs involving hypotheses that differ by less than five BLEU+1 points, but it does not cut-off ones that differ too much based on BLEU+1 or length. PRO’s acceptance step selects  $\Xi = 50$  pairs with the highest BLEU+1 differentials, which creates breeding ground for monsters since these pairs are very likely to include one monster and one good hypothesis.

Below we discuss monster slaying geared towards the selection and acceptance steps of PRO.

#### 3.2 Slaying at Selection

In the selection step, PRO filters pairs for which the difference in BLEU+1 is *less* than five points, but it has no cut-off on the *maximum* BLEU+1 differentials nor cut-offs based on absolute length or difference in *length*. Here, we propose several selection filters, both deterministic and probabilistic.

**Cut-offs.** A cut-off is a deterministic rule that filters out pairs that do not comply with some criteria. We experiment with a maximal cut-off on (a) the difference in BLEU+1 scores and (b) the difference in lengths. These are relative cut-offs because they refer to the pair, but absolute cut-offs that apply to each of the elements in the pair are also possible (not explored here). Cut-offs (a) and (b) slay monsters by not allowing the negative examples to get much worse in BLEU+1 or in length than the positive example in the pair.

**Filtering outliers.** Outliers are rare or extreme observations in a sample. We assume normal distribution of the BLEU+1 scores (or of the lengths) of the translation hypotheses for the same source sentence, and we define as outliers hypotheses whose BLEU+1 (or length) is more than  $\lambda$  standard deviations away from the sample average. We apply the outlier filter to both the positive and the negative example in a pair, but it is more important for the latter. We experiment with values of  $\lambda$  like 2 and 3. This filtering slays monsters because they are likely outliers. However, it will not work if the population gets riddled with monsters, in which case they would become the norm.

**Stochastic sampling.** Instead of filtering extreme examples, we can randomly sample pairs according to their probability of being typical. Let us assume that the values of the local scoring functions, i.e., the BLEU+1 scores, are distributed normally:  $g(i, j) \sim N(\mu, \sigma^2)$ . Given a sample of hypothesis translations  $\{j\}$  of the same source sentence  $i$ , we can estimate  $\sigma$  empirically. Then, the difference  $\Delta = g(i, j) - g(i, j')$  would be distributed normally with mean zero and variance  $2\sigma^2$ . Now, given a pair of examples, we can calculate their  $\Delta$ , and we can choose to select the pair with some probability, according to  $N(0, 2\sigma^2)$ .

#### 3.3 Slaying at Acceptance

Another problem is caused by the acceptance mechanism of PRO: among all selected pairs, it accepts the top- $\Xi$  with the highest BLEU+1 differentials. It is easy to see that these differentials are highest for nonmonster–monster pairs if such pairs exist. One way to avoid focusing primarily on such pairs is to accept a random set of  $\Xi$  pairs, among the ones that survived the selection step. One possible caveat is that we can lose some of the discriminative power of PRO by focusing on examples that are not different enough.

|                          | PRO fix                           | TESTING           |              | TUNING (run 1, it. 25, avg.) |              |      | TEST(tune:full) |            |                   |              |
|--------------------------|-----------------------------------|-------------------|--------------|------------------------------|--------------|------|-----------------|------------|-------------------|--------------|
|                          |                                   | Avg. for 3 reruns |              | Pos                          | Lengths      |      | BLEU+1          |            | Avg. for 3 reruns |              |
|                          |                                   | BLEU              | StdDev       |                              | Neg          | Ref  | Pos             | Neg        | BLEU              | StdDev       |
|                          | PRO (baseline)                    | 44.70             | 0.266        | 47.9                         | <b>229.0</b> | 52.5 | 52.2            | <b>2.8</b> | 47.80             | 0.052        |
| <b>Max diff. cut-off</b> | BLEU+1 max=10 <sup>†</sup>        | <b>47.94</b>      | <b>0.165</b> | 47.9                         | 49.6         | 49.4 | 49.4            | 39.9       | 47.77             | <b>0.035</b> |
|                          | BLEU+1 max=20 <sup>†</sup>        | 47.73             | <b>0.136</b> | 47.7                         | 55.5         | 51.1 | 49.8            | 32.7       | <b>47.85</b>      | <b>0.049</b> |
|                          | LEN max=5 <sup>†</sup>            | <b>48.09</b>      | <b>0.021</b> | 46.8                         | 47.0         | 47.9 | 52.9            | 37.8       | 47.73             | <b>0.051</b> |
|                          | LEN max=10 <sup>†</sup>           | <b>47.99</b>      | <b>0.025</b> | 47.3                         | 48.5         | 48.7 | 52.5            | 35.6       | <b>47.80</b>      | 0.056        |
| <b>Outliers</b>          | BLEU+1 $\lambda=2.0$ <sup>†</sup> | <b>48.05</b>      | <b>0.119</b> | 46.8                         | 47.2         | 47.7 | 52.2            | 39.5       | 47.47             | 0.090        |
|                          | BLEU+1 $\lambda=3.0$              | 47.12             | 1.348        | 47.6                         | <b>168.0</b> | 53.0 | 51.7            | <b>3.9</b> | 47.53             | <b>0.038</b> |
|                          | LEN $\lambda=2.0$                 | 46.68             | 2.005        | 49.3                         | <b>82.7</b>  | 53.1 | 52.3            | <b>5.3</b> | 47.49             | 0.085        |
|                          | LEN $\lambda=3.0$                 | 47.02             | 0.727        | 48.2                         | <b>163.0</b> | 51.4 | 51.4            | <b>4.2</b> | 47.65             | 0.096        |
| <b>Stoch. sampl.</b>     | $\Delta$ BLEU+1                   | 46.33             | 1.000        | 46.8                         | <b>216.0</b> | 53.3 | 53.1            | <b>2.4</b> | 47.74             | <b>0.035</b> |
|                          | $\Delta$ LEN                      | 46.36             | 1.281        | 47.4                         | <b>201.0</b> | 52.9 | 53.4            | <b>2.9</b> | 47.78             | 0.081        |

Table 3: Some fixes to PRO (select pairs with highest BLEU+1 differential, also require at least 5 BLEU+1 points difference). A dagger (<sup>†</sup>) indicates selection fixes that successfully get rid of monsters.

## 4 Attacking Monsters: Practice

Below, we first present our general experimental setup. Then, we present the results for the various selection alternatives, both with the original acceptance strategy and with random acceptance.

### 4.1 Experimental Setup

We used a phrase-based SMT model (Koehn et al., 2003) as implemented in the Moses toolkit (Koehn et al., 2007). We trained on all Arabic-English data for NIST 2012 except for UN, we tuned on (the longest-50% of) the MT06 sentences, and we tested on MT09. We used the MADA ATB segmentation for Arabic (Roth et al., 2008) and truecasing for English, phrases of maximal length 7, Kneser-Ney smoothing, and lexicalized reordering (Koehn et al., 2005), and a 5-gram language model, trained on GigaWord v.5 using KenLM (Heafield, 2011). We dropped unknown words both at tuning and testing, and we used minimum Bayes risk decoding at testing (Kumar and Byrne, 2004). We evaluated the output with NIST’s scoring tool v.13a, cased.

We used the Moses implementations of MERT, PRO and batch MIRA, with the `-return-best-dev` parameter for the latter. We ran these optimizers for up to 25 iterations and we used 1000-best lists.

For stability (Foster and Kuhn, 2009), we performed three reruns of each experiment (tuning + evaluation), and we report averaged scores.

### 4.2 Selection Alternatives

Table 3 presents the results for different selection alternatives. The first two columns show the testing results: average BLEU and standard deviation over three reruns.

The following five columns show statistics about the last iteration (it. 25) of PRO’s tuning for the worst rerun: average lengths of the positive and the negative examples and average effective reference length, followed by average BLEU+1 scores for the positive and the negative examples in the pairs. The last two columns present the results when tuning on the full tuning set. These are included to verify the behavior of PRO in a non-monster prone environment.

We can see in Table 3 that all selection mechanisms considerably improve BLEU compared to the baseline PRO, by 2-3 BLEU points. However, not every selection alternative gets rid of monsters, which can be seen by the large lengths and low BLEU+1 for the negative examples (in bold).

The max cut-offs for BLEU+1 and for lengths both slay the monsters, but the latter yields much lower standard deviation (thirteen times lower than for the baseline PRO!), thus considerably increasing PRO’s stability. On the full dataset, BLEU scores are about the same as for the original PRO (with small improvement for BLEU+1 max=20), but the standard deviations are slightly better.

Rejecting outliers using BLEU+1 and  $\lambda = 3$  is not strong enough to filter out monsters, but making this criterion more strict by setting  $\lambda = 2$ , yields competitive BLEU and kills the monsters.

Rejecting outliers based on length does not work as effectively though. We can think of two possible reasons: (i) lengths are not normally distributed, they are more Poisson-like, and (ii) the acceptance criterion is based on the top- $\Xi$  differentials based on BLEU+1, not based on length.

On the full dataset, rejecting outliers, BLEU+1 and length, yields lower BLEU and less stability.

|                      |                              | TESTING           |              | TUNING (run 1, it. 25, avg.) |              |       | TEST(tune:full) |            |                   |              |
|----------------------|------------------------------|-------------------|--------------|------------------------------|--------------|-------|-----------------|------------|-------------------|--------------|
| PRO fix              |                              | Avg. for 3 reruns |              | Lengths                      |              |       | BLEU+1          |            | Avg. for 3 reruns |              |
|                      |                              | BLEU              | StdDev       | Pos                          | Neg          | Ref   | Pos             | Neg        | BLEU              | StdDev       |
|                      | PRO (baseline)               | 44.70             | 0.266        | 47.9                         | <b>229.0</b> | 52.5  | 52.2            | <b>2.8</b> | 47.80             | 0.052        |
| <b>Rand. accept</b>  | PRO, rand <sup>††</sup>      | 47.87             | 0.147        | 47.7                         | 48.5         | 48.70 | 47.7            | 42.9       | 47.59             | 0.114        |
| <b>Outliers</b>      | BLEU+1 $\lambda=2.0$ , rand* | 47.85             | <b>0.078</b> | 48.2                         | 48.4         | 48.9  | 47.5            | 43.6       | <b>47.62</b>      | <b>0.091</b> |
|                      | BLEU+1 $\lambda=3.0$ , rand  | <b>47.97</b>      | <b>0.168</b> | 47.6                         | 47.6         | 48.4  | 47.8            | 43.6       | 47.44             | <b>0.070</b> |
|                      | LEN $\lambda=2.0$ , rand*    | 47.69             | <b>0.114</b> | 47.8                         | 47.8         | 48.6  | 47.9            | 43.6       | 47.48             | <b>0.046</b> |
|                      | LEN $\lambda=3.0$ , rand     | 47.89             | 0.235        | 47.8                         | 48.0         | 48.7  | 47.7            | 43.1       | <b>47.64</b>      | <b>0.090</b> |
| <b>Stoch. sampl.</b> | $\Delta$ BLEU+1, rand*       | <b>47.99</b>      | <b>0.087</b> | 47.9                         | 48.0         | 48.7  | 47.8            | 43.5       | <b>47.67</b>      | <b>0.096</b> |
|                      | $\Delta$ LEN, rand*          | <b>47.94</b>      | <b>0.060</b> | 47.8                         | 47.9         | 48.6  | 47.8            | 43.6       | <b>47.65</b>      | <b>0.097</b> |

Table 4: More fixes to PRO (with random acceptance, no minimum BLEU+1). The (<sup>††</sup>) indicates that random acceptance kills monsters. The asterisk (\*) indicates improved stability over random acceptance.

Reasons (i) and (ii) arguably also apply to stochastic sampling of differentials (for BLEU+1 or for length), which fails to kill the monsters, maybe because it gives them some probability of being selected by design. To alleviate this, we test the above settings with random acceptance.

### 4.3 Random Acceptance

Table 4 shows the results for accepting training pairs for PRO uniformly at random. To eliminate possible biases, we also removed the  $\min=0.05$  BLEU+1 selection criterion. Surprisingly, this setup effectively eliminated the monster problem. Further coupling this with the distributional criteria can also yield increased stability, and even small further increase in test BLEU. For instance, rejecting BLEU outliers with  $\lambda = 2$  yields comparable average test BLEU, but with only half the standard deviation.

On the other hand, using the stochastic sampling of differentials based on either BLEU+1 or lengths improves the test BLEU score while increasing the stability across runs. The random acceptance has a caveat though: it generally decreases the discriminative power of PRO, yielding worse results when tuning on the full, nonmonster prone tuning dataset. Stochastic selection does help to alleviate this problem. Yet, the results are not as good as when using a max cut-off for the length. Therefore, we recommend using the latter as a default setting.

## 5 Related Work

We are not aware of previous work that discusses the issue of monsters, but there has been work on a different, length problem with PRO (Nakov et al., 2012). We have seen that its solution, fix the smoothing in BLEU+1, did not work for us.

The stability of MERT has been improved using regularization (Cer et al., 2008), random restarts (Moore and Quirk, 2008), multiple replications (Clark et al., 2011), and parameter aggregation (Cettolo et al., 2011).

With the emergence of new optimization techniques, there have been studies that compare stability between MIRA–MERT (Chiang et al., 2008; Chiang et al., 2009; Cherry and Foster, 2012), PRO–MERT (Hopkins and May, 2011), MIRA–PRO–MERT (Cherry and Foster, 2012; Gimpel and Smith, 2012; Nakov et al., 2012).

Pathological verbosity can be an issue when tuning MERT on recall-oriented metrics such as METEOR (Lavie and Denkowski, 2009; Denkowski and Lavie, 2011). Large variance between the results obtained with MIRA has also been reported (Simianer et al., 2012). However, none of this work has focused on monsters.

## 6 Tale’s Moral and Future Battles

We have studied a problem with PRO, namely that it can fall victim to monsters, overly long negative examples with very low BLEU+1 scores, which are unsuitable for learning. We have proposed several effective ways to address this problem, based on length- and BLEU+1-based cut-offs, outlier filters and stochastic sampling. The best of these fixes have not only slayed the monsters, but have also brought much higher stability to PRO as well as improved test-time BLEU scores. These benefits are less visible on the full dataset, but we still recommend them to everybody who uses PRO as protection against monsters. Monsters are inherent in PRO; they just do not always take over.

In future work, we plan a deeper look at the mechanism of monster creation in PRO and its possible connection to PRO’s length bias.



## References

- Daniel Cer, Daniel Jurafsky, and Christopher Manning. 2008. Regularization and search for minimum error rate training. In *Proc. of Workshop on Statistical Machine Translation*, WMT '08, pages 26–34.
- Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. 2011. Methods for smoothing the optimizer instability in SMT. *MT Summit XIII: the Machine Translation Summit*, pages 32–39.
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '12, pages 427–436.
- David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 224–233.
- David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '09, pages 218–226.
- Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the Meeting of the Association for Computational Linguistics*, ACL '11, pages 176–181.
- Michael Denkowski and Alon Lavie. 2011. Meteor-tuned phrase-based SMT: CMU French-English and Haitian-English systems for WMT 2011. Technical report, CMU-LTI-11-011, Language Technologies Institute, Carnegie Mellon University.
- George Foster and Roland Kuhn. 2009. Stabilizing minimum error rate training. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '09, pages 242–249.
- Kevin Gimpel and Noah Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '12, pages 221–231.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Workshop on Statistical Machine Translation*, WMT '11, pages 187–197.
- Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, HLT-NAACL '03, pages 48–54.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT '05.
- 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 *Proc. of the Meeting of the Association for Computational Linguistics*, ACL '07, pages 177–180.
- Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, HLT-NAACL '04, pages 169–176.
- Alon Lavie and Michael Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115.
- Robert Moore and Chris Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics*, COLING '08, pages 585–592.
- Preslav Nakov, Francisco Guzmán, and Stephan Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *Proceedings of the International Conference on Computational Linguistics*, COLING '12, pages 1979–1994.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Meeting of the Association for Computational Linguistics*, ACL '03, pages 160–167.
- Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the Meeting of the Association for Computational Linguistics*, ACL '08, pages 117–120.
- Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the Meeting of the Association for Computational Linguistics*, ACL '12, pages 11–21.
- Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 764–773.

# Supervised Model Learning with Feature Grouping based on a Discrete Constraint

Jun Suzuki and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation  
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan  
{suzuki.jun, nagata.masaaki}@lab.ntt.co.jp

## Abstract

This paper proposes a framework of supervised model learning that realizes feature grouping to obtain lower complexity models. The main idea of our method is to integrate a discrete constraint into model learning with the help of the dual decomposition technique. Experiments on two well-studied NLP tasks, dependency parsing and NER, demonstrate that our method can provide state-of-the-art performance even if the degrees of freedom in trained models are surprisingly small, *i.e.*, 8 or even 2. This significant benefit enables us to provide compact model representation, which is especially useful in actual use.

## 1 Introduction

This paper focuses on the topic of supervised model learning, which is typically represented as the following form of the optimization problem:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \{ \mathcal{O}(\mathbf{w}; \mathcal{D}) \}, \quad (1)$$
$$\mathcal{O}(\mathbf{w}; \mathcal{D}) = \mathcal{L}(\mathbf{w}; \mathcal{D}) + \Omega(\mathbf{w}),$$

where  $\mathcal{D}$  is supervised training data that consists of the corresponding input  $\mathbf{x}$  and output  $\mathbf{y}$  pairs, that is,  $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$ .  $\mathbf{w}$  is an  $N$ -dimensional vector representation of a set of optimization variables, which are also interpreted as *feature weights*.  $\mathcal{L}(\mathbf{w}; \mathcal{D})$  and  $\Omega(\mathbf{w})$  represent a loss function and a regularization term, respectively. Nowadays, we, in most cases, utilize a supervised learning method expressed as the above optimization problem to estimate the feature weights of many natural language processing (NLP) tasks, such as text classification, POS-tagging, named entity recognition, dependency parsing, and semantic role labeling.

In the last decade, the  $L_1$ -regularization technique, which incorporates  $L_1$ -norm into  $\Omega(\mathbf{w})$ , has become popular and widely-used in many NLP tasks (Gao et al., 2007; Tsuruoka et al.,

2009). The reason is that  $L_1$ -regularizers encourage feature weights to be zero as much as possible in model learning, which makes the resultant model a sparse solution (many zero-weights exist). We can discard all features whose weight is zero from the *trained model*<sup>1</sup> without any loss. Therefore,  $L_1$ -regularizers have the ability to easily and automatically yield *compact* models without strong concern over feature selection.

Compact models generally have significant and clear advantages in practice: instances are faster loading speed to memory, less memory occupation, and even faster decoding is possible if the model is small enough to be stored in cache memory. Given this background, our aim is to establish a model learning framework that can reduce the model complexity beyond that possible by simply applying  $L_1$ -regularizers. To achieve our goal, we focus on the recently developed concept of automatic feature grouping (Tibshirani et al., 2005; Bondell and Reich, 2008). We introduce a model learning framework that achieves feature grouping by incorporating a discrete constraint during model learning.

## 2 Feature Grouping Concept

Going beyond  $L_1$ -regularized sparse modeling, the idea of ‘*automatic feature grouping*’ has recently been developed. Examples are fused lasso (Tibshirani et al., 2005), grouping pursuit (Shen and Huang, 2010), and OSCAR (Bondell and Reich, 2008). The concept of automatic feature grouping is to find accurate models that have fewer degrees of freedom. This is equivalent to enforce every optimization variables to be equal as much as possible. A simple example is that  $\hat{\mathbf{w}}_1 = (0.1, 0.5, 0.1, 0.5, 0.1)$  is preferred over  $\hat{\mathbf{w}}_2 = (0.1, 0.3, 0.2, 0.5, 0.3)$  since  $\hat{\mathbf{w}}_1$  and  $\hat{\mathbf{w}}_2$  have two and four unique values, respectively.

There are several merits to reducing the degree

<sup>1</sup>This paper refers to model after completion of (supervised) model learning as “trained model”

of freedom. For example, previous studies clarified that it can reduce the chance of over-fitting to the training data (Shen and Huang, 2010). This is an important property for many NLP tasks since they are often modeled with a high-dimensional feature space, and thus, the over-fitting problem is readily triggered. It has also been reported that it can improve the stability of selecting non-zero features beyond that possible with the standard  $L_1$ -regularizer given the existence of many highly correlated features (Jörnsten and Yu, 2003; Zou and Hastie, 2005). Moreover, it can dramatically reduce model complexity. This is because we can merge all features whose feature weight values are equivalent in the trained model into a single feature cluster without any loss.

### 3 Modeling with Feature Grouping

This section describes our proposal for obtaining a feature grouping solution.

#### 3.1 Integration of a Discrete Constraint

Let  $\mathcal{S}$  be a finite set of discrete values, *i.e.*, a set integer from  $-4$  to  $4$ , that is,  $\mathcal{S} = \{-4, \dots, -1, 0, 1, \dots, 4\}$ . The detailed discussion how we define  $\mathcal{S}$  can be found in our experiments section since it deeply depends on training data. Then, we define the objective that can simultaneously achieve a feature grouping and model learning as follows:

$$\begin{aligned} \mathcal{O}(\mathbf{w}; \mathcal{D}) &= \mathcal{L}(\mathbf{w}; \mathcal{D}) + \Omega(\mathbf{w}) \\ \text{s.t. } \mathbf{w} &\in \mathcal{S}^N. \end{aligned} \quad (2)$$

where  $\mathcal{S}^N$  is the cartesian power of a set  $\mathcal{S}$ . The only difference with Eq. 1 is the additional discrete constraint, namely,  $\mathbf{w} \in \mathcal{S}^N$ . This constraint means that each variable (feature weight) in trained models must take a value in  $\mathcal{S}$ , that is,  $\hat{w}_n \in \mathcal{S}$ , where  $\hat{w}_n$  is the  $n$ -th factor of  $\hat{\mathbf{w}}$ , and  $n \in \{1, \dots, N\}$ . As a result, feature weights in trained models are automatically grouped in terms of the basis of model learning. This is the basic idea of feature grouping proposed in this paper.

However, a concern is how we can efficiently optimize Eq. 2 since it involves a NP-hard combinatorial optimization problem. The time complexity of the direct optimization is exponential against  $N$ . Next section introduces a feasible algorithm.

#### 3.2 Dual Decomposition Formulation

Hereafter, we strictly assume that  $\mathcal{L}(\mathbf{w}; \mathcal{D})$  and  $\Omega(\mathbf{w})$  are both convex in  $\mathbf{w}$ . Then, the properties of our method are unaffected by the selection

of  $\mathcal{L}(\mathbf{w}; \mathcal{D})$  and  $\Omega(\mathbf{w})$ . Thus, we ignore their specific definition in this section. Typical cases can be found in the experiments section. Then, we reformulate Eq. 2 by using the dual decomposition technique (Everett, 1963):

$$\begin{aligned} \mathcal{O}(\mathbf{w}, \mathbf{u}; \mathcal{D}) &= \mathcal{L}(\mathbf{w}; \mathcal{D}) + \Omega(\mathbf{w}) + \Upsilon(\mathbf{u}) \\ \text{s.t. } \mathbf{w} &= \mathbf{u}, \text{ and } \mathbf{u} \in \mathcal{S}^N. \end{aligned} \quad (3)$$

Difference from Eq. 2, Eq. 3 has an additional term  $\Upsilon(\mathbf{u})$ , which is similar to the regularizer  $\Omega(\mathbf{w})$ , whose optimization variables  $\mathbf{w}$  and  $\mathbf{u}$  are tightened with equality constraint  $\mathbf{w} = \mathbf{u}$ . Here, this paper only considers the case  $\Upsilon(\mathbf{u}) = \frac{\lambda_2}{2} \|\mathbf{u}\|_2^2 + \lambda_1 \|\mathbf{u}\|_1$ , and  $\lambda_2 \geq 0$  and  $\lambda_1 \geq 0^2$ . This objective can also be viewed as the decomposition of the standard loss minimization problem shown in Eq. 1 and the additional discrete constraint regularizer by the dual decomposition technique.

To solve the optimization in Eq. 3, we leverage the *alternating direction method of multiplier* (ADMM) (Gabay and Mercier, 1976; Boyd et al., 2011). ADMM provides a very efficient optimization framework for the problem in the dual decomposition form. Here,  $\alpha$  represents dual variables for the equivalence constraint  $\mathbf{w} = \mathbf{u}$ . ADMM introduces the augmented Lagrangian term  $\frac{\rho}{2} \|\mathbf{w} - \mathbf{u}\|_2^2$  with  $\rho > 0$  which ensures strict convexity and increases robustness<sup>3</sup>.

Finally, the optimization problem in Eq. 3 can be converted into a series of iterative optimization problems. Detailed derivation in the general case can be found in (Boyd et al., 2011). Fig. 1 shows the entire model learning framework of our proposed method. The remarkable point is that ADMM works by iteratively computing one of the three optimization variable sets  $\mathbf{w}$ ,  $\mathbf{u}$ , and  $\alpha$  while holding the other variables fixed in the iterations  $t = 1, 2, \dots$  until convergence.

**Step1 (w-update):** This part of the optimization problem shown in Eq. 4 is essentially Eq. 1 with a ‘biased’  $L_2$ -regularizer. ‘bias’ means here that the direction of regularization is toward point  $\mathbf{a}$  instead of the origin. Note that it becomes a standard  $L_2$ -regularizer if  $\mathbf{a} = \mathbf{0}$ . We can select any learning algorithm that can handle the  $L_2$ -regularizer for this part of the optimization.

**Step2 (u-update):** This part of the optimization problem shown in Eq. 5 can be rewritten in the

<sup>2</sup>Note that this setting includes the use of only  $L_1$ -,  $L_2$ -, or without regularizers ( $L_1$  only:  $\lambda_1 > 0$  and  $\lambda_2 = 0$ ,  $L_2$  only:  $\lambda_1 = 0$  and  $\lambda_2 > 0$ , and without regularizer:  $\lambda_1 = 0$ ,  $\lambda_2 = 0$ ).

<sup>3</sup>Standard dual decomposition can be viewed as  $\rho = 0$

---

**Input:** Training data:  $\mathcal{D}$ , parameters:  $\rho, \xi, \epsilon_{\text{primal}}$ , and  $\epsilon_{\text{dual}}$

**Initialize:**  $\mathbf{w}^{(1)} = \mathbf{0}, \mathbf{u}^{(1)} = \mathbf{0}, \boldsymbol{\alpha}^{(1)} = \mathbf{0}$ , and  $t = 1$ .

**Step1 w-update:**

Solve  $\mathbf{w}^{(t+1)} = \arg \min_{\mathbf{w}} \{\mathcal{O}(\mathbf{w}; \mathcal{D}, \mathbf{u}^{(t)}, \boldsymbol{\alpha}^{(t)})\}$ .

For our case,

$$\mathcal{O}(\mathbf{w}; \mathcal{D}, \mathbf{u}, \boldsymbol{\alpha}) = \mathcal{O}(\mathbf{w}; \mathcal{D}) + \frac{\rho}{2} \|\mathbf{w} - \mathbf{a}\|_2^2, \quad (4)$$

where  $\mathbf{a} = \mathbf{u} - \boldsymbol{\alpha}$ .

**Step2 u-update:**

Solve  $\mathbf{u}^{(t+1)} = \arg \min_{\mathbf{u}} \{\mathcal{O}(\mathbf{u}; \mathcal{D}, \mathbf{w}^{(t+1)}, \boldsymbol{\alpha}^{(t)})\}$ .

For our case,

$$\begin{aligned} \mathcal{O}(\mathbf{u}; \mathcal{D}, \mathbf{w}, \boldsymbol{\alpha}) &= \frac{\lambda_2}{2} \|\mathbf{u}\|_2^2 + \lambda_1 \|\mathbf{u}\|_1 + \frac{\rho}{2} \|\mathbf{b} - \mathbf{u}\|_2^2 \\ \text{s.t. } &\mathbf{u} \in \mathcal{S}^N, \end{aligned} \quad (5)$$

where  $\mathbf{b} = \mathbf{w} + \boldsymbol{\alpha}$

**Step3  $\alpha$ -update:**

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + \xi(\mathbf{w}^{(t+1)} - \mathbf{u}^{(t+1)}) \quad (6)$$

**Step4 convergence check:**

$$\begin{aligned} \|\mathbf{w}^{(t+1)} - \mathbf{u}^{(t+1)}\|_2^2 / N &< \epsilon_{\text{primal}} \\ \|\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}\|_2^2 / N &< \epsilon_{\text{dual}} \end{aligned} \quad (7)$$

Break the loop if the above two conditions are reached, or go back to Step1 with  $t = t + 1$ .

**Output:**  $\mathbf{u}^{(t+1)}$

---

Figure 1: Entire learning framework of our method derived from ADMM (Boyd et al., 2011).

following equivalent simple form:

$$\begin{aligned} \hat{\mathbf{u}} &= \arg \min_{\mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{u} - \mathbf{b}'\|_2^2 + \lambda'_1 \|\mathbf{u}\|_1 \right\} \\ \text{s.t. } &\mathbf{u} \in \mathcal{S}^N, \end{aligned} \quad (8)$$

where  $\mathbf{b}' = \frac{\rho}{\lambda_2 + \rho} \mathbf{b}$ , and  $\lambda'_1 = \frac{\lambda_1}{\lambda_2 + \rho}$ . This optimization is still a combinatorial optimization problem. However unlike Eq. 2, this optimization can be efficiently solved.

Fig. 2 shows the procedure to obtain the *exact* solution of Eq. 5, namely  $\mathbf{u}^{(t+1)}$ . The remarkable point is that the costly combinatorial optimization problem is disappeared, and instead, we are only required to perform two feature-wise calculations whose total time complexities is  $O(N \log |\mathcal{S}|)$  and fully parallelizable. The similar technique has been introduced in Zhong and Kwok (2011) for discarding a costly combinatorial problem from the optimization with OSCAR-regularizers with the help of proximal gradient methods, *i.e.*, (Beck and Teboulle, 2009).

We omit to show the detailed derivation of Fig. 2 because of the space reason. However, this is easily understandable. The key properties are the following two folds; (i) The objective shown in Eq. 8 is a convex and also symmetric function with respect to  $\hat{\mathbf{u}}'$ , where  $\hat{\mathbf{u}}'$  is the optimal solution of Eq. 8 without the discrete constraint. Therefore, the optimal solution  $\hat{\mathbf{u}}$  is at the point where the

---

**Input:**  $\mathbf{b}' = (b'_n)_{n=1}^N, \lambda'_1$ , and  $\mathcal{S}$ .

1, Find the optimal solution of Eq. 8 without the constraint.

The optimization of mixed  $L_2$  and  $L_1$ -norms is known to have a closed form solution, *i.e.*, (Beck and Teboulle, 2009), that is;

$$\hat{u}'_n = \text{sgn}(b'_n) \max(0, |b'_n| - \lambda'_1),$$

where  $(\hat{u}'_n)_{n=1}^N = \hat{\mathbf{u}}'$ .

2, Find the nearest valid point in  $\mathcal{S}^N$  from  $\hat{\mathbf{u}}'$  in terms of the  $L_2$ -distance;

$$\hat{u}_n = \arg \min_{u \in \mathcal{S}} (\hat{u}'_n - u)^2$$

where  $(\hat{u}_n)_{n=1}^N = \hat{\mathbf{u}}$ . This can be performed by a binary search, whose time complexity is generally  $O(\log |\mathcal{S}|)$ .

**Output:**  $\hat{\mathbf{u}}$

---

Figure 2: Procedure for solving Step2

nearest valid point given  $\mathcal{S}^N$  from  $\hat{\mathbf{u}}'$  in terms of the  $L_2$ -distance. (ii) The valid points given  $\mathcal{S}^N$  are always located at the vertexes of *axis-aligned orthotopes (hyperrectangles)* in the parameter space of feature weights. Thus, the solution  $\hat{\mathbf{u}}$ , which is the nearest valid point from  $\hat{\mathbf{u}}'$ , can be obtained by individually taking the nearest value in  $\mathcal{S}$  from  $\hat{u}'_n$  for all  $n$ .

**Step3 ( $\alpha$ -update):** We perform gradient ascent on dual variables to tighten the constraint  $\mathbf{w} = \mathbf{u}$ . Note that  $\xi$  is the learning rate; we can simply set it to 1.0 for every iteration (Boyd et al., 2011).

**Step4 (convergence check):** It can be evaluated both primal and dual residuals as defined in Eq. 7 with suitably small  $\epsilon_{\text{primal}}$  and  $\epsilon_{\text{dual}}$ .

### 3.3 Online Learning

We can select an online learning algorithm for Step1 since the ADMM framework does not require exact minimization of Eq. 4. In this case, we perform one-pass update through the data in each ADMM iteration (Duh et al., 2011). Note that the total calculation cost of our method does not increase much from original online learning algorithm since the calculation cost of Steps 2 through 4 is relatively much smaller than that of Step1.

## 4 Experiments

We conducted experiments on two well-studied NLP tasks, namely named entity recognition (NER) and dependency parsing (DEPAR).

**Basic settings:** We simply reused the settings of most previous studies. We used CoNLL'03 data (Tjong Kim Sang and De Meulder, 2003) for NER, and the Penn Treebank (PTB) III corpus (Marcus et al., 1994) converted to dependency trees for DEPAR (McDonald et al., 2005).

Our decoding models are the Viterbi algorithm on CRF (Lafferty et al., 2001), and the second-order parsing model proposed by (Carreras, 2007) for NER and DEPAR, respectively. Features are automatically generated according to the pre-defined feature templates widely-used in the previous studies. We also integrated the cluster features obtained by the method explained in (Koo et al., 2008) as additional features for evaluating our method in the range of the current best systems.

**Evaluation measures:** The purpose of our experiments is to investigate the effectiveness of our proposed method in terms of both its performance and the complexity of the trained model. Therefore, our evaluation measures consist of two axes. Task performance was mainly evaluated in terms of the complete sentence accuracy (**COMP**) since the objective of all model learning methods evaluated in our experiments is to maximize COMP. We also report the  $F_{\beta=1}$  score (**F-sc**) for NER, and the unlabeled attachment score (**UAS**) for DEPAR for comparison with previous studies. Model complexity is evaluated by the number of non-zero active features (**#nzF**) and the degree of freedom (**#DoF**) (Zhong and Kwok, 2011). #nzF is the number of features whose corresponding feature weight is non-zero in the trained model, and #DoF is the number of unique non-zero feature weights.

**Baseline methods:** Our main baseline is  $L_1$ -regularized sparse modeling. To cover both batch and online learning, we selected  $L_1$ -regularized CRF (**L1CRF**) (Lafferty et al., 2001) optimized by OWL-QN (Andrew and Gao, 2007) for the NER experiment, and the  $L_1$ -regularized *regularized dual averaging* (**L1RDA**) method (Xiao, 2010)<sup>4</sup> for DEPAR. Additionally, we also evaluated  $L_2$ -regularized CRF (**L2CRF**) with L-BFGS (Liu and Nocedal, 1989) for NER, and passive-aggressive algorithm (**L2PA**) (Crammer et al., 2006)<sup>5</sup> for DEPAR since  $L_2$ -regularizer often provides better results than  $L_1$ -regularizer (Gao et al., 2007).

For a fair comparison, we applied the procedure of Step2 as a simple quantization method to trained models obtained from  $L_1$ -regularized model learning, which we refer to as (**QT**).

<sup>4</sup>RDA provided better results at least in our experiments than  $L_1$ -regularized FOBOS (Duchi and Singer, 2009), and its variant (Tsuruoka et al., 2009), which are more familiar to the NLP community.

<sup>5</sup>L2PA is also known as a loss augmented variant of one-best MIRA, well-known in DEPAR (McDonald et al., 2005).

## 4.1 Configurations of Our Method

**Base learning algorithm:** The settings of our method in our experiments imitate  $L_1$ -regularized learning algorithm since the purpose of our experiments is to investigate the effectiveness against standard  $L_1$ -regularized learning algorithms. Then, we have the following two possible settings; **DC-ADMM:** we leveraged the baseline  $L_1$ -regularized learning algorithm to solve Step1, and set  $\lambda_1 = 0$  and  $\lambda_2 = 0$  for Step2. **DCwL1-ADMM:** we leveraged the baseline  $L_2$ -regularized learning algorithm, but without  $L_2$ -regularizer, to solve Step1, and set  $\lambda_1 > 0$  and  $\lambda_2 = 0$  for Step2. The difference can be found in the objective function  $\mathcal{O}(\mathbf{w}, \mathbf{u}; \mathcal{D})$  shown in Eq. 3;

$$\begin{aligned} (\text{DC-ADMM}): \mathcal{O}(\mathbf{w}, \mathbf{u}; \mathcal{D}) &= \mathcal{L}(\mathbf{w}; \mathcal{D}) + \lambda_1 \|\mathbf{w}\|_1 \\ (\text{DCwL1-ADMM}): \mathcal{O}(\mathbf{w}, \mathbf{u}; \mathcal{D}) &= \mathcal{L}(\mathbf{w}; \mathcal{D}) + \lambda_1 \|\mathbf{u}\|_1 \end{aligned}$$

In other words, DC-ADMM utilizes  $L_1$ -regularizer as a part of base learning algorithm  $\Omega(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1$ , while DCwL1-ADMM discards regularizer of base learning algorithm  $\Omega(\mathbf{w})$ , but instead introducing  $\Upsilon(\mathbf{u}) = \lambda_1 \|\mathbf{u}\|_1$ . Note that these two configurations are essentially identical since objectives are identical, even though the formulation and algorithm is different. We only report results of DC-ADMM because of the space reason since the results of DCwL1-ADMM were nearly equivalent to those of DC-ADMM.

**Definition of  $\mathcal{S}$ :** DC-ADMM can utilize any finite set for  $\mathcal{S}$ . However, we have to carefully select it since it deeply affects the performance. Actually, this is the most considerable point of our method. We preliminarily investigated the several settings. Here, we introduce an example of template which is suitable for large feature set. Let  $\eta$ ,  $\delta$ , and  $\kappa$  represent non-negative real-value constants,  $\zeta$  be a positive integer,  $\sigma = \{-1, 1\}$ , and a function  $f_{\eta, \delta, \kappa}(x, y) = y(\eta\kappa^x + \delta)$ . Then, we define a finite set of values  $\mathcal{S}$  as follows:

$$\mathcal{S}_{\eta, \delta, \kappa, \zeta} = \{f_{\eta, \delta, \kappa}(x, y) | (x, y) \in \mathcal{S}_{\zeta} \times \sigma\} \cup \{0\},$$

where  $\mathcal{S}_{\zeta}$  is a set of non-negative integers from zero to  $\zeta - 1$ , that is,  $\mathcal{S}_{\zeta} = \{m\}_{m=0}^{\zeta-1}$ . For example, if we set  $\eta = 0.1$ ,  $\delta = 0.4$ ,  $\kappa = 4$ , and  $\zeta = 3$ , then  $\mathcal{S}_{\eta, \delta, \kappa, \zeta} = \{-2.0, -0.8, -0.5, 0, 0.5, 0.8, 2.0\}$ . The intuition of this template is that the distribution of the feature weights in trained model often takes a form a similar to that of the ‘power law’ in the case of the large feature sets. Therefore, using an exponential function with a scale and bias seems to be appropriate for fitting them.

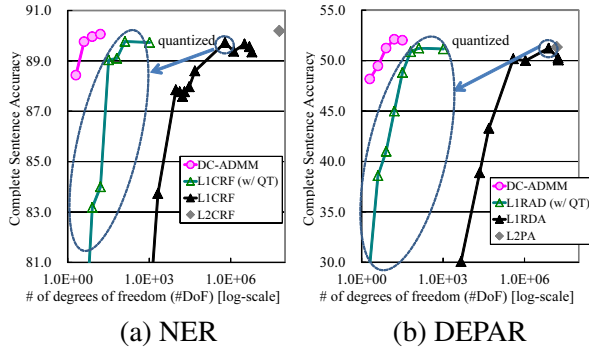


Figure 3: Performance vs. degree of freedom in the trained model for the development data

Note that we can control the upper bound of #DoF in trained model by  $\zeta$ , namely if  $\zeta = 4$  then the upper bound of #DoF is 8 (doubled by positive and negative sides). We fixed  $\rho = 1$ ,  $\xi = 1$ ,  $\lambda_2 = 0$ ,  $\kappa = 4$  (or 2 if  $\zeta \geq 5$ ),  $\delta = \eta/2$  in all experiments. Thus the only tunable parameter in our experiments is  $\eta$  for each  $\zeta$ .

## 4.2 Results and Discussions

Fig. 3 shows the task performance on the development data against the model complexities in terms of the degrees of freedom in the trained models. Plots are given by changing the  $\zeta$  value for DC-ADMM and  $L_1$ -regularized methods with QT. The plots of the standard  $L_1$ -regularized methods are given by changing the regularization constants  $\lambda_1$ . Moreover, Table 1 shows the final results of our experiments on the test data. The tunable parameters were fixed at values that provided the best performance on the development data.

According to the figure and table, the most remarkable point is that DC-ADMM successfully maintained the task performance even if #DoF (the degree of freedom) was 8, and the performance drop-offs were surprisingly limited even if #DoF was 2, which is the upper bound of feature grouping. Moreover, it is worth noting that the DC-ADMM performance is sometimes improved. The reason may be that such low degrees of freedom prevent over-fitting to the training data. Surprisingly, the simple quantization method (QT) provided fairly good results. However, we emphasize that the models produced by the QT approach offer no guarantee as to the optimal solution. In contrast, DC-ADMM can truly provide the optimal solution of Eq. 3 since the discrete constraint is also considered during the model learning.

In general, a trained model consists of two parts:

| NER                     | Test  |       | Model complex. |       |
|-------------------------|-------|-------|----------------|-------|
|                         | COMP  | F-sc  | #nzF           | #DoF  |
| L2CRF                   | 84.88 | 89.97 | 61.6M          | 38.6M |
| L1CRF                   | 84.85 | 89.99 | 614K           | 321K  |
| (w/ QT $\zeta = 4$ )    | 78.39 | 85.33 | 568K           | 8     |
| (w/ QT $\zeta = 2$ )    | 73.40 | 81.45 | 454K           | 4     |
| (w/ QT $\zeta = 1$ )    | 65.53 | 75.87 | 454K           | 2     |
| DC-ADMM ( $\zeta = 4$ ) | 84.96 | 89.92 | 643K           | 8     |
| ( $\zeta = 2$ )         | 84.04 | 89.35 | 455K           | 4     |
| ( $\zeta = 1$ )         | 83.06 | 88.62 | 364K           | 2     |

| DEPER                   | Test  |       | Model complex. |       |
|-------------------------|-------|-------|----------------|-------|
|                         | COMP  | UAS   | #nzF           | #DoF  |
| L2PA                    | 49.67 | 93.51 | 15.5M          | 5.59M |
| L1RDA                   | 49.54 | 93.48 | 7.76M          | 3.56M |
| (w/ QT $\zeta = 4$ )    | 38.58 | 90.85 | 6.32M          | 8     |
| (w/ QT $\zeta = 2$ )    | 34.19 | 89.42 | 3.08M          | 4     |
| (w/ QT $\zeta = 1$ )    | 30.42 | 88.67 | 3.08M          | 2     |
| DC-ADMM ( $\zeta = 4$ ) | 49.83 | 93.55 | 5.81M          | 8     |
| ( $\zeta = 2$ )         | 48.97 | 93.18 | 4.11M          | 4     |
| ( $\zeta = 1$ )         | 46.56 | 92.86 | 6.37M          | 2     |

Table 1: Comparison results of the methods on test data (K: thousand, M: million)

feature weights and an indexed structure of feature strings, which are used as the key for obtaining the corresponding feature weight. This paper mainly discussed how to reduce the size of the former part, and described its successful reduction. We note that it is also possible to reduce the latter part especially if the feature string structure is TRIE. We omit the details here since it is not the main topic of this paper, but by merging feature strings that have the same feature weights, the size of entire trained models in our DEPAR case can be reduced to about 10 times smaller than those obtained by standard  $L_1$ -regularization, *i.e.*, to 12.2 MB from 124.5 MB.

## 5 Conclusion

This paper proposed a model learning framework that can simultaneously realize feature grouping by the incorporation of a simple discrete constraint into model learning optimization. This paper also introduced a feasible algorithm, DC-ADMM, which can vanish the infeasible combinatorial optimization part from the entire learning algorithm with the help of the ADMM technique. Experiments showed that DC-ADMM drastically reduced model complexity in terms of the degrees of freedom in trained models while maintaining the performance. There may exist theoretically cleverer approaches to feature grouping, but the performance of DC-ADMM is close to the upper bound. We believe our method, DC-ADMM, to be very useful for actual use.

## References

- Galen Andrew and Jianfeng Gao. 2007. Scalable Training of L1-regularized Log-linear Models. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 33–40. Omnipress.
- Amir Beck and Marc Teboulle. 2009. A Fast Iterative Shrinkage-thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.
- Howard D. Bondell and Brian J. Reich. 2008. Simultaneous Regression Shrinkage, Variable Selection and Clustering of Predictors with OSCAR. *Biometrics*, 64(1):115.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Foundations and Trends in Machine Learning.
- Xavier Carreras. 2007. Experiments with a Higher-Order Projective Dependency Parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 957–961.
- 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.
- John Duchi and Yoram Singer. 2009. Efficient Online and Batch Learning Using Forward Backward Splitting. *Journal of Machine Learning Research*, 10:2899–2934.
- Kevin Duh, Jun Suzuki, and Masaaki Nagata. 2011. Distributed Learning-to-Rank on Streaming Data using Alternating Direction Method of Multipliers. In *NIPS’11 Big Learning Workshop*.
- Hugh Everett. 1963. Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11(3):399–417.
- Daniel Gabay and Bertrand Mercier. 1976. A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation. *Computers and Mathematics with Applications*, 2(1):17–40.
- Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. 2007. A comparative study of parameter estimation methods for statistical natural language processing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 824–831, Prague, Czech Republic, June. Association for Computational Linguistics.
- Rebecka Jörnsten and Bin Yu. 2003. Simultaneous Gene Clustering and Subset Selection for Sample Classification Via MDL. *Bioinformatics*, 19(9):1100–1109.
- Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning (ICML 2001)*, pages 282–289.
- Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Programming, Ser. B*, 45(3):503–528.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online Large-margin Training of Dependency Parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98.
- Xiaotong Shen and Hsin-Cheng Huang. 2010. Grouping Pursuit Through a Regularization Solution Surface. *Journal of the American Statistical Association*, 105(490):727–739.
- Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. Sparsity and Smoothness via the Fused Lasso. *Journal of the Royal Statistical Society Series B*, pages 91–108.
- Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003*, pages 142–147.
- Yoshimasa Tsuruoka, Jun’ichi Tsujii, and Sophia Ananiadou. 2009. Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485.
- Lin Xiao. 2010. Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. *Journal of Machine Learning Research*, 11:2543–2596.
- Leon Wenliang Zhong and James T. Kwok. 2011. Efficient Sparse Modeling with Automatic Feature Grouping. In *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.

# Exploiting Topic based Twitter Sentiment for Stock Prediction

Jianfeng Si\* Arjun Mukherjee† Bing Liu† Qing Li\* Huayi Li† Xiaotie Deng‡

\*Department of Computer Science, City University of Hong Kong, Hong Kong, China

{thankjeff@gmail.com, qing.li@cityu.edu.hk}

†Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA

{arjun4787@gmail.com, liub@cs.uic.edu, lhymvp@gmail.com}

‡AIMS Lab, Department of Computer Science, Shanghai Jiaotong University, Shanghai, China

‡deng-xt@cs.sjtu.edu.cn

## Abstract

This paper proposes a technique to leverage topic based sentiments from Twitter to help predict the stock market. We first utilize a continuous Dirichlet Process Mixture model to learn the daily topic set. Then, for each topic we derive its sentiment according to its opinion words distribution to build a sentiment time series. We then regress the stock index and the Twitter sentiment time series to predict the market. Experiments on real-life S&P100 Index show that our approach is effective and performs better than existing state-of-the-art non-topic based methods.

## 1 Introduction

Social media websites such as Twitter, Facebook, etc., have become ubiquitous platforms for social networking and content sharing. Every day, they generate a huge number of messages, which give researchers an unprecedented opportunity to utilize the messages and the public opinions contained in them for a wide range of applications (Liu, 2012). In this paper, we use them for the application of stock index time series analysis.

Here are some example tweets upon querying the keyword “\$aapl” (which is the stock symbol for Apple Inc.) in Twitter:

1. “Shanghai Oriental Morning Post confirming w Sources that **\$AAPL** TV will debut in May, Prices range from \$1600-\$3200, but \$32,000 for a 50"wow.”
2. “**\$AAPL** permanently lost its bid for a ban on U.S. sales of the Samsung Galaxy Nexus <http://dthin.gs/XqcY74>.”
3. “**\$AAPL** is loosing customers. everybody is buying android phones! **\$GOOG**.”

\* The work was done when the first author was visiting University of Illinois at Chicago.

As shown, the retrieved tweets may talk about Apple’s products, Apple’s competition relationship with other companies, etc. These messages are often related to people’s sentiments about Apple Inc., which can affect or reflect its stock trading since positive sentiments can impact sales and financial gains. Naturally, this hints that topic based sentiment is a useful factor to consider for stock prediction as they reflect people’s sentiment on different topics in a certain time frame.

This paper focuses on daily one-day-ahead prediction of stock index based on the temporal characteristics of topics in Twitter in the recent past. Specifically, we propose a non-parametric topic-based sentiment time series approach to analyzing the streaming Twitter data. The key motivation here is that Twitter’s streaming messages reflect fresh sentiments of people which are likely to be correlated with stocks in a short time frame. We also analyze the effect of training window size which best fits the temporal dynamics of stocks. Here window size refers to the number of days of tweets used in model building.

Our final prediction model is built using vector autoregression (VAR). To our knowledge, this is the first attempt to use non-parametric continuous topic based Twitter sentiments for stock prediction in an autoregressive framework.

## 2 Related Work

### 2.1 Market Prediction and Social Media

Stock market prediction has attracted a great deal of attention in the past. Some recent researches suggest that news and social media such as blogs, micro-blogs, etc., can be analyzed to extract public sentiments to help predict the market (Lavrenko et al., 2000; Schumaker and Chen, 2009). Bollen et al. (2011) used tweet based public mood to predict the movement of Dow Jones



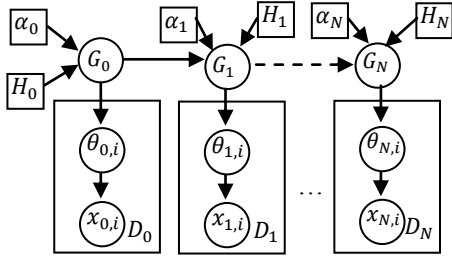


Figure 1: Continuous DPM.

Industrial Average index. Ruiz et al. (2012) studied the relationship between Twitter activities and stock market under a graph based view. Feldman et al. (2011) introduced a hybrid approach for stock sentiment analysis based on companies’ news articles.

## 2.2 Aspect and Sentiment Models

Topic modeling as a task of corpus exploration has attracted significant attention in recent years. One of the basic and most widely used models is Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA can learn a predefined number of topics and has been widely applied in its extended forms in sentiment analysis and many other tasks (Mei et al., 2007; Branavan et al., 2008; Lin and He, 2009; Zhao et al., 2010; Wang et al., 2010; Brody and Elhadad, 2010; Jo and Oh, 2011; Moghaddam and Ester, 2011; Sauper et al., 2011; Mukherjee and Liu, 2012; He et al., 2012).

The Dirichlet Processes Mixture (DPM) model is a non-parametric extension of LDA (Teh et al., 2006), which can estimate the number of topics inherent in the data itself. In this work, we employ topic based sentiment analysis using DPM on Twitter posts (or tweets). First, we employ a DPM to estimate the number of topics in the streaming snapshot of tweets in each day.

Next, we build a sentiment time series based on the estimated topics of daily tweets. Lastly, we regress the stock index and the sentiment time series in an autoregressive framework.

## 3 Model

We now present our stock prediction framework.

### 3.1 Continuous DPM Model

Comparing to edited articles, it is much harder to preset the number of topics to best fit continuous streaming Twitter data due to the large topic diversity in tweets. Thus, we resort to a non-parametric approach: the Dirichlet Process Mixture (DPM) model, and let the model estimate the number of topics inherent in the data itself.

Mixture model is widely used in clustering and

can be formalized as follows:

$$x_i \sim \sum_{k=1}^K \pi_k p(x_i | z_i = k) \quad (1)$$

where  $x_i$  is a data point,  $z_i$  is its cluster label,  $K$  is the number of topics,  $p(x_i | z_i = k)$  is the statistical (topic) models:  $\{\Phi_k\}_{k=1}^K$  and  $\pi_k$  is the component weight satisfying  $\pi_k \geq 0$  and  $\sum_k \pi_k = 1$ .

In our setting of DPM, the number of mixture components (topics)  $K$  is unfixed *a priori* but estimated from tweets in each day. DPM is defined as in (Neal, 2010):

$$\begin{aligned} x_i | \theta_i &\sim \text{Mult}(\theta_i) \\ \theta_i | G &\sim G \\ G &\sim \text{DP}(H, \alpha) \end{aligned} \quad (2)$$

where  $\theta_i$  is the parameter of the model that  $x_i$  belongs to, and  $G$  is defined as a Dirichlet Process with the base measure  $H$  and the concentration parameter  $\alpha$  (Neal, 2010).

We note that neighboring days may share the same or closely related topics because some topics may last for a long period of time covering multiple days, while other topics may just last for a short period of time. Given a set of time-stamped tweets, the overall generative process should be dynamic as the topics evolve over time. There are several ways to model this dynamic nature (Sun et al., 2010; Kim and Oh, 2011; Chua and Asur, 2012; Blei and Lafferty, 2006; Wang et al., 2008). In this paper, we follow the approach of Sun et al. (2010) due to its generality and extensibility.

Figure 1 shows the graphical model of our continuous version of DPM (which we call cDPM). As shown, the tweets set is divided into daily based collections:  $\{D_0, D_1, \dots, D_N\}$ .  $\{x_{t,i}\}_{i=1}^{|D_t|}$  are the observed tweets and  $\{\theta_{t,i}\}_{i=1}^{|D_t|}$  are the model parameters (latent topics) that generate these tweets. For each subset of tweets,  $D_t$  (tweets of day  $t$ ), we build a DPM on it. For the first day ( $t = 0$ ), the model functions the same as a standard DPM, i.e., all the topics use the same base measure,  $H_0 \sim \text{Dir}(\beta)$ . However, for later days ( $t > 0$ ), besides the base measure,  $H_t \sim \text{Dir}(\beta)$ , we make use of topics learned from previous days as priors. This ensures smooth topic chains or links (details in §3.2). For efficiency, we only consider topics of one previous day as priors.

We use collapsed Gibbs sampling (Bishop, 2006) for model inference. Hyper-parameters are set to:  $\alpha_0 = \alpha_1 = \dots = \alpha = 1$ ;  $\beta = 0.5$  as in (Sun et al., 2010; Teh et al., 2006) which have been shown to work well. Because a tweet has at most 140 characters, we assume that each tweet contains only one topic. Hence, we only need to

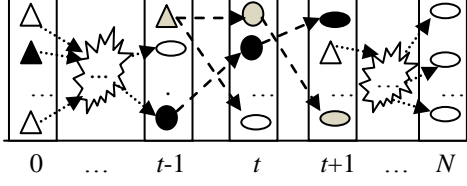


Figure 2: Linking the continuous topics via neighboring priors.

sample the topic assignment  $z_i$  for each tweet  $x_i$ .

According to different situations with respect to a topic's prior, for each tweet  $x_i$  in  $D_t$ , the conditional distribution for  $z_i$  given all other tweets' topic assignments, denoted by  $z_{-i}$ , can be summarized as follows:

1.  $k^*$  is a new topic: Its candidate priors contain the symmetric base prior  $Dir(\beta)$  and topics  $\{\phi_{t-1,k}\}_{k=1}^{K_{t-1}}$  learned from  $D_{t-1}$  if  $t > 0$ .

- If  $k^*$  takes a symmetric base prior:

$$p(z_i = k^* | z_{-i}, x_i, H) \sim \frac{\alpha}{n-1+\alpha} \frac{\Gamma(\beta|V)}{\Gamma(\beta|V+n_i)} \frac{\prod_{v=1}^{|V|} \Gamma(\beta+n_{i,v})}{\prod_{v=1}^{|V|} \Gamma(\beta)} \quad (3)$$

where the first part denotes the prior probability according to the Dirichlet Process and the second part is the data likelihood (this interpretation can similarly be applied to the following three equations).

- If  $k^*$  takes one topic  $k$  from  $\{\phi_{t-1,k}\}_{k=1}^{K_{t-1}}$  as its prior:

$$p(z_i = k^* | z_{-i}, x_i, H) \sim \frac{\alpha \pi_{t-1,k}}{n-1+\alpha} \frac{\Gamma(\beta|V)}{\Gamma(\beta|V+n_i)} \frac{\prod_{v=1}^{|V|} \Gamma(|V|\phi_{t-1,k}(v)+n_{i,v})}{\prod_{v=1}^{|V|} \Gamma(|V|\beta\phi_{t-1,k}(v))} \quad (4)$$

2.  $k$  is an existing topic: We already know its prior.

- If  $k$  takes a symmetric base prior:

$$p(z_i = k | z_{-i}, x_i, H) \sim \frac{n_k^{-i}}{n-1+\alpha} \frac{\Gamma(\beta|V+n_{k,(.)}^{-i})}{\Gamma(\beta|V+n_i+n_{k,(.)}^{-i})} \frac{\prod_{v=1}^{|V|} \Gamma(\beta+n_{i,v}+n_{k,v}^{-i})}{\prod_{v=1}^{|V|} \Gamma(\beta+n_{k,v}^{-i})} \quad (5)$$

- If  $k$  takes topic  $\phi_{t-1,k}$  as its prior:

$$p(z_i = k | z_{-i}, x_i, H) \sim \frac{n_k^{-i}}{n-1+\alpha} \frac{\Gamma(\beta|V+n_{k,(.)}^{-i})}{\Gamma(\beta|V+n_i+n_{k,(.)}^{-i})} \frac{\prod_{v=1}^{|V|} \Gamma(\beta|V\phi_{t-1,k}(v)+n_{i,v}+n_{k,v}^{-i})}{\prod_{v=1}^{|V|} \Gamma(\beta|V\phi_{t-1,k}(v)+n_{k,v}^{-i})} \quad (6)$$

Notations in the above equations are listed as follows:

- $K_{t-1}$  is the number of topics learned in day  $t-1$ .
- $|V|$  is the vocabulary size.
- $n_i$  is the document length of  $x_i$ .
- $n_{i,v}$  is the term frequency of word  $v$  in  $x_i$ .
- $\phi_{t-1,k}(v)$  is the probability of word  $v$  in previous day's topic  $k$ .
- $n_k^{-i}$  is the number of tweets assigned to topic  $k$

excluding the current one  $x_i$ .

- $n_{k,v}^{-i}$  is the term frequency of word  $v$  in topic  $k$ , with statistic from  $x_i$  excluded. While  $n_{k,(.)}^{-i}$  denotes the marginalized sum of all words in topic  $k$  with statistic from  $x_i$  excluded.

Similarly, the posteriors on  $\{\phi_{t,k}(v)\}$  (topic word distributions) are given according to their prior situations as follows:

- If topic  $k$  takes the base prior:

$$\phi_{t,k}(v) = (\beta + n_{k,v}) / (\beta|V| + n_{k,(.)}) \quad (7)$$

where  $n_{k,v}$  is the frequency of word  $v$  in topic  $k$  and  $n_{k,(.)}$  is the marginalized sum over all words.

- otherwise, it is defined recursively as:

$$\phi_{t,k}(v) = (\beta|V|\phi_{t-1,k}(v) + n_{k,v}) / (\beta|V| + n_{k,(.)}) \quad (8)$$

where  $\phi_{t-1,k}$  serves as the topic prior for  $\phi_{t,k}$ .

Finally, for each day we estimate the topic weights,  $\pi_k$  as follows:

$$\pi_k = n_k / \sum_{k'} n_{k'} \quad (9)$$

where  $n_k$  is the number of tweets in topic  $k$ .

### 3.2 Topic-based Sentiment Time Series

Based on an opinion lexicon  $O$  (a list of positive and negative opinion words, e.g., *good* and *bad*), each opinion word,  $o \in O$  is assigned with a polarity label  $l(o)$  as “+1” if it is positive and “-1” if negative. We split each tweet's text into opinion part and non-opinion part. Only non-opinion words in tweets are used for Gibbs sampling.

Based on DPM, we learn a set of topics from the non-opinion words space  $V$ . The corresponding tweets' opinion words share the same topic assignments as its tweet. Then, we compute the posterior on opinion word probability,  $\phi_{t,k}^i(o)$  for topic  $k$  analogously to equations (7) and (8). Finally, we define the topic based sentiment score  $S(t, k)$  of topic  $k$  in day  $t$  as a weighted linear combination of the opinion polarity labels:

$$S(t, k) = \sum_{o=1}^{|O|} \phi_{t,k}^i(o) l(o); S(t, k) \in [-1, 1] \quad (10)$$

According to the generative process of cDPM, topics between neighboring days are linked if a topic  $k$  takes another topic as its prior. We regard this as evolution of topic  $k$ . Although there may be slight semantic variation, the assumption is reasonable. Then, the sentiment scores for each topic series form the sentiment time series  $\{\dots, S(t-1, k), S(t, k), S(t+1, k), \dots\}$ .

Figure 2 demonstrates the linking process where a triangle denotes a new topic (with base symmetric prior), a circle denotes a middle topic (taking a topic from the previous day as its prior,

while also supplying prior for the next day) and an ellipse denotes an end topic (no further topics use it as a prior). In this example, two continuous topic chains or links (via linked priors) exist for the time interval  $[t-1, t+1]$ : one in light grey color, and the other in black. As shown, there may be more than one topic chain/link (5-20 in our experiments) for a certain time interval<sup>1</sup>. Thus, we sort multiple sentiment series according to their accumulative weights of topics over each link:  $\sum_{t=t_1}^{t_2} \pi_{t,k}$ . In our experiments, we try the top five series and use the one that gives the best result, which is mostly the first (top ranked) series with a few exceptions of the second series. The topics mostly focus on hot keywords like: *news*, *stocknews*, *earning*, *report*, which stimulate active discussions on the social media platform.

### 3.3 Time Series Analysis with VAR

For model building, we use vector autoregression (VAR). The first order (time steps of historical information to use:  $lag = 1$ ) VAR model for two time series  $\{x_t\}$  and  $\{y_t\}$  is given by:

$$\begin{aligned} x_t &= \vartheta_{11}x_{t-1} + \vartheta_{12}y_{t-1} + \varepsilon_{x,t} \\ y_t &= \vartheta_{21}x_{t-1} + \vartheta_{22}y_{t-1} + \varepsilon_{y,t} \end{aligned} \quad (11)$$

where  $\{\varepsilon\}$  are the white noises and  $\{\vartheta\}$  are model parameters. We use the “dse” library<sup>2</sup> in the *R* language to fit our VAR model based on least square regression.

Instead of training in one period and predicting over another disjointed period, we use a moving training and prediction process under sliding windows<sup>3</sup> (i.e., train in  $[t, t+w]$  and predict index on  $t+w+1$ ) with two main considerations:

- Due to the dynamic and random nature of both the stock market and public sentiments, we are more interested in their short term relationship.
- Based on the sliding windows, we have more training and testing points.

Figure 3 details the algorithm for stock index prediction. The accuracy is computed based on the index up and down dynamics, the function  $Match(y^*, y)$  returns *True* only if  $y^*$  (our prediction) and  $y$  (actual value) share the same index up or down direction.

<sup>1</sup> The actual topic priors for topic links are governed by the four cases of the Gibbs Sampler.

<sup>2</sup> <http://cran.r-project.org/web/packages/dse>

<sup>3</sup> This is similar to the autoregressive moving average (ARMA) models.

#### Parameter:

$w$ : training window size;  $lag$ : the order of VAR;

**Input:**  $t$ : the date of time series;  $\{x_t\}$ : sentiment time series;  $\{y_t\}$ : index time series;

**Output:** prediction accuracy.

1. for  $t = 0, 1, 2, \dots, N-w-1$
2. {
3.  $Model_t = VAR(x[t, t+w], y[t, t+w], lag);$
4.  $y_{t+w+1}^* = Model_t.Predict(x[t+w+1-lag, t+w], y[t+w+1-lag, t+w]);$
5. if ( $Match(y_{t+w+1}^*, y_{t+w+1})$ )  
 $rightNum++;$
6. }
7.  $Accuracy = rightNum / (N-w);$
8. Return *Accuracy*;

Figure 3: Prediction algorithm and accuracy

## 4 Dataset

We collected the tweets via Twitter’s REST API for streaming data, using symbols of the Standard & Poor’s 100 stocks (S&P100) as keywords. In this study, we focus only on predicting the S&P100 index. The time period of our dataset is between Nov. 2, 2012 and Feb. 7, 2013, which gave us 624782 tweets. We obtained the S&P100 index’s daily close values from Yahoo Finance.

## 5 Experiment

### 5.1 Selecting a Sentiment Metric

Bollen et al. (2011) used the mood dimension, *Calm* together with the index value itself to predict the Dow Jones Industrial Average. However, their *Calm* lexicon is not publicly available. We thus are unable to perform a direct comparison with their system. We identified and labeled a *Calm* lexicon (words like “*anxious*”, “*shocked*”, “*settled*” and “*dormant*”) using the opinion lexicon<sup>4</sup> of Hu and Liu (2004) and computed the sentiment score using the method of Bollen et al. (2011) (sentiment ratio). Our pilot experiments showed that using the full opinion lexicon of Hu and Liu (2004) actually performs consistently better than the *Calm* lexicon. Hence, we use the entire opinion lexicon in Hu and Liu (2004).

### 5.2 S&P100INDEX Movement Prediction

We evaluate the performance of our method by comparing with two baselines. The first (*Index*) uses only the index itself, which reduces the VAR model to the univariate autoregressive model (AR), resulting in only one index time series  $\{y_t\}$  in the algorithm of Figure 3.

<sup>4</sup> <http://cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>

| Lag | <i>Index</i> | <i>Raw</i> | <i>cDPM</i>         |
|-----|--------------|------------|---------------------|
| 1   | 0.48(0.54)   | 0.57(0.59) | 0.60(0.64)          |
| 2   | 0.58(0.65)   | 0.53(0.62) | 0.60(0.63)          |
| 3   | 0.52(0.56)   | 0.53(0.60) | 0.61( <b>0.68</b> ) |

Table 1: Average (best) accuracies over all training window sizes and different lags 1, 2, 3.

| Lag | <i>Raw</i> vs. <i>Index</i> | <i>cDPM</i> vs. <i>Index</i> | <i>cDPM</i> vs. <i>Raw</i> |
|-----|-----------------------------|------------------------------|----------------------------|
| 1   | 18.8%                       | 25.0%                        | 5.3%                       |
| 2   | -8.6%                       | 3.4%                         | 13.2%                      |
| 3   | 1.9%                        | 17.3%                        | 15.1%                      |

Table 2: Pairwise improvements among *Index*, *Raw* and *cDPM* averaged over all training window sizes.

When considering Twitter sentiments, existing works (Bollen et al., 2011, Ruiz et al., 2012) simply compute the sentiment score as ratio of pos/neg opinion words per day. This generates a lexicon-based sentiment time series, which is then combined with the index value series to give us the second baseline *Raw*.

In summary, *Index* uses index only with the AR model while *Raw* uses index and opinion lexicon based time series. Our *cDPM* uses index and the proposed topic based sentiment time series. Both *Raw* and *cDPM* employ the two dimensional VAR model. We experiment with different lag settings from 1-3 days.

We also experiment with different training window sizes, ranging from 15 - 30 days, and compute the prediction accuracy for each window size. Table 1 shows the respective average and best accuracies over all window sizes for each lag and Table 2 summarizes the pairwise performance improvements of averaged scores over all training window sizes. Figure 4 show the detailed accuracy comparison for lag 1 and lag 3.

From Table 1, 2, and Figure 4, we note:

- i. Topic-based public sentiments from tweets can improve stock prediction over simple sentiment ratio which may suffer from backchannel noise and lack of focus on prevailing topics. For example, on lag 2, *Raw* performs worse by 8.6% than *Index* itself.
- ii. *cDPM* outperforms all others in terms of both the best accuracy (*lag* 3) and the average accuracies for different window sizes. The maximum average improvement reaches 25.0% compared to *Index* at lag 1 and 15.1% compared to *Raw* at lag 3. This is due to the fact that *cDPM* learns the topic based sentiments instead of just using the opinion words' ratio like *Raw*, and in a short time period, some topics are more correlated with the stock mar-

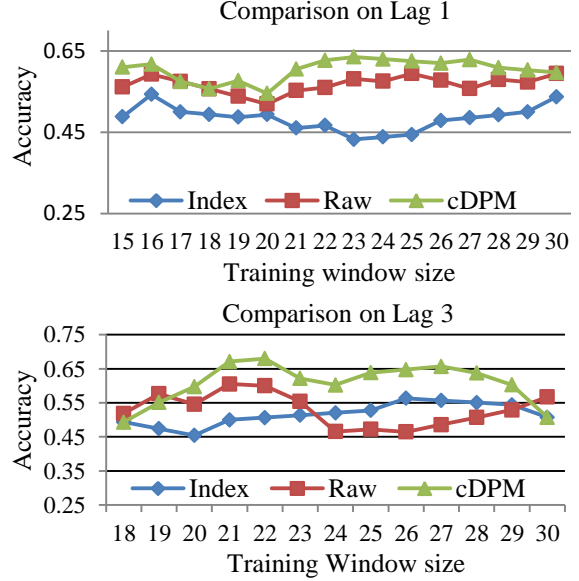


Figure 4: Comparison of prediction accuracy of up/down stock index on S&P 100 index for different training window sizes.

ket than others. Our proposed sentiment time series using *cDPM* can capture this phenomenon and also help reduce backchannel noise of raw sentiments.

- iii. On average, *cDPM* gets the best performance for training window sizes within [21, 22], and the best prediction accuracy is 68.0% on window size 22 at lag 3.

## 6 Conclusions

Predicting the stock market is an important but difficult problem. This paper showed that Twitter's topic based sentiment can improve the prediction accuracy beyond existing non-topic based approaches. Specifically, a non-parametric topic-based sentiment time series approach was proposed for the Twitter stream. For prediction, vector autoregression was used to regress S&P100 index with the learned sentiment time series. Besides the short term dynamics based prediction, we believe that the proposed method can be extended for long range dependency analysis of Twitter sentiments and stocks, which can render deep insights into the complex phenomenon of stock market. This will be part of our future work.

## Acknowledgments

This work was supported in part by a grant from the National Science Foundation (NSF) under grant no. IIS-1111092 and a strategic research grant from City University of Hong Kong (project number: 7002770).

## References

- Bishop, C. M. 2006. Pattern Recognition and Machine Learning. Springer.
- Blei, D., Ng, A. and Jordan, M. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.
- Blei, D. and Lafferty, J. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning (ICML-2006)*.
- Bollen, J., Mao, H. N., and Zeng, X. J. 2011. Twitter mood predicts the stock market. *Journal of Computer Science* 2(1):1-8.
- Branavan, S., Chen, H., Eisenstein J. and Barzilay, R. 2008. Learning document-level semantic properties from free-text annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-2008)*.
- Brody, S. and Elhadad, S. 2010. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the ACL (NAACL-2010)*.
- Chua, F. C. T. and Asur, S. 2012. Automatic Summarization of Events from Social Media, Technical Report, HP Labs.
- Feldman, R., Benjamin, R., Roy, B. H. and Moshe, F. 2011. The Stock Sonar - Sentiment analysis of stocks based on a hybrid approach. In *Proceedings of 23rd IAAI Conference on Artificial Intelligence (IAAI-2011)*.
- He, Y., Lin, C., Gao, W., and Wong, K. F. 2012. Tracking sentiment and topic dynamics from social media. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM-2012)*.
- Hu, M. and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*.
- Jo, Y. and Oh, A. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of ACM Conference in Web Search and Data Mining (WSDM-2011)*.
- Kim, D. and Oh, A. 2011. Topic chains for understanding a news corpus. *CICLING* (2): 163-176.
- Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D. and Allan, J. 2000. Mining of concurrent text and time series. In *Proceedings of the 6th KDD Workshop on Text Mining*, 37–44.
- Lin, C. and He, Y. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2009)*.
- Liu, B. 2012. Sentiment analysis and opinion mining. Morgan & Claypool Publishers.
- Mei, Q., Ling, X., Wondra, M., Su, H. and Zhai, C. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of International Conference on World Wide Web (WWW-2007)*.
- Moghaddam, S. and Ester, M. 2011. ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the Annual ACM SIGIR International conference on Research and Development in Information Retrieval (SIGIR-2011)*.
- Mukherjee A. and Liu, B. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-2012)*.
- Neal, R.M. 2000. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249-265.
- Ruiz, E. J., Hristidis, V., Castillo, C., Gionis, A. and Jaimes, A. 2012. Correlating financial time series with micro-blogging activity. In *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM-2012)*, 513-522.
- Sauper, C., Haghighi, A. and Barzilay, R. 2011. Content models with attitude. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Schumaker, R. P. and Chen, H. 2009. Textual analysis of stock market prediction using breaking financial news. *ACM Transactions on Information Systems* 27(February (2)):1–19.
- Sun, Y. Z., Tang, J. Han, J., Gupta M. and Zhao, B. 2010. Community Evolution Detection in Dynamic Heterogeneous Information Networks. In *Proceedings of KDD Workshop on Mining and Learning with Graphs (MLG'2010)*, Washington, D.C.
- Teh, Y., Jordan M., Beal, M. and Blei, D. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101[476]:1566-1581.
- Wang, C. Blei, D. and Heckerman, D. 2008. Continuous Time Dynamic Topic Models. *Uncertainty in Artificial Intelligence (UAI 2008)*, 579-586
- Wang, H., Lu, Y. and Zhai, C. 2010. Latent aspect rating analysis on review text data: a rating regression approach. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2010)*.
- Zhao, W. Jiang, J. Yan, Y. and Li, X. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2010)*.

# Learning Entity Representation for Entity Disambiguation

Zhengyan He<sup>†</sup> Shujie Liu<sup>‡</sup> Mu Li<sup>‡</sup> Ming Zhou<sup>‡</sup> Longkai Zhang<sup>†</sup> Houfeng Wang<sup>†\*</sup>

<sup>†</sup> Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China

<sup>‡</sup> Microsoft Research Asia

hezhengyan.hit@gmail.com {shujliu,muli,mingzhou}@microsoft.com  
zhlongk@qq.com wanghf@pku.edu.cn

## Abstract

We propose a novel entity disambiguation model, based on Deep Neural Network (DNN). Instead of utilizing simple similarity measures and their disjoint combinations, our method directly optimizes document and entity representations for a given similarity measure. Stacked Denoising Auto-encoders are first employed to learn an initial document representation in an unsupervised pre-training stage. A supervised fine-tuning stage follows to optimize the representation towards the similarity measure. Experiment results show that our method achieves state-of-the-art performance on two public datasets without any manually designed features, even beating complex collective approaches.

## 1 Introduction

Entity linking or disambiguation has recently received much attention in natural language processing community (Bunescu and Pasca, 2006; Han et al., 2011; Kataria et al., 2011; Sen, 2012). It is an essential first step for succeeding sub-tasks in knowledge base construction (Ji and Grishman, 2011) like populating attribute to entities. Given a sentence with four mentions, “The [[Python]] of [[Delphi]] was a creature with the body of a snake. This creature dwelled on [[Mount Parnassus]], in central [[Greece]].” How can we determine that Python is an earth-dragon in Greece mythology and not the popular programming language, Delphi is not the auto parts supplier, and Mount Parnassus is in Greece, not in Colorado?

A most straightforward method is to compare the context of the mention and the definition of candidate entities. Previous work has explored many ways of measuring the relatedness of context

$d$  and entity  $e$ , such as dot product, cosine similarity, Kullback-Leibler divergence, Jaccard distance, or more complicated ones (Zheng et al., 2010; Kulkarni et al., 2009; Hoffart et al., 2011; Bunescu and Pasca, 2006; Cucerzan, 2007; Zhang et al., 2011). However, these measures are often duplicate or over-specified, because they are disjointly combined and their atomic nature determines that they have no internal structure.

Another line of work focuses on collective disambiguation (Kulkarni et al., 2009; Han et al., 2011; Ratnov et al., 2011; Hoffart et al., 2011). Ambiguous mentions within the same context are resolved simultaneously based on the coherence among decisions. Collective approaches often undergo a non-trivial decision process. In fact, (Ratnov et al., 2011) show that even though global approaches can be improved, local methods based on only similarity  $sim(d, e)$  of context  $d$  and entity  $e$  are hard to beat. This somehow reveals the importance of a good modeling of  $sim(d, e)$ .

Rather than learning context entity association at word level, topic model based approaches (Kataria et al., 2011; Sen, 2012) can learn it in the semantic space. However, the one-topic-per-entity assumption makes it impossible to scale to large knowledge base, as every entity has a separate word distribution  $P(w|e)$ ; besides, the training objective does not directly correspond with disambiguation performances.

To overcome disadvantages of previous approaches, we propose a novel method to learn context entity association enriched with deep architecture. Deep neural networks (Hinton et al., 2006; Bengio et al., 2007) are built in a hierarchical manner, and allow us to compare context and entity at some higher level abstraction; while at lower levels, general concepts are shared across entities, resulting in compact models. Moreover, to make our model highly correlated with disambiguation performance, our method directly optimizes doc-

\*Corresponding author

ument and entity representations for a fixed similarity measure. In fact, the underlying representations for computing similarity measure add internal structure to the given similarity measure. Features are learned leveraging large scale annotation of Wikipedia, without any manual design efforts. Furthermore, the learned model is compact compared with topic model based approaches, and can be trained discriminatively without relying on expensive sampling strategy. Despite its simplicity, it beats all complex collective approaches in our experiments. The learned similarity measure can be readily incorporated into any existing collective approaches, which further boosts performance.

## 2 Learning Representation for Contextual Document

Given a mention string  $m$  with its context document  $d$ , a list of candidate entities  $C(m)$  are generated for  $m$ , for each candidate entity  $e_i \in C(m)$ , we compute a ranking score  $sim(d_m, e_i)$  indicating how likely  $m$  refers to  $e_i$ . The linking result is  $e = \arg \max_{e_i} sim(d_m, e_i)$ .

Our algorithm consists of two stages. In the pre-training stage, Stacked Denoising Auto-encoders are built in an unsupervised layer-wise fashion to discover general concepts encoding  $d$  and  $e$ . In the supervised fine-tuning stage, the entire network weights are fine-tuned to optimize the similarity score  $sim(d, e)$ .

### 2.1 Greedy Layer-wise Pre-training

Stacked Auto-encoders (Bengio et al., 2007) is one of the building blocks of deep learning. Assume the input is a vector  $x$ , an auto-encoder consists of an encoding process  $h(x)$  and a decoding process  $g(h(x))$ . The goal is to minimize the reconstruction error  $\mathcal{L}(x, g(h(x)))$ , thus retaining maximum information. By repeatedly stacking new auto-encoder on top of previously learned  $h(x)$ , stacked auto-encoders are obtained. This way we learn multiple levels of representation of input  $x$ .

One problem of auto-encoder is that it treats all words equally, no matter it is a function word or a content word. Denoising Auto-encoder (DA) (Vincent et al., 2008) seeks to reconstruct  $x$  given a random corruption  $\tilde{x}$  of  $x$ . DA can capture global structure while ignoring noise as the author shows in image processing. In our case, we input each document as a binary bag-of-words vector (Fig.

1). DA will capture general concepts and ignore noise like function words. By applying masking noise (randomly mask 1 with 0), the model also exhibits a fill-in-the-blank property (Vincent et al., 2010): the missing components must be recovered from partial input. Take “greece” for example, the model must learn to predict it with “python” “mount”, through some hidden unit. The hidden unit may somehow express the concept of Greece mythology.

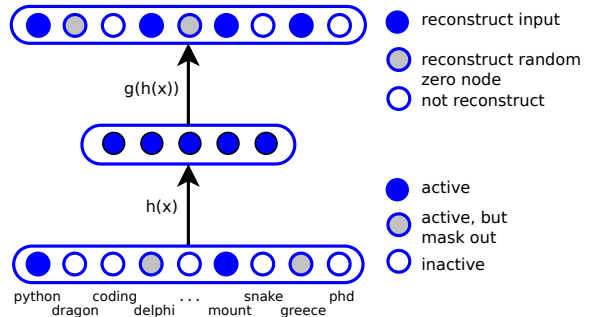


Figure 1: DA and reconstruction sampling.

In order to distinguish between a large number of entities, the vocabulary size must be large enough. This adds considerable computational overhead because the reconstruction process involves expensive dense matrix multiplication. Reconstruction sampling keeps the sparse property of matrix multiplication by reconstructing a small subset of original input, with no loss of quality of the learned representation (Dauphin et al., 2011).

### 2.2 Supervised Fine-tuning

This stage we optimize the learned representation (“hidden layer n” in Fig. 2) towards the ranking score  $sim(d, e)$ , with large scale Wikipedia annotation as supervision. We collect hyperlinks in Wikipedia as our training set  $\{(d_i, e_i, m_i)\}$ , where  $m_i$  is the mention string for candidate generation. The network weights below “hidden layer n” are initialized with the pre-training stage.

Next, we stack another layer on top of the learned representation. The whole network is tuned by the final supervised objective. The reason to stack another layer on top of the learned representation, is to capture problem specific structures. Denote the encoding of  $d$  and  $e$  as  $\hat{d}$  and  $\hat{e}$  respectively, after stacking the problem-specific layer, the representation for  $d$  is given as  $f(d) = \text{sigmoid}(W \times \hat{d} + b)$ , where  $W$  and  $b$  are weight and bias term respectively.  $f(e)$  follows the same

encoding process.

The similarity score of  $(d, e)$  pair is defined as the dot product of  $f(d)$  and  $f(e)$  (Fig. 2):

$$\text{sim}(d, e) = \text{Dot}(f(d), f(e)) \quad (1)$$

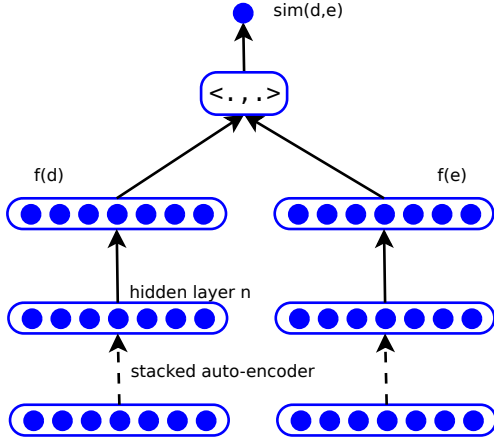


Figure 2: Network structure of fine-tuning stage.

Our goal is to rank the correct entity higher than the rest candidates relative to the context of the mention. For each training instance  $(d, e)$ , we contrast it with one of its negative candidate pair  $(d, e')$ . This gives the pairwise ranking criterion:

$$\mathcal{L}(d, e) = \max\{0, 1 - \text{sim}(d, e) + \text{sim}(d, e')\} \quad (2)$$

Alternatively, we can contrast with all its candidate pairs  $(d, e_i)$ . That is, we raise the similarity score of true pair  $\text{sim}(d, e)$  and penalize all the rest  $\text{sim}(d, e_i)$ . The loss function is defined as negative log of *softmax* function:

$$\mathcal{L}(d, e) = -\log \frac{\exp \text{sim}(d, e)}{\sum_{e_i \in C(m)} \exp \text{sim}(d, e_i)} \quad (3)$$

Finally, we seek to minimize the following training objective across all training instances:

$$\mathcal{L} = \sum_{d, e} \mathcal{L}(d, e) \quad (4)$$

The loss function is closely related to contrastive estimation (Smith and Eisner, 2005), which defines where the positive example takes probability mass from. We find that by penalizing more negative examples, convergence speed can be greatly accelerated. In our experiments, the *softmax* loss function consistently outperforms pairwise ranking loss function, which is taken as our default setting.

However, the *softmax* training criterion adds additional computational overhead when performing mini-batch Stochastic Gradient Descent (SGD). Although we can use a plain SGD (i.e. mini-batch size is 1), mini-batch SGD is faster to converge and more stable. Assume the mini-batch size is  $m$  and the number of candidates is  $n$ , a total of  $m \times n$  forward-backward passes over the network are performed to compute a similarity matrix (Fig. 3), while pairwise ranking criterion only needs  $2 \times m$ . We address this problem by grouping training pairs with same mention  $m$  into one mini-batch  $\{(d, e_i) | e_i \in C(m)\}$ . Observe that if candidate entities overlap, they share the same forward-backward path. Only  $m + n$  forward-backward passes are needed for each mini-batch now.

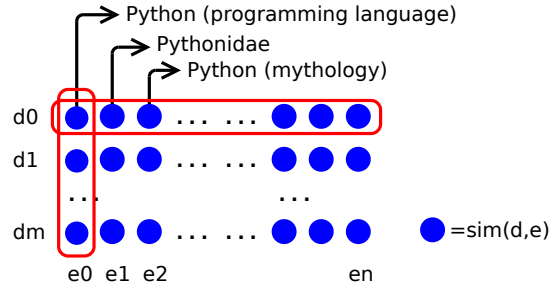


Figure 3: Sharing path within mini-batch.

The re-organization of mini-batch is similar in spirit to Backpropagation Through Structure (BTS) (Goller and Kuchler, 1996). BTS is a variant of the general backpropagation algorithm for structured neural network. In BTS, parent node is computed with its child nodes at the forward pass stage; child node receives gradient as the sum of derivatives from all its parents. Here (Fig. 2), parent node is the score node  $\text{sim}(d, e)$  and child nodes are  $f(d)$  and  $f(e)$ . In Figure 3, each row shares forward path of  $f(d)$  while each column shares forward path of  $f(e)$ . At backpropagation stage, gradient is summed over each row of score nodes for  $f(d)$  and over each column for  $f(e)$ .

Till now, our input simply consists of bag-of-words binary vector. We can incorporate any handcrafted feature  $f(d, e)$  as:

$$\text{sim}(d, e) = \text{Dot}(f(d), f(e)) + \vec{\lambda} \vec{f}(d, e) \quad (5)$$

In fact, we find that with only  $\text{Dot}(f(d), f(e))$  as ranking score, the performance is sufficiently good. So we leave this as our future work.



### 3 Experiments and Analysis

**Training settings:** In pre-training stage, input layer has 100,000 units, all hidden layers have 1,000 units with rectifier function  $\max(0, x)$ . Following (Glorot et al., 2011), for the first reconstruction layer, we use sigmoid activation function and cross-entropy error function. For higher reconstruction layers, we use *softplus* ( $\log(1 + \exp(x))$ ) as activation function and squared loss as error function. For corruption process, we use a masking noise probability in  $\{0.1, 0.4, 0.7\}$  for the first layer, a Gaussian noise with standard deviation of 0.1 for higher layers. For reconstruction sampling, we set the reconstruction rate to 0.01. In fine-tuning stage, the final layer has 200 units with sigmoid activation function. The learning rate is set to  $1e-3$ . The mini-batch size is set to 20.

We run all our experiments on a Linux machine with 72GB memory 6 core Xeon CPU. The model is implemented in Python with C extensions, numpy configured with Openblas library. Thanks to reconstruction sampling and refined mini-batch arrangement, it takes about 1 day to converge for pre-training and 3 days for fine-tuning, which is fast given our training set size.

**Datasets:** We use half of Wikipedia<sup>1</sup> plain text (~1.5M articles split into sections) for pre-training. We collect a total of 40M hyperlinks grouped by name string  $m$  for fine-tuning stage. We holdout a subset of hyperlinks for model selection, and we find that 3 layers network with a higher masking noise rate (0.7) always gives best performance.

We select TAC-KBP 2010 (Ji and Grishman, 2011) dataset for non-collective approaches, and AIDA<sup>2</sup> dataset for collective approaches. For both datasets, we evaluate the non-NIL queries. The TAC-KBP and AIDA testb dataset contains 1020 and 4485 non-NIL queries respectively.

For candidate generation, mention-to-entity dictionary is built by mining Wikipedia structures, following (Cucerzan, 2007). We keep top 30 candidates by prominence  $P(e|m)$  for speed consideration. The candidate generation recall are 94.0% and 98.5% for TAC and AIDA respectively.

**Analysis:** Table 1 shows evaluation results across several best performing systems. (Han et al., 2011) is a collective approach, using Personalized PageRank to propagate evidence between

<sup>1</sup>available at <http://dumps.wikimedia.org/enwiki/>, we use the 20110405 xml dump.

<sup>2</sup>available at <http://www.mpi-inf.mpg.de/yago-naga/aida/>

different decisions. To our surprise, our method with only local evidence even beats several complex collective methods with simple word similarity. This reveals the importance of context modeling in semantic space. Collective approaches can improve performance only when local evidence is not confident enough. When embedding our similarity measure  $\text{sim}(d, e)$  into (Han et al., 2011), we achieve the best results on AIDA.

A close error analysis shows some typical errors due to the lack of prominence feature and name matching feature. Some queries accidentally link to rare candidates and some link to entities with completely different names. We will add these features as mentioned in Eq. 5 in future. We will also add NIL-detection module, which is required by more realistic application scenarios. A first thought is to construct pseudo-NIL with Wikipedia annotations and automatically learn the threshold and feature weight as in (Bunescu and Pasca, 2006; Kulkarni et al., 2009).

| Methods                              | micro P@1 | macro P@1 |
|--------------------------------------|-----------|-----------|
| TAC 2010 eval                        |           |           |
| Lcc (2010) (top1, noweb)             | 79.22     | -         |
| Siel 2010 (top2, noweb)              | 71.57     | -         |
| our best                             | 80.97     | -         |
| AIDA dataset (collective approaches) |           |           |
| AIDA (2011)                          | 82.29     | 82.02     |
| Shirakawa et al. (2011)              | 81.40     | 83.57     |
| Kulkarni et al. (2009)               | 72.87     | 76.74     |
| wordsim (cosine)                     | 48.38     | 37.30     |
| Han (2011) +wordsim                  | 78.97     | 75.77     |
| our best (non-collective)            | 84.82     | 83.37     |
| Han (2011) + our best                | 85.62     | 83.95     |

Table 1: Evaluation on TAC and AIDA dataset.

### 4 Conclusion

We propose a deep learning approach that automatically learns context-entity similarity measure for entity disambiguation. The intermediate representations are learned leveraging large scale annotations of Wikipedia, without any manual effort of designing features. The learned representation of entity is compact and can scale to very large knowledge base. Furthermore, experiment reveals the importance of context modeling in this field. By incorporating our learned measure into collective approach, performance is further improved.

## Acknowledgments

We thank Nan Yang, Jie Liu and Fei Wang for helpful discussions. This research was partly supported by National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), National Natural Science Foundation of China (No.91024009) and Major National Social Science Fund of China(No. 12&ZD227).

## References

- Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. 2007. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153.
- R. Bunescu and M. Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16.
- S. Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 6, pages 708–716.
- Y. Dauphin, X. Glorot, and Y. Bengio. 2011. Large-scale learning of embeddings with reconstruction sampling. In *Proceedings of the Twenty-eighth International Conference on Machine Learning (ICML11)*.
- X. Glorot, A. Bordes, and Y. Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*.
- Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.
- X. Han, L. Sun, and J. Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM.
- G.E. Hinton, S. Osindero, and Y.W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.
- J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenu, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.
- Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA, June. Association for Computational Linguistics.
- S.S. Kataria, K.S. Kumar, R. Rastogi, P. Sen, and S.H. Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *Proceedings of KDD*.
- S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM.
- J. Lehmann, S. Monahan, L. Nezda, A. Jung, and Y. Shi. 2010. Lcc approaches to knowledge base population at tac 2010. In *Proc. TAC 2010 Workshop*.
- L. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.
- P. Sen. 2012. Collective context-aware topic models for entity disambiguation. In *Proceedings of the 21st international conference on World Wide Web*, pages 729–738. ACM.
- M. Shirakawa, H. Wang, Y. Song, Z. Wang, K. Nakayama, T. Hara, and S. Nishio. 2011. Entity disambiguation based on a probabilistic taxonomy. Technical report, Technical Report MSR-TR-2011-125, Microsoft Research.
- N.A. Smith and J. Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362. Association for Computational Linguistics.
- P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.
- Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408.
- W. Zhang, Y.C. Sim, J. Su, and C.L. Tan. 2011. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1909–1914. AAAI Press.
- Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491, Los Angeles, California, June. Association for Computational Linguistics.

# Natural Language Models for Predicting Programming Comments

**Dana Movshovitz-Attias**

Computer Science Department  
Carnegie Mellon University  
dma@cs.cmu.edu

**William W. Cohen**

Computer Science Department  
Carnegie Mellon University  
wcohen@cs.cmu.edu

## Abstract

Statistical language models have successfully been used to describe and analyze natural language documents. Recent work applying language models to programming languages is focused on the task of predicting code, while mainly ignoring the prediction of programmer comments. In this work, we predict comments from JAVA source files of open source projects, using topic models and n-grams, and we analyze the performance of the models given varying amounts of background data on the project being predicted. We evaluate models on their comment-completion capability in a setting similar to code-completion tools built into standard code editors, and show that using a comment completion tool can save up to 47% of the comment typing.

## 1 Introduction and Related Work

Statistical language models have traditionally been used to describe and analyze natural language documents. Recently, software engineering researchers have adopted the use of language models for modeling software code. Hindle et al. (2012) observe that, as code is created by humans it is likely to be repetitive and predictable, similar to natural language. NLP models have thus been used for a variety of software development tasks such as code token completion (Han et al., 2009; Jacob and Tairas, 2010), analysis of names in code (Lawrie et al., 2006; Binkley et al., 2011) and mining software repositories (Gabel and Su, 2008).

An important part of software programming and maintenance lies in documentation, which may come in the form of tutorials describing the code, or inline comments provided by the programmer. The documentation provides a high level description of the task performed by the code, and may

include examples of use-cases for specific code segments or identifiers such as classes, methods and variables. Well documented code is easier to read and maintain in the long-run but writing comments is a laborious task that is often overlooked or at least postponed by many programmers.

Code commenting not only provides a summarization of the conceptual idea behind the code (Sridhara et al., 2010), but can also be viewed as a form of document expansion where the comment contains significant terms relevant to the described code. Accurately predicted comment words can therefore be used for a variety of linguistic uses including improved search over code bases using natural language queries, code categorization, and locating parts of the code that are relevant to a specific topic or idea (Tseng and Juang, 2003; Wan et al., 2007; Kumar and Carterette, 2013; Shepherd et al., 2007; Rastkar et al., 2011). A related and well studied NLP task is that of predicting natural language caption and commentary for images and videos (Blei and Jordan, 2003; Feng and Lapata, 2010; Feng and Lapata, 2013; Wu and Li, 2011).

In this work, our goal is to apply statistical language models for predicting class comments. We show that n-gram models are extremely successful in this task, and can lead to a saving of up to 47% in comment typing. This is expected as n-grams have been shown as a strong model for language and speech prediction that is hard to improve upon (Rosenfeld, 2000). In some cases however, for example in a document expansion task, we wish to extract important terms relevant to the code regardless of local syntactic dependencies. We hence also evaluate the use of LDA (Blei et al., 2003) and link-LDA (Erosheva et al., 2004) topic models, which are more relevant for the term extraction scenario. We find that the topic model performance can be improved by distinguishing *code* and *text* tokens in the code.

## 2 Method

### 2.1 Models

We train  $n$ -gram models ( $n = 1, 2, 3$ ) over source code documents containing sequences of combined code and text tokens from multiple training datasets (described below). We use the Berkeley Language Model package (Pauls and Klein, 2011) with absolute discounting (Kneser-Ney smoothing; (1995)) which includes a backoff strategy to lower-order  $n$ -grams. Next, we use LDA topic models (Blei et al., 2003) trained on the same data, with 1, 5, 10 and 20 topics. The joint distribution of a topic mixture  $\theta$ , and a set of  $N$  topics  $z$ , for a single source code document with  $N$  observed word tokens,  $d = \{w_i\}_{i=1}^N$ , given the Dirichlet parameters  $\alpha$  and  $\beta$ , is therefore

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_w p(z|\theta)p(w|z, \beta) \quad (1)$$

Under the models described so far, there is no distinction between text and code tokens.

Finally, we consider documents as having a mixed membership of two entity types, *code* and *text* tokens,  $d = (\{w_i^{code}\}_{i=1}^{C_n}, \{w_i^{text}\}_{i=1}^{T_n})$ , where the *text* words are tokens from comment and string literals, and the *code* words include the programming language syntax tokens (e.g., `public`, `private`, `for`, etc') and all identifiers. In this case, we train link-LDA models (Erosheva et al., 2004) with 1, 5, 10 and 20 topics. Under the link-LDA model, the mixed-membership joint distribution of a topic mixture, words and topics is then

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \cdot \prod_{w^{text}} p(z^{text}|\theta)p(w^{text}|z^{text}, \beta) \cdot \prod_{w^{code}} p(z^{code}|\theta)p(w^{code}|z^{code}, \beta) \quad (2)$$

where  $\theta$  is the joint topic distribution,  $w$  is the set of observed document words,  $z^{text}$  is a topic associated with a text word, and  $z^{code}$  a topic associated with a code word.

The LDA and link-LDA models use Gibbs sampling (Griffiths and Steyvers, 2004) for topic inference, based on the implementation of Balasubramanian and Cohen (2011) with single or multiple entities per document, respectively.

### 2.2 Testing Methodology

Our goal is to predict the tokens of the JAVA class comment (the one preceding the class definition) in each of the test files. Each of the models described above assigns a probability to the next comment token. In the case of  $n$ -grams, the probability of a token word  $w_i$  is given by considering previous words  $p(w_i|w_{i-1}, \dots, w_0)$ . This probability is estimated given the previous  $n - 1$  tokens as  $p(w_i|w_{i-1}, \dots, w_{i-(n-1)})$ .

For the topic models, we separate the document tokens into the class definition and the comment we wish to predict. The set of tokens of the class comment  $w^c$ , are all considered as text tokens. The rest of the tokens in the document  $w^r$ , are considered to be the class definition, and they may contain both code and text tokens (from string literals and other comments in the source file). We then compute the posterior probability of document topics by solving the following inference problem conditioned on the  $w^r$  tokens

$$p(\theta, z^r|w^r, \alpha, \beta) = \frac{p(\theta, z^r, w^r|\alpha, \beta)}{p(w^r|\alpha, \beta)} \quad (3)$$

This gives us an estimate of the document distribution,  $\theta$ , with which we infer the probability of the comment tokens as

$$p(w^c|\theta, \beta) = \sum_z p(w^c|z, \beta)p(z|\theta) \quad (4)$$

Following Blei et al. (2003), for the case of a single entity LDA, the inference problem from equation (3) can be solved by considering  $p(\theta, z, w|\alpha, \beta)$ , as in equation (1), and by taking the marginal distribution of the document tokens as a continuous mixture distribution for the set  $w = w^r$ , by integrating over  $\theta$  and summing over the set of topics  $z$

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \cdot \left( \prod_w \sum_z p(z|\theta)p(w|z, \beta) \right) d\theta \quad (5)$$

For the case of link-LDA where the document is comprised of two entities, in our case *code* tokens and *text* tokens, we can consider the mixed-membership joint distribution  $\theta$ , as in equation (2), and similarly the marginal distribution  $p(w|\alpha, \beta)$  over both code and text tokens from  $w^r$ . Since comment words in  $w^c$  are all considered as text tokens they are sampled using *text* topics, namely  $z^{text}$ , in equation (4).

### 3 Experimental Settings

#### 3.1 Data and Training Methodology

We use source code from nine open source JAVA projects: Ant, Cassandra, Log4j, Maven, Minor-Third, Batik, Lucene, Xalan and Xerces. For each project, we divide the source files into a training and testing dataset. Then, for each project in turn, we consider the following three main training scenarios, leading to using three training datasets.

To emulate a scenario in which we are predicting comments in the middle of project development, we can use data (documented code) from the same project. In this case, we use the in-project training dataset (*IN*). Alternatively, if we train a comment prediction model at the beginning of the development, we need to use source files from other, possibly related projects. To analyze this scenario, for each of the projects above we train models using an out-of-project dataset (*OUT*) containing data from the other eight projects.

Typically, source code files contain a greater amount of code versus comment text. Since we are interested in predicting comments, we consider a third training data source which contains more English text as well as some code segments. We use data from the popular Q&A website StackOverflow (*SO*) where users ask and answer technical questions about software development, tools, algorithms, etc'. We downloaded a dataset of all actions performed on the site since it was launched in August 2008 until August 2012. The data includes 3,453,742 questions and 6,858,133 answers posted by 1,295,620 users. We used only posts that are tagged as JAVA related questions and answers.

All the models for each project are then tested on the testing set of that project. We report results averaged over all projects in Table 1.

Source files were tokenized using the Eclipse JDT compiler tools, separating code tokens and identifiers. Identifier names (of classes, methods and variables), were further tokenized by camel case notation (e.g., 'minMargin' was converted to 'min margin'). Non alpha-numeric tokens (e.g., dot, semicolon) were discarded from the code, as well as numeric and single character literals. Text from comments or any string literals within the code were further tokenized with the Mallet statistical natural language processing package (McCallum, 2002). Posts from SO were parsed using

the Apache Tika toolkit<sup>1</sup> and then tokenized with the Mallet package. We considered as raw code tokens anything labeled using a `<code>` markup (as indicated by the SO users who wrote the post).

#### 3.2 Evaluation

Since our models are trained using various data sources the vocabularies used by each of them are different, making the comment likelihood given by each model incomparable due to different sets of out-of-vocabulary tokens. We thus evaluate models using a character saving metric which aims at quantifying the percentage of characters that can be saved by using the model in a word-completion settings, similar to standard code completion tools built into code editors. For a comment word with  $n$  characters,  $w = w_1, \dots, w_n$ , we predict the two most likely words given each model filtered by the first  $0, \dots, n$  characters of  $w$ . Let  $k_i$  be the minimal  $k_i$  for which  $w$  is in the top two predicted word tokens where tokens are filtered by the first  $k_i$  characters. Then, the number of saved characters for  $w$  is  $n - k$ . In Table 1 we report the average percentage of saved characters per comment using each of the above models. The final results are also averaged over the nine input projects. As an example, in the predicted comment shown in Table 2, taken from the project *Minor-Third*, the token *entity* is the most likely token according to the model *SO trigram*, out of tokens starting with the prefix 'en'. The saved characters in this case are 'tity'.

### 4 Results

Table 1 displays the average percentage of characters saved per class comment using each of the models. Models trained on in-project data (*IN*) perform significantly better than those trained on another data source, regardless of the model type, with an average saving of 47.1% characters using a trigram model. This is expected, as files from the same project are likely to contain similar comments, and identifier names that appear in the comment of one class may appear in the code of another class in the same project. Clearly, in-project data should be used when available as it improves comment prediction leading to an average increase of between 6% for the worst model (26.6 for *OUT* unigram versus 33.05 for *IN*) and 14% for the best (32.96 for *OUT* trigram versus 47.1 for *IN*).

<sup>1</sup><http://tika.apache.org/>

| Model      | $n$ -gram       |                 |                 | LDA             |                 |                 |                 | Link-LDA        |                 |                 |                 |
|------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|            | 1               | 2               | 3               | 20              | 10              | 5               | 1               | 20              | 10              | 5               | 1               |
| <i>IN</i>  | 33.05<br>(3.62) | 43.27<br>(5.79) | 47.1<br>(6.87)  | 34.20<br>(3.63) | 33.93<br>(3.67) | 33.63<br>(3.67) | 33.05<br>(3.62) | 35.76<br>(3.95) | 35.81<br>(4.12) | 35.37<br>(3.98) | 34.59<br>(3.92) |
| <i>OUT</i> | 26.6<br>(3.37)  | 31.52<br>(4.17) | 32.96<br>(4.33) | 26.79<br>(3.26) | 26.8<br>(3.36)  | 26.86<br>(3.44) | 26.6<br>(3.37)  | 28.03<br>(3.60) | 28<br>(3.56)    | 28<br>(3.67)    | 27.82<br>(3.62) |
| <i>SO</i>  | 27.8<br>(3.51)  | 33.29<br>(4.40) | 34.56<br>(4.78) | 27.25<br>(3.67) | 27.22<br>(3.44) | 27.34<br>(3.55) | 27.8<br>(3.51)  | 28.08<br>(3.48) | 28.12<br>(3.58) | 27.94<br>(3.56) | 27.9<br>(3.45)  |

Table 1: Average percentage of characters saved per comment using  $n$ -gram, LDA and link-LDA models trained on three training sets: *IN*, *OUT*, and *SO*. The results are averaged over nine JAVA projects (with standard deviations in parenthesis).

| Model              | Predicted Comment                       |
|--------------------|---|
| <i>IN</i> trigram  | “Train a <u>named-entity</u> extractor” |
| <i>IN</i> link-LDA | “Train a <u>named-entity</u> extractor” |
| <i>OUT</i> trigram | “Train a <u>named-entity</u> extractor” |
| <i>SO</i> trigram  | “Train a <u>named-entity</u> extractor” |

Table 2: Sample comment from the *Minor-Third* project predicted using *IN*, *OUT* and *SO* based models. Saved characters are underlined.

Of the out-of-project data sources, models using a greater amount of text (*SO*) mostly outperformed models based on more code (*OUT*). This increase in performance, however, comes at a cost of greater run-time due to the larger word dictionary associated with the *SO* data. Note that in the scope of this work we did not investigate the contribution of each of the background projects used in *OUT*, and how their relevance to the target prediction project effects their performance.

The trigram model shows the best performance across all training data sources (47% for *IN*, 32% for *OUT* and 34% for *SO*). Amongst the tested topic models, link-LDA models which distinguish *code* and *text* tokens perform consistently better than simple LDA models in which all tokens are considered as text. We did not however find a correlation between the number of latent topics learned by a topic model and its performance. In fact, for each of the data sources, a different number of topics gave the optimal character saving results.

Note that in this work, all topic models are based on unigram tokens, therefore their results are most comparable with that of the unigram in

| Dataset    | $n$ -gram | link-LDA |
|------------|-----------|----------|
| <i>IN</i>  | 2778.35   | 574.34   |
| <i>OUT</i> | 1865.67   | 670.34   |
| <i>SO</i>  | 1898.43   | 638.55   |

Table 3: Average words per project for which each tested model completes the word better than the other. This indicates that each of the models is better at predicting a different set of comment words.

Table 1, which does not benefit from the back-off strategy used by the bigram and trigram models. By this comparison, the link-LDA topic model proves more successful in the comment prediction task than the simpler models which do not distinguish *code* and *text* tokens. Using  $n$ -grams without backoff leads to results significantly worse than any of the presented models (not shown).

Table 2 shows a sample comment segment for which words were predicted using trigram models from all training sources and an in-project link-LDA. The comment is taken from the *TrainExtractor* class in the *Minor-Third* project, a machine learning library for annotating and categorizing text. Both *IN* models show a clear advantage in completing the project-specific word *Train*, compared to models based on out-of-project data (*OUT* and *SO*). Interestingly, in this example the trigram is better at completing the term *named-entity* given the prefix *named*. However, the topic model is better at completing the word *extractor* which refers to the target class. This example indicates that each model type may be more successful in predicting different comment words, and that combining multiple models may be advantageous.

This can also be seen by the analysis in Table 3 where we compare the average number of words completed better by either the best n-gram or topic model given each training dataset. Again, while n-grams generally complete more words better, a considerable portion of the words is better completed using a topic model, further motivating a hybrid solution.

## 5 Conclusions

We analyze the use of language models for predicting class comments for source file documents containing a mixture of *code* and *text* tokens. Our experiments demonstrate the effectiveness of using language models for comment completion, showing a saving of up to 47% of the comment characters. When available, using in-project training data proves significantly more successful than using out-of-project data. However, we find that when using out-of-project data, a dataset based on more words than code performs consistently better. The results also show that different models are better at predicting different comment words, which motivates a hybrid solution combining the advantages of multiple models.

## Acknowledgments

This research was supported by the NSF under grant CCF-1247088.

## References

- Ramnath Balasubramanian and William W Cohen. 2011. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *Proceedings of the 7th SIAM International Conference on Data Mining*.
- Dave Binkley, Matthew Hearn, and Dawn Lawrie. 2011. Improving identifier informativeness using part of speech information. In *Proc. of the Working Conference on Mining Software Repositories*. ACM.
- David M Blei and Michael I Jordan. 2003. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*.
- Elena Erosheva, Stephen Fienberg, and John Lafferty. 2004. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*.
- Yansong Feng and Mirella Lapata. 2010. How many words is a picture worth? automatic caption generation for news images. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Yansong Feng and Mirella Lapata. 2013. Automatic caption generation for news images. *IEEE transactions on pattern analysis and machine intelligence*.
- Mark Gabel and Zhendong Su. 2008. Javert: fully automatic mining of general temporal properties from dynamic traces. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pages 339–349. ACM.
- Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proc. of the National Academy of Sciences of the United States of America*.
- Sangmok Han, David R Wallace, and Robert C Miller. 2009. Code completion from abbreviated input. In *Automated Software Engineering, 2009. ASE'09. 24th IEEE/ACM International Conference on*, pages 332–343. IEEE.
- Abram Hindle, Earl T Barr, Zhendong Su, Mark Gabel, and Premkumar Devanbu. 2012. On the naturalness of software. In *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE.
- Ferosh Jacob and Robert Tairas. 2010. Code template inference using language models. In *Proceedings of the 48th Annual Southeast Regional Conference*. ACM.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95.*, volume 1, pages 181–184. IEEE.
- Naveen Kumar and Benjamin Carterette. 2013. Time based feedback and query expansion for twitter search. In *Advances in Information Retrieval*, pages 734–737. Springer.
- Dawn Lawrie, Christopher Morrell, Henry Feild, and David Binkley. 2006. Whats in a name? a study of identifiers. In *Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on*, pages 3–12. IEEE.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit.
- 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*, volume 1, pages 258–267.
- Sarah Rastkar, Gail C Murphy, and Alexander WJ Bradley. 2011. Generating natural language summaries for crosscutting source code concerns. In *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*, pages 103–112. IEEE.

- Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.
- David Shepherd, Zachary P Fry, Emily Hill, Lori Pollock, and K Vijay-Shanker. 2007. Using natural language program analysis to locate and understand action-oriented concerns. In *Proceedings of the 6th international conference on Aspect-oriented software development*, pages 212–224. ACM.
- Giriprasad Sridhara, Emily Hill, Divya Muppaneni, Lori Pollock, and K Vijay-Shanker. 2010. Towards automatically generating summary comments for java methods. In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, pages 43–52. ACM.
- Yuen-Hsien Tseng and Da-Wei Juang. 2003. Document-self expansion for text categorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 399–400. ACM.
- Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Single document summarization with document expansion. In *Proc. of the National Conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Roung-Shiunn Wu and Po-Chun Li. 2011. Video annotation using hierarchical dirichlet process mixture model. *Expert Systems with Applications*, 38(4):3040–3048.



# Paraphrasing Adaptation for Web Search Ranking

**Chenguang Wang\***

School of EECS

Peking University

wangchenguang@pku.edu.cn

**Ming Zhou**

Microsoft Research Asia

mingzhou@microsoft.com

**Nan Duan**

Microsoft Research Asia

nanduan@microsoft.com

**Ming Zhang**

School of EECS

Peking University

mzhang@net.pku.edu.cn

## Abstract

Mismatch between queries and documents is a key issue for the web search task. In order to narrow down such mismatch, in this paper, we present an in-depth investigation on adapting a paraphrasing technique to web search from three aspects: a search-oriented paraphrasing model; an NDCG-based parameter optimization algorithm; an enhanced ranking model leveraging augmented features computed on paraphrases of original queries. Experiments performed on the large scale query-document data set show that, the search performance can be significantly improved, with +3.28% and +1.14% NDCG gains on dev and test sets respectively.

## 1 Introduction

Paraphrasing is an NLP technique that generates alternative expressions to convey the same meaning of the input text in different ways. Researchers have made great efforts to improve paraphrasing from different perspectives, such as paraphrase extraction (Zhao et al., 2007), paraphrase generation (Quirk et al., 2004), model optimization (Zhao et al., 2009) and etc. But as far as we know, none of previous work has explored the impact of using a well designed paraphrasing engine for web search ranking task specifically.

In web search, mismatches between queries and their relevant documents are usually caused by expressing the same meaning in different natural language ways. E.g., *X is the author of Y* and *Y was written by X* have identical meaning in most cases, but they are quite different in literal sense. The capability of paraphrasing is just right to alleviate such issues. Motivated by this, this paper presents

---

\* This work has been done while the author was visiting Microsoft Research Asia.

an in-depth study on adapting paraphrasing to web search. First, we propose a search-oriented paraphrasing model, which includes specifically designed features for web queries that can enable a paraphrasing engine to learn preferences on different paraphrasing strategies. Second, we optimize the parameters of the paraphrasing model according to the Normalized Discounted Cumulative Gain (NDCG) score, by leveraging the minimum error rate training (MERT) algorithm (Och, 2003). Third, we propose an enhanced ranking model by using augmented features computed on paraphrases of original queries.

Many query reformulation approaches have been proposed to tackle the query-document mismatch issue, which can be generally summarized as query expansion and query substitution. Query expansion (Baeza-Yates, 1992; Jing and Croft, 1994; Lavrenko and Croft, 2001; Cui et al., 2002; Yu et al., 2003; Zhang and Yu, 2006; Craswell and Szummer, 2007; Elsas et al., 2008; Xu et al., 2009) adds new terms extracted from different sources to the original query directly; while query substitution (Brill and Moore, 2000; Jones et al., 2006; Guo et al., 2008; Wang and Zhai, 2008; Dang and Croft, 2010) uses probabilistic models, such as graphical models, to predict the sequence of rewritten query words to form a new query. Comparing to these works, our paraphrasing engine alters queries in a similar way to statistical machine translation, with systematic tuning and decoding components. Zhao et al. (2009) proposes a unified paraphrasing framework that can be adapted to different applications using different usability models. Our work can be seen as an extension along this line of research, by carrying out in-depth study on adapting paraphrasing to web search.

Experiments performed on the large scale data set show that, by leveraging additional matching features computed on query paraphrases, significant NDCG gains can be achieved on both dev

(+3.28%) and test (+1.14%) sets.

## 2 Paraphrasing for Web Search

In this section, we first summarize our paraphrase extraction approaches, and then describe our paraphrasing engine for the web search task from three aspects, including: 1) a search-oriented paraphrasing model; 2) an NDCG-based parameter optimization algorithm; 3) an enhanced ranking model with augmented features that are computed based on the extra knowledge provided by the paraphrase candidates of the original queries.

### 2.1 Paraphrase Extraction

Paraphrases can be mined from various resources. Given a bilingual corpus, we use Bannard and Callison-Burch (2005)’s pivot-based approach to extract paraphrases. Given a monolingual corpus, Lin and Pantel (2001)’s method is used to extract paraphrases based on distributional hypothesis. Additionally, human annotated data can also be used as high-quality paraphrases. We use Miller (1995)’s approach to extract paraphrases from the synonym dictionary of WordNet. Word alignments within each paraphrase pair are generated using GIZA++ (Och and Ney, 2000).

### 2.2 Search-Oriented Paraphrasing Model

Similar to statistical machine translation (SMT), given an input query  $Q$ , our paraphrasing engine generates paraphrase candidates<sup>1</sup> based on a linear model.

$$\begin{aligned}\hat{Q} &= \arg \max_{Q' \in \mathcal{H}(Q)} P(Q'|Q) \\ &= \arg \max_{Q' \in \mathcal{H}(Q)} \sum_{m=1}^M \lambda_m h_m(Q, Q')\end{aligned}$$

$\mathcal{H}(Q)$  is the hypothesis space containing all paraphrase candidates of  $Q$ ,  $h_m$  is the  $m^{th}$  feature function with weight  $\lambda_m$ ,  $Q'$  denotes one candidate. In order to enable our paraphrasing model to learn the preferences on different paraphrasing strategies according to the characteristics of web queries, we design search-oriented features<sup>2</sup> based on word alignments within  $Q$  and  $Q'$ , which can be described as follows:

<sup>1</sup>We apply CYK algorithm (Chappelier and Rajman, 1998), which is most commonly used in SMT (Chiang, 2005), to generating paraphrase candidates.

<sup>2</sup>Similar features have been demonstrated effective in (Jones et al., 2006). But we use SMT-like model to generate query reformulations.

- Word Addition feature  $h_{WADD}(Q, Q')$ , which is defined as the number of words in the paraphrase candidate  $Q'$  without being aligned to any word in the original query  $Q$ .
- Word Deletion feature  $h_{WDEL}(Q, Q')$ , which is defined as the number of words in the original query  $Q$  without being aligned to any word in the paraphrase candidate  $Q'$ .
- Word Overlap feature  $h_{WCO}(Q, Q')$ , which is defined as the number of word pairs that align identical words between  $Q$  and  $Q'$ .
- Word Alteration feature  $h_{WA}(Q, Q')$ , which is defined as the number of word pairs that align different words between  $Q$  and  $Q'$ .
- Word Reorder feature  $h_{WR}(Q, Q')$ , which is modeled by a relative distortion probability distribution, similar to the distortion model in (Koehn et al., 2003).
- Length Difference feature  $h_{LD}(Q, Q')$ , which is defined as  $|Q'| - |Q|$ .
- Edit Distance feature  $h_{ED}(Q, Q')$ , which is defined as the character-level edit distance between  $Q$  and  $Q'$ .

Besides, a set of traditional SMT features (Koehn et al., 2003) are also used in our paraphrasing model, including translation probability, lexical weight, word count, paraphrase rule count<sup>3</sup>, and language model feature.

### 2.3 NDCG-based Parameter Optimization

We utilize minimum error rate training (MERT) (Och, 2003) to optimize feature weights of the paraphrasing model according to NDCG. We define  $\mathcal{D}$  as the entire document set.  $\mathcal{R}$  is a ranking model<sup>4</sup> that can rank documents in  $\mathcal{D}$  based on each input query.  $\{Q_i, \mathcal{D}_i^{Label}\}_{i=1}^S$  is a human-labeled development set.  $Q_i$  is the  $i^{th}$  query and  $\mathcal{D}_i^{Label} \subset \mathcal{D}$  is a subset of documents, in which the relevance between  $Q_i$  and each document is labeled by human annotators.

MERT is used to optimize feature weights of our linear-formed paraphrasing model. For

<sup>3</sup>Paraphrase rule count is the number of rules that are used to generate paraphrase candidates.

<sup>4</sup>The ranking model  $\mathcal{R}$  (Liu et al., 2007) uses matching features computed based on original queries and documents.

each query  $Q_i$  in  $\{Q_i\}_{i=1}^S$ , we first generate  $N$ -best paraphrase candidates  $\{Q_i^j\}_{j=1}^N$ , and compute NDCG score for each paraphrase based on documents ranked by the ranker  $\mathcal{R}$  and labeled documents  $\mathcal{D}_i^{Label}$ . We then optimize the feature weights according to the following criterion:

$$\hat{\lambda}_1^M = \arg \min_{\lambda_1^M} \left\{ \sum_{i=1}^S Err(\mathcal{D}_i^{Label}, \hat{Q}_i; \lambda_1^M, \mathcal{R}) \right\}$$

The objective of MERT is to find the optimal feature weight vector  $\hat{\lambda}_1^M$  that minimizes the error criterion  $Err$  according to the NDCG scores of top-1 paraphrase candidates.

The error function  $Err$  is defined as:

$$Err(\mathcal{D}_i^{Label}, \hat{Q}_i; \lambda_1^M, \mathcal{R}) = 1 - \mathcal{N}(\mathcal{D}_i^{Label}, \hat{Q}_i, \mathcal{R})$$

where  $\hat{Q}_i$  is the best paraphrase candidate according to the paraphrasing model based on the weight vector  $\lambda_1^M$ ,  $\mathcal{N}(\mathcal{D}_i^{Label}, \hat{Q}_i, \mathcal{R})$  is the NDCG score of  $\hat{Q}_i$  computed on the documents ranked by  $\mathcal{R}$  of  $\hat{Q}_i$  and labeled document set  $\mathcal{D}_i^{Label}$  of  $Q_i$ . The relevance rating labeled by human annotators can be represented by five levels: ‘‘Perfect’’, ‘‘Excellent’’, ‘‘Good’’, ‘‘Fair’’, and ‘‘Bad’’. When computing NDCG scores, these five levels are commonly mapped to the numerical scores 31, 15, 7, 3, 0 respectively.

## 2.4 Enhanced Ranking Model

In web search, the key objective of the ranking model is to rank the retrieved documents based on their relevance to a given query.

Given a query  $Q$  and its retrieved document set  $\mathbf{D} = \{D_Q\}$ , for each  $D_Q \in \mathbf{D}$ , we use the following ranking model to compute their relevance, which is formulated as a weighted combination of matching features:

$$\mathcal{R}(Q, D_Q) = \sum_{k=1}^K \lambda_k F_k(Q, D_Q)$$

$\mathbf{F} = \{F_1, \dots, F_K\}$  denotes a set of matching features that measure the matching degrees between  $Q$  and  $D_Q$ ,  $F_k(Q, D_Q) \in \mathbf{F}$  is the  $k^{th}$  matching feature,  $\lambda_k$  is its corresponding feature weight.

How to learn the weight vector  $\{\lambda_k\}_{k=1}^K$  is a standard learning-to-rank task. The goal of learning is to find an optimal weight vector  $\{\hat{\lambda}_k\}_{k=1}^K$ , such that for any two documents  $D_Q^i \in \mathbf{D}$  and  $D_Q^j \in \mathbf{D}$ , the following condition holds:

$$\mathcal{R}(Q, D_Q^i) > \mathcal{R}(Q, D_Q^j) \Leftrightarrow r_{D_Q^i} > r_{D_Q^j}$$

where  $r_{D_Q}$  denotes a numerical relevance rating labeled by human annotators denoting the relevance between  $Q$  and  $D_Q$ .

As the ultimate goal of improving paraphrasing is to help the search task, we present a straightforward but effective method to enhance the ranking model  $\mathcal{R}$  described above, by leveraging paraphrase candidates of the original query as the extra knowledge to compute matching features.

Formally, given a query  $Q$  and its  $N$ -best paraphrase candidates  $\{Q'_1, \dots, Q'_N\}$ , we enrich the original feature vector  $\mathbf{F}$  to  $\{\mathbf{F}, \mathbf{F}_1, \dots, \mathbf{F}_N\}$  for  $Q$  and  $D_Q$ , where all features in  $\mathbf{F}_n$  have the same meanings as they are in  $\mathbf{F}$ , however, their feature values are computed based on  $Q'_n$  and  $D_Q$ , instead of  $Q$  and  $D_Q$ . In this way, the paraphrase candidates act as hidden variables and expanded matching features between queries and documents, making our ranking model more tunable and flexible for web search.

## 3 Experiment

### 3.1 Data and Metric

Paraphrase pairs are extracted as we described in Section 2.1. The bilingual corpus includes 5.1M sentence pairs from the NIST 2008 constrained track of Chinese-to-English machine translation task. The monolingual corpus includes 16.7M queries from the log of a commercial search engine. Human annotated data contains 0.3M synonym pairs from WordNet dictionary. Word alignments of each paraphrase pair are trained by GIZA++. The language model is trained based on a portion of queries, in which the frequency of each query is higher than a predefined threshold, 5. The number of paraphrase pairs is 58M. The minimum length of paraphrase rule is 1, while the maximum length of paraphrase rule is 5.

We randomly select 2, 838 queries from the log of a commercial search engine, each of which attached with a set of documents that are annotated with relevance ratings described in Section 2.3. We use the first 1, 419 queries together with their annotated documents as the development set to tune paraphrasing parameters (as we discussed in Section 2.3), and use the rest as the test set. The ranking model is trained based on the development set. NDCG is used as the evaluation metric of the web search task.

### 3.2 Baseline Systems

The baselines of the paraphrasing and the ranking model are described as follows:

The paraphrasing baseline is denoted as **BL-Para**, which only uses traditional SMT features described at the end of Section 2.2. Weights are optimized by MERT using BLEU (Papineni et al., 2002) as the error criterion. Development data are generated based on the English references of NIST 2008 constrained track of Chinese-to-English machine translation task. We use the first reference as the source, and the rest as its paraphrases.

The ranking model baseline (Liu et al., 2007) is denoted as **BL-Rank**, which only uses matching features computed based on original queries and different meta-streams of web pages, including URL, page title, page body, meta-keywords, meta-description and anchor texts. The feature functions we use include unigram/bigram/trigram BM25 and original/normalized Perfect-Match. The ranking model is learned based on  $SVM^{rank}$  toolkit (Joachims, 2006) with default parameter setting.

### 3.3 Impacts of Search-Oriented Features

We first evaluate the effectiveness of the search-oriented features. To do so, we add these features into the paraphrasing model baseline, and denote it as **BL-Para+SF**, whose weights are optimized in the same way with BL-Para. The ranking model baseline BL-Rank is used to rank the documents. We then compare the NDCG@1 scores of the best documents retrieved using either original query, or query paraphrases generated by BL-Para and BL-Para+SF respectively, and list comparison results in Table 1, where Cand@1 denotes the best paraphrase candidate generated by each paraphrasing model.

| Test Set       |                |                   |
|----------------|----------------|-------------------|
|                | <b>BL-Para</b> | <b>BL-Para+SF</b> |
| Original Query | Cand@1         | Cand@1            |
| 27.28%         | 26.44%         | 26.53%            |

Table 1: Impacts of search-oriented features.

From Table 1, we can see, even using the best query paraphrase, its corresponding NDCG score is still lower than the NDCG score of the original query. This performance dropping makes sense, as changing user queries brings the risks of query drift. When adding search-oriented features into the baseline, the performance changes little, as these two models are optimized based on BLEU

score only, without considering characteristics of mismatches in search.

### 3.4 Impacts of Optimization Algorithm

We then evaluate the impact of our NDCG-based optimization method. We add the optimization algorithm described in Section 2.3 into BL-Para+SF, and get a paraphrasing model **BL-Para+SF+Opt**. The ranking model baseline BL-Rank is used. Similar to the experiment in Table 1, we compare the NDCG@1 scores of the best documents retrieved using query paraphrases generated by BL-Para+SF and BL-Para+SF+Opt respectively, with results shown in Table 2.

| Test Set       |                   |                       |
|----------------|-------------------|-----------------------|
|                | <b>BL-Para+SF</b> | <b>BL-Para+SF+Opt</b> |
| Original Query | Cand@1            | Cand@1                |
| 27.28%         | 26.53%            | 27.06%(+0.53%)        |

Table 2: Impacts of NDCG-based optimization.

Table 2 indicates that, by leveraging NDCG as the error criterion for MERT, search-oriented features benefit more (+0.53% NDCG) in selecting the best query paraphrase from the whole paraphrasing search space. The improvement is statistically significant ( $p < 0.001$ ) by t-test (Smucker et al., 2007). The quality of the top-1 paraphrase generated by BL-Para+SF+Opt is very close to the original query.

### 3.5 Impacts of Enhanced Ranking Model

We last evaluate the effectiveness of the enhanced ranking model. The ranking model baseline BL-Rank only uses original queries to compute matching features between queries and documents; while the enhanced ranking model, denoted as **BL-Rank+Para**, uses not only the original query but also its top-1 paraphrase candidate generated by BL-Para+SF+Opt to compute augmented matching features described in Section 2.4.

| Dev Set             |                |                |
|---------------------|----------------|----------------|
|                     | NDCG@1         | NDCG@5         |
| <b>BL-Rank</b>      | 25.31%         | 33.76%         |
| <b>BL-Rank+Para</b> | 28.59%(+3.28%) | 34.25%(+0.49%) |
| Test Set            |                |                |
|                     | NDCG@1         | NDCG@5         |
| <b>BL-Rank</b>      | 27.28%         | 34.79%         |
| <b>BL-Rank+Para</b> | 28.42%(+1.14%) | 35.68%(+0.89%) |

Table 3: Impacts of enhanced ranking model.

From Table 3, we can see that NDCG@ $k$  ( $k = 1, 5$ ) scores of BL-Rank+Para outperforms BL-Rank on both dev and test sets. T-test shows that

the improvement is statistically significant ( $p < 0.001$ ). Such end-to-end NDCG improvements come from the extra knowledge provided by the hidden paraphrases of original queries. This narrows down the query-document mismatch issue to a certain extent.

#### 4 Conclusion and Future Work

In this paper, we present an in-depth study on using paraphrasing for web search, which pays close attention to various aspects of the application including choice of model and optimization technique. In the future, we will compare and combine paraphrasing with other query reformulation techniques, e.g., pseudo-relevance feedback (Yu et al., 2003) and a conditional random field-based approach (Guo et al., 2008).

#### Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC Grant No. 61272343) as well as the Doctoral Program of Higher Education of China (FSSP Grant No. 20120001110112).

#### References

- Ricardo A Baeza-Yates. 1992. Introduction to data structures and algorithms related to information retrieval.
- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*, pages 597–604.
- Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of ACL*, pages 286–293.
- Jean-Cédric Chappelier and Martin Rajman. 1998. A generalized cyk algorithm for parsing stochastic cfg. In *Workshop on Tabulation in Parsing and Deduction*, pages 133–137.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270.
- Nick Craswell and Martin Szummer. 2007. Random walks on the click graph. In *Proceedings of SIGIR, SIGIR '07*, pages 239–246.
- Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of WWW*, pages 325–332.
- Van Dang and Bruce W. Croft. 2010. Query reformulation using anchor text. In *Proceedings of WSDM*, pages 41–50.
- Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. 2008. Retrieval and feedback models for blog feed search. In *Proceedings of SIGIR*, pages 347–354.
- Jiafeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. 2008. A unified and discriminative model for query refinement. In *Proceedings of SIGIR, SIGIR '08*, pages 379–386.
- Yufeng Jing and W. Bruce Croft. 1994. An association thesaurus for information retrieval. In *In RIAO 94 Conference Proceedings*, pages 146–160.
- Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of KDD*, pages 217–226.
- Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of WWW*, pages 387–396.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54.
- Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of SIGIR*, pages 120–127.
- Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, pages 343–360.
- Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. 2007. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR workshop*, pages 3–10.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, pages 39–41.
- Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, pages 142–149.
- Mark D Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of CIKM*, pages 623–632.

- Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, Proceedings of CIKM, pages 479–488.
- Yang Xu, Gareth J.F. Jones, and Bin Wang. 2009. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of SIGIR*, pages 59–66.
- Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of WWW*, pages 11–18.
- Wei Zhang and Clement Yu. 2006. Uic at trec 2006 blog track. In *Proceedings of TREC*.
- Shiqi Zhao, Ming Zhou, and Ting Liu. 2007. Learning question paraphrases for qa from encarta logs. In *Proceedings of IJCAI*, pages 1795–1800.
- Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of ACL*, pages 834–842.

# Semantic Parsing as Machine Translation

**Jacob Andreas**

Computer Laboratory  
University of Cambridge  
jda33@cam.ac.uk

**Andreas Vlachos**

Computer Laboratory  
University of Cambridge  
av308@cam.ac.uk

**Stephen Clark**

Computer Laboratory  
University of Cambridge  
sc609@cam.ac.uk

## Abstract

Semantic parsing is the problem of deriving a structured meaning representation from a natural language utterance. Here we approach it as a straightforward machine translation task, and demonstrate that standard machine translation components can be adapted into a semantic parser. In experiments on the multilingual GeoQuery corpus we find that our parser is competitive with the state of the art, and in some cases achieves higher accuracy than recently proposed purpose-built systems. These results support the use of machine translation methods as an informative baseline in semantic parsing evaluations, and suggest that research in semantic parsing could benefit from advances in machine translation.

## 1 Introduction

Semantic parsing (SP) is the problem of transforming a natural language (NL) utterance into a machine-interpretable meaning representation (MR). It is well-studied in NLP, and a wide variety of methods have been proposed to tackle it, e.g. rule-based (Popescu et al., 2003), supervised (Zelle, 1995), unsupervised (Goldwasser et al., 2011), and response-based (Liang et al., 2011).

At least superficially, SP is simply a machine translation (MT) task: we transform an NL utterance in one language into a statement of another (un-natural) meaning representation language (MRL). Indeed, successful semantic parsers often resemble MT systems in several important respects, including the use of word alignment models as a starting point for rule extraction (Wong and Mooney, 2006; Kwiatkowski et al., 2010) and the use of automata such as tree transducers (Jones et al., 2012) to encode the relationship between NL and MRL.

The key difference between the two tasks is that in SP, the target language (the MRL) has very different properties to an NL. In particular, MRs must conform strictly to a particular structure so that they are machine-interpretable. Contrast this with ordinary MT, where varying degrees of wrongness are tolerated by human readers (and evaluation metrics). To avoid producing malformed MRs, almost all of the existing research on SP has focused on developing models with richer structure than those commonly used for MT.

In this work we attempt to determine how accurate a semantic parser we can build by treating SP as a pure MT task, and describe pre- and post-processing steps which allow structure to be preserved in the MT process.

Our contributions are as follows: We develop a semantic parser using off-the-shelf MT components, exploring phrase-based as well as hierarchical models. Experiments with four languages on the popular GeoQuery corpus (Zelle, 1995) show that our parser is competitive with the state-of-the-art, in some cases achieving higher accuracy than recently introduced purpose-built semantic parsers. Our approach also appears to require substantially less time to train than the two best-performing semantic parsers. These results support the use of MT methods as an informative baseline in SP evaluations and show that research in SP could benefit from research advances in MT.

## 2 MT-based semantic parsing

The input is a corpus of NL utterances paired with MRs. In order to learn a semantic parser using MT we linearize the MRs, learn alignments between the MRL and the NL, extract translation rules, and learn a language model for the MRL. We also specify a decoding procedure that will return structured MRs for an utterance during prediction.

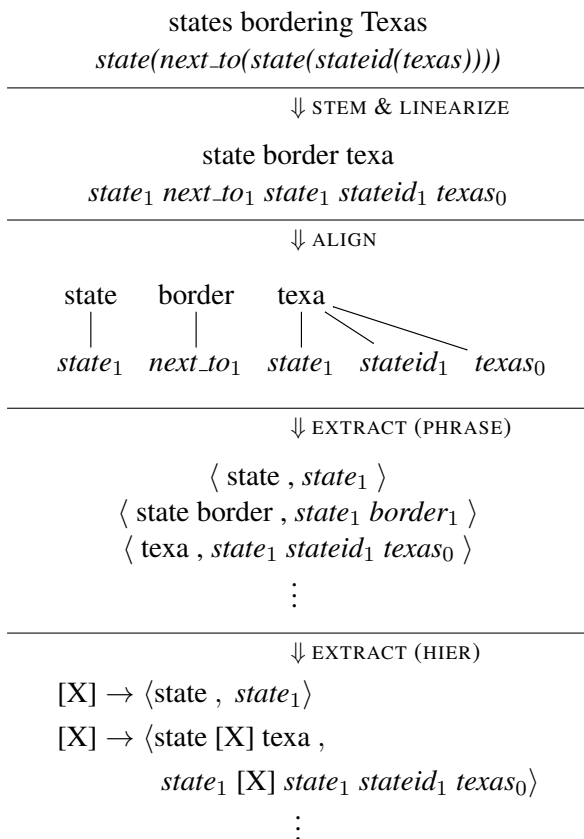


Figure 1: Illustration of preprocessing and rule extraction.

**Linearization** We assume that the MRL is variable-free (that is, the meaning representation for each utterance is tree-shaped), noting that formalisms with variables, like the  $\lambda$ -calculus, can be mapped onto variable-free logical forms with combinatory logics (Curry et al., 1980).

In order to learn a semantic parser using MT we begin by converting these MRs to a form more similar to NL. To do so, we simply take a preorder traversal of every functional form, and label every function with the number of arguments it takes. After translation, recovery of the function is easy: if the arity of every function in the MRL is known, then every traversal uniquely specifies its corresponding tree. Using an example from GeoQuery, given an input function of the form

$answer(population(city(cityid('seattle', 'wa'))))$

we produce a “decorated” translation input of the form

$answer_1 population_1 city_1 cityid_2 seattle_0 wa_0$

where each subscript indicates the symbol’s arity (constants, including strings, are treated as zero-argument functions). Explicit argument number

labeling serves two functions. Most importantly, it eliminates any possible ambiguity from the tree reconstruction which takes place during decoding: given any sequence of decorated MRL tokens, we can always reconstruct the corresponding tree structure (if one exists). Arity labeling additionally allows functions with variable numbers of arguments (e.g. *cityid*, which in some training examples is unary) to align with different natural language strings depending on context.

**Alignment** Following the linearization of the MRs, we find alignments between the MR tokens and the NL tokens using the IBM Model 4 (Brown et al., 1993). Once the alignment algorithm is run in both directions (NL to MRL, MRL to NL), we symmetrize the resulting alignments to obtain a consensus many-to-many alignment (Och and Ney, 2000; Koehn et al., 2005).

**Rule extraction** From the many-to-many alignment we need to extract a translation rule table, consisting of corresponding phrases in NL and MRL. We consider a phrase-based translation model (Koehn et al., 2003) and a hierarchical translation model (Chiang, 2005). Rules for the phrase-based model consist of pairs of aligned source and target sequences, while hierarchical rules are SCFG productions containing at most two instances of a single nonterminal symbol.

Note that both extraction algorithms can learn rules which a traditional tree-transducer-based approach cannot—for example the right hand side

$[X] river_1 all_0 traverse_1 [X]$

corresponding to the pair of disconnected tree fragments:

$$\begin{array}{cc} [X] & traverse \\ \downarrow & \downarrow \\ river & [X] \\ \downarrow & \\ all & \end{array}$$

(where each  $X$  indicates a gap in the rule).

**Language modeling** In addition to translation rules learned from a parallel corpus, MT systems also rely on an  $n$ -gram language model for the target language, estimated from a (typically larger) monolingual corpus. In the case of SP, such a monolingual corpus is rarely available, and we instead use the MRs available in the training data to learn a language model of the MRL. This information helps guide the decoder towards well-formed



structures; it encodes, for example, the preferences of predicates of the MRL for certain arguments.

**Prediction** Given a new NL utterance, we need to find the  $n$  best translations (i.e. sequences of decorated MRL tokens) that maximize the weighted sum of the translation score (the probabilities of the translations according to the rule translation table) and the language model score, a process usually referred to as decoding. Standard decoding procedures for MT produce an  $n$ -best list of all possible translations, but here we need to restrict ourselves to translations corresponding to well-formed MRs. In principle this could be done by re-writing the beam search algorithm used in decoding to immediately discard malformed MRs; for the experiments in this paper we simply filter the regular  $n$ -best list until we find a well-formed MR. This filtering can be done with time linear in the length of the example by exploiting the argument label numbers introduced during linearization. Finally, we insert the brackets according to the tree structure specified by the argument number labels.

### 3 Experimental setup

**Dataset** We conduct experiments on the GeoQuery data set. The corpus consists of a set of 880 natural-language questions about U.S. geography in four languages (English, German, Greek and Thai), and their representations in a variable-free MRL that can be executed against a Prolog database interface. Initial experimentation was done using 10 fold cross-validation on the 600-sentence development set and the final evaluation on a held-out test set of 280 sentences. All semantic parsers for GeoQuery we compare against also makes use of *NP lists* (Jones et al., 2012), which contain MRs for every noun phrase that appears in the NL utterances of each language. In our experiments, the NP list was included by appending all entries as extra training sentences to the end of the training corpus of each language with 50 times the weight of regular training examples, to ensure that they are learned as translation rules.

Evaluation for each utterance is performed by executing both the predicted and the gold standard MRs against the database and obtaining their respective answers. An MR is correct if it obtains the same answer as the gold standard MR, allowing for a fair comparison between systems using different learning paradigms. Following Jones et

al. (2012) we report accuracy, i.e. the percentage of NL questions with correct answers, and  $F_1$ , i.e. the harmonic mean of precision (percentage of correct answers obtained).

**Implementation** In all experiments, we use the IBM Model 4 implementation from the GIZA++ toolkit (Och and Ney, 2000) for alignment, and the phrase-based and hierarchical models implemented in the Moses toolkit (Koehn et al., 2007) for rule extraction. The best symmetrization algorithm, translation and language model weights for each language are selected using cross-validation on the development set. In the case of English and German, we also found that stemming (Bird et al., 2009; Porter, 1980) was helpful in reducing data sparsity.

## 4 Results

We first compare the results for the two translation rule extraction models, phrase-based and hierarchical (“MT-phrase” and “MT-hier” respectively in Table 1). We find that the hierarchical model performs better in all languages apart from Greek, indicating that the long-range reorderings learned by a hierarchical translation system are useful for this task. These benefits are most pronounced in the case of Thai, likely due to the the language’s comparatively different word order.

We also present results for both models without using the NP lists for training in Table 2. As expected, the performances are almost uniformly lower, but the parser still produces correct output for the majority of examples.

As discussed above, one important modification of the MT paradigm which allows us to produce structured output is the addition of structure-checking to the beam search. It is not evident, *a priori*, that this search procedure is guaranteed to find any well-formed outputs in reasonable time; to test the effect of this extra requirement on

|                 | en   | de   | el   | th   |
|-----------------|------|------|------|------|
| MT-phrase       | 75.3 | 68.8 | 70.4 | 53.0 |
| MT-phrase (-NP) | 63.4 | 65.8 | 64.0 | 39.8 |
| MT-hier         | 80.5 | 68.9 | 69.1 | 70.4 |
| MT-hier (-NP)   | 62.5 | 69.9 | 62.9 | 62.1 |

Table 2: GeoQuery accuracies with and without NPs. Rows with (-NP) did not use the NP list.

|             | English [en] |                | German [de] |                | Greek [el] |                | Thai [th] |                |
|-------------|--------------|----------------|-------------|----------------|------------|----------------|-----------|----------------|
|             | Acc.         | F <sub>1</sub> | Acc.        | F <sub>1</sub> | Acc.       | F <sub>1</sub> | Acc.      | F <sub>1</sub> |
| WASP        | 71.1         | 77.7           | 65.7        | 74.9           | 70.7       | 78.6           | 71.4      | 75.0           |
| UBL         | 82.1         | 82.1           | 75.0        | 75.0           | 73.6       | 73.7           | 66.4      | 66.4           |
| tsVB        | 79.3         | 79.3           | 74.6        | 74.6           | 75.4       | 75.4           | 78.2      | 78.2           |
| hybrid-tree | 76.8         | 81.0           | 62.1        | 68.5           | 69.3       | 74.6           | 73.6      | 76.7           |
| MT-phrase   | 75.3         | 75.8           | 68.8        | 70.8           | 70.4       | 73.0           | 53.0      | 54.4           |
| MT-hier     | 80.5         | 81.8           | 68.9        | 71.8           | 69.1       | 72.3           | 70.4      | 70.7           |

Table 1: Accuracy and F<sub>1</sub> scores for the multilingual GeoQuery test set. Results for other systems as reported by Jones et al. (2012).

the speed of SP, we investigate how many MRs the decoder needs to generate before producing one which is well-formed. In practice, increasing search depth in the  $n$ -best list from 1 to 50 results in a gain of no more than a percentage point or two, and we conclude that our filtering method is appropriate for the task.

We also compare the MT-based semantic parsers to several recently published ones: WASP (Wong and Mooney, 2006), which like the hierarchical model described here learns a SCFG to translate between NL and MRL; tsVB (Jones et al., 2012), which uses variational Bayesian inference to learn weights for a tree transducer; UBL (Kwiatkowski et al., 2010), which learns a CCG lexicon with semantic annotations; and hybrid-tree (Lu et al., 2008), which learns a synchronous generative model over variable-free MRs and NL strings.

In the results shown in Table 1 we observe that on English GeoQuery data, the hierarchical translation model achieves scores competitive with the state of the art, and in every language one of the MT systems achieves accuracy at least as good as a purpose-built semantic parser.

We conclude with an informal test of training speeds. While differences in implementation and factors like programming language choice make a direct comparison of times necessarily imprecise, we note that the MT system takes less than three minutes to train on the GeoQuery corpus, while the publicly-available implementations of tsVB and UBL require roughly twenty minutes and five hours respectively on a 2.1 GHz CPU. So in addition to competitive performance, the MT-based parser also appears to be considerably more efficient at training time than other parsers in the literature.

## 5 Related Work

WASP, an early automatically-learned SP system, was strongly influenced by MT techniques. Like the present work, it uses GIZA++ alignments as a starting point for the rule extraction procedure, and algorithms reminiscent of those used in syntactic MT to extract rules.

tsVB also uses a piece of standard MT machinery, specifically tree transducers, which have been profitably employed for syntax-based machine translation (Maletti, 2010). In that work, however, the usual MT parameter-estimation technique of simply counting the number of rule occurrences does not improve scores, and the authors instead resort to a variational inference procedure to acquire rule weights. The present work is also the first we are aware of which uses phrase-based rather than tree-based machine translation techniques to learn a semantic parser. hybrid-tree (Lu et al., 2008) similarly describes a generative model over derivations of MRL trees.

The remaining system discussed in this paper, UBL (Kwiatkowski et al., 2010), leverages the fact that the MRL does not simply encode trees, but rather  $\lambda$ -calculus expressions. It employs resolution procedures specific to the  $\lambda$ -calculus such as splitting and unification in order to generate rule templates. Like other systems described, it uses GIZA alignments for initialization. Other work which generalizes from variable-free meaning representations to  $\lambda$ -calculus expressions includes the natural language generation procedure described by Lu and Ng (2011).

UBL, like an MT system (and unlike most of the other systems discussed in this section), extracts rules at multiple levels of granularity by means of this splitting and unification procedure. hybrid-tree similarly benefits from the introduction of

multi-level rules composed from smaller rules, a process similar to the one used for creating phrase tables in a phrase-based MT system.

## 6 Discussion

Our results validate the hypothesis that it is possible to adapt an ordinary MT system into a working semantic parser. In spite of the comparative simplicity of the approach, it achieves scores comparable to (and sometimes better than) many state-of-the-art systems. For this reason, we argue for the use of a machine translation baseline as a point of comparison for new methods. The results also demonstrate the usefulness of two techniques which are crucial for successful MT, but which are not widely used in semantic parsing. The first is the incorporation of a language model (or comparable long-distance structure-scoring model) to assign scores to predicted parses independent of the transformation model. The second is the use of large, composed rules (rather than rules which trigger on only one lexical item, or on tree portions of limited depth (Lu et al., 2008)) in order to “memorize” frequently-occurring large-scale structures.

## 7 Conclusions

We have presented a semantic parser which uses techniques from machine translation to learn mappings from natural language to variable-free meaning representations. The parser performs comparably to several recent purpose-built semantic parsers on the GeoQuery dataset, while training considerably faster than state-of-the-art systems. Our experiments demonstrate the usefulness of several techniques which might be broadly applied to other semantic parsers, and provides an informative basis for future work.

## Acknowledgments

Jacob Andreas is supported by a Churchill Scholarship. Andreas Vlachos is funded by the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270019 (SPACEBOOK project [www.spacebook-project.eu](http://www.spacebook-project.eu)).

## References

Steven Bird, Edward Loper, and Edward Klein. 2009. *Natural Language Processing with Python*. O’Reilly Media, Inc.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 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, Ann Arbor, Michigan.

H.B. Curry, J.R. Hindley, and J.P. Seldin. 1980. *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495, Portland, Oregon.

Bevan K. Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic parsing with bayesian tree transducers. In *Proceedings of the 50th Annual Meeting of the Association of Computational Linguistics*, pages 488–496, Jeju, Korea.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch-Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, Massachusetts.

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of*

*the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon.

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 1611–1622. Association for Computational Linguistics.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 783–792, Edinburgh, UK.

Andreas Maletti. 2010. Survey: Tree transducers in machine translation. In *Proceedings of the 2nd Workshop on Non-Classical Models for Automata and Applications*, Jena, Germany.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China.

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, Santa Monica, CA.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 439–446, New York.

John M. Zelle. 1995. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin.

# A relatedness benchmark to test the role of determiners in compositional distributional semantics

Raffaella Bernardi and Georgiana Dinu and Marco Marelli and Marco Baroni

Center for Mind/Brain Sciences (University of Trento, Italy)

first.last@unitn.it

## Abstract

Distributional models of semantics capture word meaning very effectively, and they have been recently extended to account for compositionally-obtained representations of phrases made of content words. We explore whether compositional distributional semantic models can also handle a construction in which grammatical terms play a crucial role, namely determiner phrases (DPs). We introduce a new publicly available dataset to test distributional representations of DPs, and we evaluate state-of-the-art models on this set.

## 1 Introduction

*Distributional semantics models* (DSMs) approximate meaning with vectors that record the distributional occurrence patterns of words in corpora. DSMs have been effectively applied to increasingly more sophisticated semantic tasks in linguistics, artificial intelligence and cognitive science, and they have been recently extended to capture the meaning of phrases and sentences via compositional mechanisms. However, scaling up to larger constituents poses the issue of how to handle *grammatical* words, such as determiners, prepositions, or auxiliaries, that lack rich conceptual content, and operate instead as the logical “glue” holding sentences together.

In typical DSMs, grammatical words are treated as “stop words” to be discarded, or at best used as context features in the representation of *content* words. Similarly, current *compositional* DSMs (cDSMs) focus almost entirely on phrases made of two or more content words (e.g., adjective-noun or verb-noun combinations) and completely ignore grammatical words, to the point that even the test set of transitive sentences proposed by Grefenstette and Sadrzadeh (2011) contains only

Tarzan-style statements with determiner-less subjects and objects: “*table show result*”, “*priest say mass*”, etc. As these examples suggest, however, as soon as we set our sight on modeling phrases and sentences, grammatical words are hard to avoid. Stripping off grammatical words has more serious consequences than making you sound like the Lord of the Jungle. Even if we accept the view of, e.g., Garrette et al. (2013), that the logical framework of language should be left to other devices than distributional semantics, and the latter should be limited to similarity scoring, still ignoring grammatical elements is going to dramatically distort the very similarity scores (c)DSMs should provide. If we want to use a cDSM for the classic similarity-based paraphrasing task, the model shouldn’t conclude that “*The table shows many results*” is identical to “*the table shows no results*” since the two sentences contain the same content words, or that “*to kill many rats*” and “*to kill few rats*” are equally good paraphrases of “*to exterminate rats*”.

We focus here on how cDSMs handle *determiners* and the phrases they form with nouns (*determiner phrases*, or DPs).<sup>1</sup> While determiners are only a subset of grammatical words, they are a large and important subset, constituting the natural stepping stone towards sentential distributional semantics: Compositional methods have already been successfully applied to simple noun-verb and noun-verb-noun structures (Mitchell and Lapata, 2008; Grefenstette and Sadrzadeh, 2011), and determiners are just what is missing to turn these skeletal constructions into full-fledged sentences. Moreover, determiner-noun phrases are, in superficial syntactic terms, similar to the adjective-noun phrases that have already been extensively studied from a cDSM perspective by Baroni and Zampar-

<sup>1</sup>Some linguists refer to what we call DPs as noun phrases or NPs. We say DPs simply to emphasize our focus on determiners.

elli (2010), Guevara (2010) and Mitchell and Lapata (2010). Thus, we can straightforwardly extend the methods already proposed for adjective-noun phrases to DPs.

We introduce a new task, a similarity-based challenge, where we consider nouns that are strongly conceptually related to certain DPs and test whether cDSMs can pick the most appropriate related DP (e.g., *monarchy* is more related to *one ruler* than *many rulers*).<sup>2</sup> We make our new dataset publicly available, and we hope that it will stimulate further work on the distributional semantics of grammatical elements.<sup>3</sup>

## 2 Composition models

Interest in compositional DSMs has skyrocketed in the last few years, particularly since the influential work of Mitchell and Lapata (2008; 2009; 2010), who proposed three simple but effective composition models. In these models, the composed vectors are obtained through component-wise operations on the constituent vectors. Given input vectors  $\mathbf{u}$  and  $\mathbf{v}$ , the multiplicative model (**mult**) returns a composed vector  $\mathbf{p}$  with:  $p_i = u_i v_i$ . In the weighted additive model (**wadd**), the composed vector is a weighted sum of the two input vectors:  $\mathbf{p} = \alpha \mathbf{u} + \beta \mathbf{v}$ , where  $\alpha$  and  $\beta$  are two scalars. Finally, in the **dilation** model, the output vector is obtained by first decomposing one of the input vectors, say  $\mathbf{v}$ , into a vector parallel to  $\mathbf{u}$  and an orthogonal vector. Following this, the parallel vector is dilated by a factor  $\lambda$  before re-combining. This results in:  $\mathbf{p} = (\lambda - 1)\langle \mathbf{u}, \mathbf{v} \rangle \mathbf{u} + \langle \mathbf{u}, \mathbf{u} \rangle \mathbf{v}$ .

A more general form of the additive model (**fulladd**) has been proposed by Guevara (2010) (see also Zanzotto et al. (2010)). In this approach, the two vectors to be added are pre-multiplied by weight matrices estimated from corpus-extracted examples:  $\mathbf{p} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{v}$ .

Baroni and Zamparelli (2010) and Coecke et al. (2010) take inspiration from formal semantics to characterize composition in terms of *function application*. The former model adjective-noun phrases by treating the adjective as a function from nouns onto modified nouns. Given that linear functions can be expressed by matrices and their application by matrix-by-vector multiplication, a

<sup>2</sup>Baroni et al. (2012), like us, study determiner phrases with distributional methods, but they do not model them compositionally.

<sup>3</sup>Dataset and code available from [clic.cimec.unitn.it/composes](http://clic.cimec.unitn.it/composes).

functor (such as the adjective) is represented by a matrix  $\mathbf{U}$  to be multiplied with the argument vector  $\mathbf{v}$  (e.g., the noun vector):  $\mathbf{p} = \mathbf{U}\mathbf{v}$ . Adjective matrices are estimated from corpus-extracted examples of noun vectors and corresponding output adjective-noun phrase vectors, similarly to Guevara’s approach.<sup>4</sup>

## 3 The noun-DP relatedness benchmark

Paraphrasing a single word with a phrase is a natural task for models of compositionality (Turney, 2012; Zanzotto et al., 2010) and determiners sometimes play a crucial role in defining the meaning of a noun. For example a *trilogy* is composed of *three works*, an *assemblage* includes *several things* and an *orchestra* is made of *many musicians*. These examples are particularly interesting, since they point to a “conceptual” use of determiners, as components of the stable and generic meaning of a content word (as opposed to situation-dependent deictic and anaphoric usages): for these determiners the boundary between content and grammatical word is somewhat blurred, and they thus provide a good entry point for testing DSM representations of DPs on a classic similarity task. In other words, we can set up an experiment in which having an effective representation of the determiner is crucial in order to obtain the correct result.

Using regular expressions over WordNet glosses (Fellbaum, 1998) and complementing them with definitions from various online dictionaries, we constructed a list of more than 200 nouns that are strongly conceptually related to a specific DP. We created a multiple-choice test set by matching each noun with its associated DP (*target DP*), two “foil” DPs sharing the same noun as the target but combined with other determiners (*same-N foils*), one DP made of the target determiner combined with a random noun (*same-D foil*), the target determiner (*D foil*), and the target noun (*N foil*). A few examples are shown in Table 1. After the materials were checked by all authors, two native speakers took the multiple-choice test. We removed the cases (32) where these subjects provided an unexpected answer. The final set,

<sup>4</sup>Other approaches to composition in DSMs have been recently proposed by Socher et al. (2012) and Turney (2012). We leave their empirical evaluation on DPs to further work, in the first case because it is not trivial to adapt their complex architecture to our setting; in the other because it is not clear how Turney would extend his approach to represent DPs.

| <i>noun</i> | <i>target DP</i> | <i>same-N foil 1</i> | <i>same-N foil 2</i> | <i>same-D foil</i>  | <i>D foil</i> | <i>N foil</i> |
|-------------|------------------|----------------------|----------------------|---------------------|---------------|---------------|
| duel        | two opponents    | various opponents    | three opponents      | two engineers       | two           | opponents     |
| homeless    | no home          | too few homes        | one home             | no incision         | no            | home          |
| polygamy    | several wives    | most wives           | fewer wives          | several negotiators | several       | wives         |
| opulence    | too many goods   | some goods           | no goods             | too many abductions | too many      | goods         |

Table 1: Examples from the noun-DP relatedness benchmark

characterized by full subject agreement, contains 173 nouns, each matched with 6 possible answers. The target DPs contain 23 distinct determiners.

## 4 Setup

Our semantic space provides distributional representations of determiners, nouns and DPs. We considered a set of 50 determiners that include all those in our benchmark and range from quantifying determiners (*every, some...*) and low numerals (*one to four*), to multi-word units analyzed as single determiners in the literature, such as *a few, all that, too much*. We picked the 20K most frequent nouns in our source corpus considering singular and plural forms as separate words, since number clearly plays an important role in DP semantics. Finally, for each of the target determiners we added to the space the 2K most frequent DPs containing that determiner and a target noun.

Co-occurrence statistics were collected from the concatenation of ukWaC, a mid-2009 dump of the English Wikipedia and the British National Corpus,<sup>5</sup> with a total of 2.8 billion tokens. We use a bag-of-words approach, counting co-occurrence with all context words in the same sentence with a target item. We tuned a number of parameters on the independent MEN word-relatedness benchmark (Bruni et al., 2012). This led us to pick the top 20K most frequent content word lemmas as context items, Pointwise Mutual Information as weighting scheme, and dimensionality reduction by Non-negative Matrix Factorization.

Except for the parameter-free *mult* method, parameters of the composition methods are estimated by minimizing the average Euclidean distance between the model-generated and corpus-extracted vectors of the 20K DPs we consider.<sup>6</sup> For the *lexfunc* model, we assume that the determiner is the functor and the noun is the argument,

<sup>5</sup>wacky.sslmit.unibo.it; www.natcorp.ox.ac.uk

<sup>6</sup>All vectors are normalized to unit length before composition. Note that the objective function used in estimation minimizes the distance between model-generated and corpus-extracted vectors. We do *not* use labeled evaluation data to optimize the model parameters.

| <i>method</i> | <i>accuracy</i> | <i>method</i> | <i>accuracy</i> |
|---------------|-----------------|---------------|-----------------|
| lexfunc       | 39.3            | noun          | 17.3            |
| fulladd       | 34.7            | random        | 16.7            |
| observed      | 34.1            | mult          | 12.7            |
| dilation      | 31.8            | determiner    | 4.6             |
| wadd          | 23.1            |               |                 |

Table 2: Percentage accuracy of composition methods on the relatedness benchmark

and estimate separate matrices representing each determiner using the 2K DPs in the semantic space that contain that determiner. For *dilation*, we treat direction of stretching as a parameter, finding that it is better to stretch the noun.

Similarly to the classic TOEFL synonym detection challenge (Landauer and Dumais, 1997), our models tackle the relatedness task by measuring cosines between each target noun and the candidate answers and returning the item with the highest cosine.

## 5 Results

Table 2 reports the accuracy results (mean ranks of correct answers confirm the same trend). All models except *mult* and *determiner* outperform the trivial *random* guessing baseline, although they are all well below the 100% accuracy of the humans who took our test. For the *mult* method we observe a very strong bias for choosing a single word as answer (>60% of the times), which in the test set is always incorrect. This leads to its accuracy being below the chance level. We suspect that the highly “intersective” nature of this model (we obtain very sparse composed DP vectors, only  $\approx 4\%$  dense) leads to it not being a reliable method for comparing sequences of words of different length: Shorter sequences will be considered more similar due to their higher density. The *determiner*-only baseline (using the vector of the component determiner as surrogate for the DP) fails because D vectors tend to be far from N vectors, thus the N foil is often preferred to the correct response (that is represented, for this baseline, by its D). In the *noun*-only baseline (use the vector of the component noun as surrogate for the DP),

the correct response is identical to the same-N and N foils, thus forcing a random choice between these. Not surprisingly, this approach performs quite badly. The *observed* DP vectors extracted directly from the corpus compete with the top compositional methods, but do not surpass them.<sup>7</sup>

The *lexfunc* method is the best compositional model, indicating that its added flexibility in modeling composition pays off empirically. The *fulladd* model is not as good, but also performs well. The *wadd* and especially *dilation* models perform relatively well, but they are penalized by the fact that they assign more weight to the noun vectors, making the right answer dangerously similar to the same-N and N foils.

Taking a closer look at the performance of the best model (*lexfunc*), we observe that it is not equally distributed across determiners. Focusing on those determiners appearing in at least 4 correct answers, they range from those where *lexfunc* performance was very significantly above chance ( $p < 0.001$  of equal or higher chance performance): *too few*, *all*, *four*, *too much*, *less*, *several*; to those on which performance was still significant but less impressively so ( $0.001 < p < 0.05$ ): *several*, *no*, *various*, *most*, *two*, *too many*, *many*, *one*; to those where performance was not significantly better than chance at the 0.05 level: *much*, *more*, *three*, *another*. Given that, on the one hand, performance is not constant across determiners, and on the other no obvious groupings can account for their performance difference (compare the excellent *lexfunc* performance on *four* to the lousy one on *three*!), future research should explore the contextual properties of specific determiners that make them more or less amenable to be captured by compositional DSMs.

## 6 Conclusion

DSMs, even when applied to phrases, are typically seen as models of content word meaning. However, to scale up compositionally beyond the simplest constructions, cDSMs must deal with grammatical terms such as determiners. This paper started exploring this issue by introducing a new and publicly available set testing DP semantics in a similarity-based task and using it to systematically evaluate, for the first time, cDSMs on a con-

<sup>7</sup>The *observed* method is in fact at advantage in our experiment because a considerable number of DP foils are not found in the corpus and are assigned similarity 0 with the target.

struction involving grammatical words. The most important take-home message is that distributional representations are rich enough to encode information about determiners, achieving performance well above chance on the new benchmark.

Theoretical considerations would lead one to expect a “functional” approach to determiner representations along the lines of Baroni and Zamparelli (2010) and Coecke et al. (2010) to outperform those approaches that combine vectors separately representing determiners and nouns. This prediction was largely borne out in the results, although the additive models, and particularly *fulladd*, were competitive rivals.

We attempted to capture the distributional semantics of DPs using a fairly standard, “vanilla” semantic space characterized by latent dimensions that summarize patterns of co-occurrence with content word contexts. By inspecting the context words that are most associated with the various latent dimensions we obtained through Non-negative Matrix Factorization, we notice how they are capturing broad, “topical” aspects of meaning (the first dimension is represented by *scripture*, *believer*, *resurrection*, the fourth by *fever*, *infection*, *infected*, and so on). Considering the sort of semantic space we used (which we took to be a reasonable starting point because of its effectiveness in a standard lexical task), it is actually surprising that we obtained the significant results we obtained. Thus, a top priority in future work is to explore different contextual features, such as adverbs and grammatical terms, that might carry information that is more directly relevant to the semantics of determiners.

Another important line of research pertains to improving composition methods: Although the best model, at 40% accuracy, is well above chance, we are still far from the 100% performance of humans. We will try, in particular, to include non-linear transformations in the spirit of Socher et al. (2012), and look for better ways to automatically select training data.

Last but not least, in the near future we would like to test if cDSMs, besides dealing with similarity-based aspects of determiner meaning, can also help in capturing those formal properties of determiners, such as monotonicity or definiteness, that theoretical semanticists have been traditionally interested in.



## 7 Acknowledgments

This research was supported by the ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES).

## References

- Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP*, pages 1183–1193, Boston, MA.
- Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-Chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of EACL*, pages 23–32, Avignon, France.
- Elia Bruni, Gemma Boleda, Marco Baroni, and Nam Khanh Tran. 2012. Distributional semantics in Technicolor. In *Proceedings of ACL*, pages 136–145, Jeju Island, Korea.
- Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Dan Garrette, Katrin Erk, and Ray Mooney. 2013. A formal approach to linking logical form and vector-space lexical semantics. In H. Bunt, J. Bos, and S. Pulman, editors, *Computing Meaning, Vol. 4*. In press.
- Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of EMNLP*, pages 1394–1404, Edinburgh, UK.
- Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of GEMS*, pages 33–37, Uppsala, Sweden.
- Thomas Landauer and Susan Dumais. 1997. A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, pages 236–244, Columbus, OH.
- Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of EMNLP*, pages 430–439, Singapore.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.
- Richard Socher, Brody Huval, Christopher Manning, and Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211, Jeju Island, Korea.
- Peter Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.
- Fabio Zanzotto, Ioannis Korkontzelos, Francesca Falucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of COLING*, pages 1263–1271, Beijing, China.

# An Empirical Study on Uncertainty Identification in Social Media Context

Zhongyu Wei<sup>1</sup>, Junwen Chen<sup>1</sup>, Wei Gao<sup>2</sup>,  
Binyang Li<sup>1</sup>, Lanjun Zhou<sup>1</sup>, Yulan He<sup>3</sup>, Kam-Fai Wong<sup>1</sup>

<sup>1</sup>The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

<sup>2</sup>Qatar Computing Research Institute, Qatar Foundation, Doha, Qatar

<sup>3</sup>School of Engineering & Applied Science, Aston University, Birmingham, UK

{zywei, jwchen, byli, ljzhou, kfwong}@se.cuhk.edu.hk

wgao@qf.org.qa, y.he@cantab.net

## Abstract

Uncertainty text detection is important to many social-media-based applications since more and more users utilize social media platforms (e.g., Twitter, Facebook, etc.) as information source to produce or derive interpretations based on them. However, existing uncertainty cues are ineffective in social media context because of its specific characteristics. In this paper, we propose a variant of annotation scheme for uncertainty identification and construct the first uncertainty corpus based on tweets. We then conduct experiments on the generated tweets corpus to study the effectiveness of different types of features for uncertainty text identification.

## 1 Introduction

Social media is not only a social network tool for people to communicate but also plays an important role as information source with more and more users searching and browsing news on it. People also utilize information from social media for developing various applications, such as earthquake warning systems (Sakaki et al., 2010) and fresh webpage discovery (Dong et al., 2010). However, due to its casual and word-of-mouth peculiarities, the quality of information in social media in terms of factuality becomes a premier concern. Chances are there for uncertain information or even rumors flooding in such a context of free form. We analyzed a tweet dataset which includes 326,747 posts (Details are given in Section 3) collected during 2011 London Riots, and result reveals that at least 18.91% of these tweets bear uncertainty characteristics<sup>1</sup>. Therefore, distinguishing uncertain statements from factual ones is crucial for users to synthesize social media information to produce or derive reliable interpretations,

<sup>1</sup>The preliminary study was done based on a manually defined uncertainty cue-phrase list. Tweets containing at least one hedge cue were treated as uncertain.

and this is expected helpful for applications like credibility analysis (Castillo et al., 2011) and rumor detection (Qazvinian et al., 2011) based on social media.

Although uncertainty has been studied theoretically for a long time as a grammatical phenomena (Seifert and Welte, 1987), the computational treatment of uncertainty is a newly emerging area of research. Szarvas et al. (2012) pointed out that “Uncertainty - in its most general sense - can be interpreted as lack of information: the receiver of the information (i.e., the hearer or the reader) cannot be certain about some pieces of information”. In recent years, the identification of uncertainty in formal text, e.g., biomedical text, reviews or newswire, has attracted lots of attention (Kilicoglu and Bergler, 2008; Medlock and Briscoe, 2007; Szarvas, 2008; Light et al., 2004). However, uncertainty identification in social media context is rarely explored.

Previous research shows that uncertainty identification is domain dependent as the usage of hedge cues varies widely in different domains (Morante and Sporleder, 2012). Therefore, the employment of existing out-of-domain corpus to social media context is ineffective. Furthermore, compared to the existing uncertainty corpus, the expression of uncertainty in social media is fairly different from that in formal text in a sense that people usually raise questions or refer to external information when making uncertain statements. But, neither of the uncertainty expressions can be represented based on the existing types of uncertainty defined in the literature. Therefore, a different uncertainty classification scheme is needed in social media context.

In this paper, we propose a novel uncertainty classification scheme and construct the first uncertainty corpus based on social media data – tweets in specific here. And then we conduct experiments for uncertainty post identification and study the effectiveness of different categories of features based on the generated corpus.

## 2 Related work

We introduce some popular uncertainty corpora and methods for uncertainty identification.

### 2.1 Uncertainty corpus

Several text corpora from various domains have been annotated over the past few years at different levels (e.g., expression, event, relation, sentence) with information related to uncertainty.

Sauri and Pustejovsky (2009) presented a corpus annotated with information about the factuality of events, namely *Factbank*, which is constructed based on *TimeBank*<sup>2</sup> containing 3,123 annotated sentences from 208 news documents with 8 different levels of uncertainty defined.

Vincze et al. (2008) constructed the BioSocpe corpus, which consists of medical and biological texts annotated for negation, uncertainty and their linguistic scope. This corpus contains 20,924 sentences.

Ganter et al. (2009) generated Wikipedia Weasels Corpus, where *Weasel tags* in Wikipedia articles is adopted readily as labels for uncertainty annotation. It contains 168,923 unique sentences with 437 weasel tags in total.

Although several uncertainty corpora exist, there is not a uniform set of standard for uncertainty annotation. Szarvas et al. (2012) normalized the annotation of the three corpora aforementioned. However, the context of these corpora is different from that of social media. Typically, these documents annotated are grammatically correct, carefully punctuated, formally structured and logically expressed.

### 2.2 Uncertainty identification

Previous work on uncertainty identification focused on classifying sentences into uncertain or definite categories. Existing approaches are mainly based on supervised methods (Light et al., 2004; Medlock and Briscoe, 2007; Medlock, 2008; Szarvas, 2008) using the annotated corpus with different types of features including Part-Of-Speech (POS) tags, stems, n-grams, etc..

Classification of uncertain sentences was consolidated as a task in the 2010 edition of CoNLL shared task on learning to detect hedge cues and their scope in natural language text (Farkas et al., 2010). The best system for Wikipedia data (Georgescu, 2010) employed Support Vector Machine (SVM), and the best system for biological data (Tang et al., 2010) adopted Conditional

<sup>2</sup><http://www.timeml.org/site/timebank/timebank.html>

Random Fields (CRF).

In our work, we conduct an empirical study of uncertainty identification on tweets dataset and explore the effectiveness of different types of features (i.e., content-based, user-based and Twitter-specific) from social media context.

## 3 Uncertainty corpus for microblogs

### 3.1 Types of uncertainty in microblogs

Traditionally, uncertainty can be divided into two categories, namely *Epistemic* and *Hypothetical* (Kiefer, 2005). For *Epistemic*, there are two sub-classes *Possible* and *Probable*. For *Hypothetical*, there are four sub-classes including *Investigation*, *Condition*, *Doxastic* and *Dynamic*. The detail of the classification is described as below (Kiefer, 2005):

**Epistemic:** On the basis of our world knowledge we cannot decide at the moment whether the statement is true or false.

**Hypothetical:** This type of uncertainty includes four sub-classes:

- **Doxastic:** Expresses the speaker’s beliefs and hypotheses.
- **Investigation:** Proposition under investigation.
- **Condition:** Proposition under condition.
- **Dynamic:** Contains deontic, dispositional, circumstantial and buletic modality.

Compared to the existing uncertainty corpora, social media authors enjoy free form of writing. In order to study the difference, we annotated a small set of 827 randomly sampled tweets according to the scheme of uncertainty types above, in which we found 65 uncertain tweets. And then, we manually identified all the possible uncertain tweets, and found 246 really uncertain ones out of these 827 tweets, which means that 181 uncertain tweets are missing based on this scheme. We have the following three salient observations:

– Firstly, there is no tweet found with the type of *Investigation*. We find people seldom use words like “examine” or “test” (indicative words of *Investigation* category) when posting tweets. Once they do this, the statement should be considered as highly certain. For example, @dobibid *I have tested the link, it is fake!*

– Secondly, people frequently raise questions about some specific topics for confirmation which expresses uncertainty. For example, @ITVCentral

Can you confirm that Birmingham children’s hospital has/hasn’t been attacked by rioters?

– Thirdly, people tend to post message with external information (e.g., story from friends) which reveals uncertainty. For example, *Friend who works at the children’s hospital in Birmingham says the riot police are protecting it.*

Based on these observations, we propose a variant of uncertainty types in social media context by eliminating the category of *Investigation* and adding the category of *Question* and *External* under *Hypothetical*, as shown in Table 3.1. Note that our proposed scheme is based on Kiefer’s work (2005) which was previously extended to normalize uncertainty corpora in different genres by Szarvas et al. (2012). But we did not try these extended schema for specific genres since even the most general one (Kiefer, 2005) was proved unsuitable for social media context.

### 3.2 Annotation result

The dataset we annotated was collected from Twitter using Streaming API during summer riots in London during August 6-13 2011, including 326,747 tweets in total. Search criteria include hashtags like #ukriots, #londonriots, #prayforlondon, and so on. We further extracted the tweets relating to seven significant events during the riot identified by UK newspaper The Guardian from this set of tweets. We annotated all the 4,743 extracted tweets for the seven events<sup>3</sup>.

Two annotators were trained to annotate the dataset independently. Given a collection of tweets  $T = \{t_1, t_2, t_3 \dots t_n\}$ , the annotation task is to label each tweet  $t_i$  as either uncertain or certain. Uncertainty assertions are to be identified in terms of the judgements about the author’s intended meaning rather than the presence of uncertain cue-phrase. For those tweets annotated as uncertain, sub-class labels are also required according to the classification indicated in Table 3.1 (i.e., multi-label is allowed).

The Kappa coefficient (Carletta, 1996) indicating inter-annotator agreement was 0.9073 for the certain/uncertain binary classification and was 0.8271 for fine-grained annotation. The conflict labels from the two annotators were resolved by a third annotator. Annotation result is displayed in Table 3.2, where 926 out of 4,743 tweets are labeled as uncertain accounting for 19.52%. *Question* is the uncertainty category with most tweets, followed by *External*. Only 21 tweets are labeled

<sup>3</sup><http://www.guardian.co.uk/uk/interactive/2011/dec/07/london-riots-twitter>

|              |            |     |
|--------------|------------|-----|
| Tweet#       | 4743       |     |
| Uncertainty# | 926        |     |
| Epistemic    | Possible#  | 16  |
|              | Probable#  | 129 |
| Hypothetical | Condition# | 71  |
|              | Doxastic#  | 48  |
|              | Dynamic#   | 21  |
|              | External#  | 208 |
|              | Question#  | 488 |

Table 2: Statistics of annotation result

as *Dynamic* and all of them are buletic modality<sup>4</sup> which shares similarity with *Doxastic*. Therefore, we consider *Dynamic* together with *Domestic* in the error analysis for simplicity. During the preliminary annotation, we found that uncertainty cue-phrase is a good indicator for uncertainty tweets since tweets labeled as uncertain always contain at least one cue-phrase. Therefore, annotators are also required identify cue-phrases which trigger the sense of uncertainty in the tweet. All cue-phrases appearing more than twice are collected to form a uncertainty cue-phrase list.

## 4 Experiment and evaluation

We aim to identify those uncertainty tweets from tweet collection automatically based on machine learning approaches. In addition to n-gram features, we also explore the effectiveness of three categories of social media specific features including content-based, user-based and Twitter-specific ones. The description of the three categories of features is shown in Table 4. Since the length of tweet is relatively short, we therefore did not carry out stopwords removal or stemming.

Our preliminary experiments showed that combining unigrams with bigrams and trigrams gave better performance than using any one or two of these three features. Therefore, we just report the result based on the combination of them as n-gram features. Five-fold cross validation is used for evaluation. Precision, recall and F-1 score of uncertainty category are used as the metrics.

### 4.1 Overall performance

The overall performance of different approaches is shown in Table 4.1. We used uncertainty cue-phrase matching approach as baseline, denoted by *CP*. For *CP*, we labeled tweets containing at least one entry in uncertainty cue-phrase list (described in Section 3) as uncertain. All the other approaches are supervised methods using *SVM* based on different feature sets. *n-gram* stands for n-gram feature set, *C* means content-based feature set, *U* denotes user-based feature set, *T* represents

<sup>4</sup>Proposition expresses plans, intentions or desires.

| Category     | Subtype        | Cue Phrase         | Example   |
|--------------|----------------|--------------------|---|
| Epistemic    | Possible, etc. | may, etc.          | It may be raining.  |
|              | Probable       | likely, etc.       | It is probably raining.   |
| Hypothetical | Condition      | if, etc.           | If it rains, we'll stay in.                                       |
|              | Doxastic       | believe, etc.      | He believes that the Earth is flat.                               |
|              | Dynamic        | hope, etc.         | fake picture of the london eye on fire... i hope                  |
|              | External       | someone said, etc. | Someone said that London zoo was attacked.                        |
|              | Question       | seriously?, etc.   | Birmingham riots are moving to the children hospital?! seriously? |

Table 1: Classification of uncertainty in social media context

| Category         | Name            | Description                                  |
|------------------|-----------------|--|
| Content-based    | Length          | Length of the tweet                          |
|                  | Cue_Phrase      | Whether the tweet contains a uncertainty cue |
|                  | OOV_Ratio       | Ratio of words out of vocabulary             |
| Twitter-specific | URL             | Whether the tweet contains a URL             |
|                  | URL_Count       | Frequency of URLs in corpus                  |
|                  | Retweet_Count   | How many times has this tweet been retweeted |
|                  | Hashtag         | Whether the tweet contains a hashtag         |
|                  | Hashtag_Count   | Number of Hashtag in tweets                  |
|                  | Reply           | Is the current tweet a reply tweet           |
| User-based       | Rtweet          | Is the current tweet a retweet tweet         |
|                  | Follower_Count  | Number of follower the user owns             |
|                  | List_Count      | Number of list the users owns                |
|                  | Friend_Count    | Number of friends the user owns              |
|                  | Favorites_Count | Number of favorites the user owns            |
|                  | Tweet_Count     | Number of tweets the user published          |
|                  | Verified        | Whether the user is verified                 |

Table 3: Feature list for uncertainty classification

| Approach                         | Precision     | Recall        | F-1           | Type   | Poss. | Prob. | D.&D. | Cond. | Que. | Ext. |
|----------------------------------|---------------|---------------|---------------|--------|-------|-------|-------|-------|------|------|
| CP                               | 0.3732        | <b>0.9589</b> | 0.5373        | Total# | 16    | 129   | 69    | 71    | 488  | 208  |
| SVM <sub>n-gram</sub>            | 0.7278        | 0.8259        | 0.7737        | Error# | 11    | 20    | 18    | 11    | 84   | 40   |
| SVM <sub>n-gram+C</sub>          | 0.8010        | 0.8260        | 0.8133        | %      | 0.69  | 0.16  | 0.26  | 0.15  | 0.17 | 0.23 |
| SVM <sub>n-gram+U</sub>          | 0.7708        | 0.8271        | 0.7979        |        |       |       |       |       |      |      |
| SVM <sub>n-gram+T</sub>          | 0.7578        | 0.8266        | 0.7907        |        |       |       |       |       |      |      |
| SVM <sub>n-gram+ALL</sub>        | <b>0.8162</b> | 0.8269        | <b>0.8215</b> |        |       |       |       |       |      |      |
| SVM <sub>n-gram+Cue_Phrase</sub> | 0.7989        | 0.8266        | 0.8125        |        |       |       |       |       |      |      |
| SVM <sub>n-gram+Length</sub>     | 0.7372        | 0.8216        | 0.7715        |        |       |       |       |       |      |      |
| SVM <sub>n-gram+OOV_Ratio</sub>  | 0.7414        | 0.8233        | 0.7802        |        |       |       |       |       |      |      |

Table 4: Result of uncertainty tweets identification

Twitter-specific feature set and *ALL* is the combination of *C*, *U* and *T*.

Table 4.1 shows that *CP* achieves the best recall but its precision is the lowest. The learning based methods with different feature sets give some similar recalls. Compared to *CP*, *SVM<sub>n-gram</sub>* increases the F-1 score by 43.9% due to the salient improvement on precision and small drop of recall. The performance improves in terms of precision and F-1 score when the feature set is expanded by adding *C*, *U* or *T* onto *n-gram*, where *+C* brings the highest gain, and *SVM<sub>n-gram+ALL</sub>* performs best in terms of precision and F-1 score. We then study the effectiveness of the three content-based features, and result shows that the presence of uncertain cue-phrase is most indicative for uncertainty tweet identification.

## 4.2 Error analysis

We analyze the prediction errors based on *SVM<sub>n-gram+ALL</sub>*. The distribution of errors in terms of different types of uncertainty is shown

Table 5: Error distributions

in Table 4.2. Our method performs worst on the type of *Possible* and on the combination of *Dynamic* and *Doxastic* because these two types have the least number of samples in the corpus and the classifier tends to be undertrained without enough samples.

## 5 Conclusion and future work

In this paper, we propose a variant of classification scheme for uncertainty identification in social media and construct the first uncertainty corpus based on tweets. We perform uncertainty identification experiments on the generated dataset to explore the effectiveness of different types of features. Result shows that the three categories of social media specific features can improve uncertainty identification. Furthermore, content-based features bring the highest improvement among the three and the presence of uncertain cue-phrase contributes most for content-based features.

In future, we will explore to use uncertainty identification for social media applications.

## 6 Acknowledgement

This work is partially supported by General Research Fund of Hong Kong (No. 417112).

## References

- Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, 22(2):249–254.
- Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 675–684.
- Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. 2010. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 331–340. ACM.
- 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 14th Conference on Computational Natural Language Learning—Shared Task*, pages 1–12. Association for Computational Linguistics.
- Viola Ganter and Michael Strube. 2009. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009*, pages 173–176. Association for Computational Linguistics.
- Maria Georgescu. 2010. A hedgehop over a max-margin framework using hedge cues. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task*, pages 26–31. Association for Computational Linguistics.
- Ferenc Kiefer. 2005. *Lehetoseg es szükségesség[Possibility and necessity]*. Tinta Kiado, Budapest.
- H. Kilicoglu and S. Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC bioinformatics*, 9(Suppl 11):S10.
- Marc Light, Xin Ying Qiu, and Padmini Srinivasan. 2004. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of BioLink 2004 workshop on linking biological literature, ontologies and databases: tools for users*, pages 17–24.
- B. Medlock and T. Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 992–999.
- Ben Medlock. 2008. Exploring hedge identification in biomedical literature. *Journal of Biomedical Informatics*, 41(4):636–654.
- Roser Morante and Caroline Sporleder. 2012. Modality and negation: An introduction to the special issue. *Computational Linguistics*, 38(2):223–260.
- Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599.
- 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*, pages 851–860. ACM.
- R. Saurí and J. Pustejovsky. 2009. Factbank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268.
- Stephan Seifert and Werner Welte. 1987. *A basic bibliography on negation in natural language*, volume 313. Gunter Narr Verlag.
- György Szarvas, Veronika Vincze, Richárd Farkas, György Móra, and Iryna Gurevych. 2012. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367.
- György Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics*.
- Buzhou Tang, Xiaolong Wang, Xuan Wang, Bo Yuan, and Shixi Fan. 2010. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task*, pages 13–17. Association for Computational Linguistics.
- V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. 2008. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(Suppl 11):S9.

# PARMA: A Predicate Argument Aligner

Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews,  
Charley Beller, Chris Callison-Burch, Jay DeYoung, Justin Snyder,  
Jonathan Weese, Tan Xu<sup>†</sup>, and Xuchen Yao

Human Language Technology Center of Excellence  
Johns Hopkins University, Baltimore, Maryland USA

<sup>†</sup>University of Maryland, College Park, Maryland USA

## Abstract

We introduce PARMA, a system for cross-document, semantic predicate and argument alignment. Our system combines a number of linguistic resources familiar to researchers in areas such as recognizing textual entailment and question answering, integrating them into a simple discriminative model. PARMA achieves state of the art results on an existing and a new dataset. We suggest that previous efforts have focussed on data that is biased and too easy, and we provide a more difficult dataset based on translation data with a low baseline which we beat by 17% F1.

## 1 Introduction

A key step of the information extraction pipeline is entity disambiguation, in which discovered entities across many sentences and documents must be organized to represent real world entities. The NLP community has a long history of entity disambiguation both within and across documents. While most information extraction work focuses on entities and noun phrases, there have been a few attempts at predicate, or event, disambiguation. Commonly a situational predicate is taken to correspond to either an event or a state, lexically realized in verbs such as “elect” or nominalizations such as “election”. Similar to entity coreference resolution, almost all of this work assumes unanchored mentions: predicate argument tuples are grouped together based on coreferent events. The first work on event coreference dates back to Bagga and Baldwin (1999). More recently, this task has been considered by Bejan and Harabagiu (2010) and Lee et al. (2012). As with unanchored entity disambiguation, these methods rely on clustering methods and evaluation metrics.

Another view of predicate disambiguation seeks

to link or align predicate argument tuples to an existing anchored resource containing references to events or actions, similar to anchored entity disambiguation (entity linking) (Dredze et al., 2010; Han and Sun, 2011). The most relevant, and perhaps only, work in this area is that of Roth and Frank (2012) who linked predicates across document pairs, measuring the F1 of aligned pairs.

Here we present PARMA, a new system for predicate argument alignment. As opposed to Roth and Frank, PARMA is designed as a trainable platform for the incorporation of the sort of lexical semantic resources used in the related areas of Recognizing Textual Entailment (RTE) and Question Answering (QA). We demonstrate the effectiveness of this approach by achieving state of the art performance on the data of Roth and Frank despite having little relevant training data. We then show that while the “lemma match” heuristic provides a strong baseline on this data, this appears to be an artifact of their data creation process (which was heavily reliant on word overlap). In response, we evaluate on a new and more challenging dataset for predicate argument alignment derived from multiple translation data. We release PARMA as a new framework for the incorporation and evaluation of new resources for predicate argument alignment.<sup>1</sup>

## 2 PARMA

PARMA (Predicate ARguMent Aligner) is a pipelined system with a wide variety of features used to align predicates and arguments in two documents. Predicates are represented as mention spans and arguments are represented as coreference chains (sets of mention spans) provided by in-document coreference resolution systems such as included in the Stanford NLP toolkit. Results indicated that the chains are of sufficient quality so as not to limit performance, though future work

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

## RF

- Australian [police]<sub>1</sub> have [arrested]<sub>2</sub> a man in the western city of Perth over an alleged [plot]<sub>3</sub> to [bomb]<sub>4</sub> Israeli diplomatic [buildings]<sub>5</sub> in the country , police and the suspect s [lawyer]<sub>6</sub> [said]<sub>7</sub>
- Federal [police]<sub>1</sub> have [arrested]<sub>2</sub> a man over an [alleged]<sub>5</sub> [plan]<sub>3</sub> to [bomb]<sub>4</sub> Israeli diplomatic [posts]<sub>8</sub> in Australia , the suspect s [attorney]<sub>6</sub> [said]<sub>7</sub> Tuesday

## LDC MTC

- As I [walked]<sub>1</sub> to the [veranda]<sub>2</sub> side , I [saw]<sub>2</sub> that a [tent]<sub>3</sub> is being decorated for [Mahfil-e-Naat]<sub>4</sub> -LRB- A [get-together]<sub>5</sub> in which the poetic lines in praise of Prophet Mohammad are recited -RRB-
- I [came]<sub>1</sub> towards the [balcony]<sub>2</sub> , and while walking over there I [saw]<sub>2</sub> that a [camp]<sub>3</sub> was set up outside for the [Naatia]<sub>4</sub> [meeting]<sub>5</sub> .

Figure 1: Example of gold-standard alignment pairs from Roth and Frank’s data set and our data set created from the LDC’s Multiple Translation Corpora. The RF data set exhibits high lexical overlap, where most of the alignments are between identical words like *police-police* and *said-said*. The LDC MTC was constructed to increase lexical diversity, leading to more challenging alignments like *veranda-balcony* and *tent-camp*

may relax this assumption.

We refer to a predicate or an argument as an “item” with type *predicate* or *argument*. An alignment between two documents is a subset of all pairs of items in either documents with the same type.<sup>2</sup> We call the two documents being aligned the source document  $S$  and the target document  $T$ . Items are referred to by their index, and  $a_{i,j}$  is a binary variable representing an alignment between item  $i$  in  $S$  and item  $j$  in  $T$ . A full alignment is an assignment  $\vec{a} = \{a_{ij} : i \in N_S, j \in N_T\}$ , where  $N_S$  and  $N_T$  are the set of item indices for  $S$  and  $T$  respectively.

We train a logistic regression model on example alignments and maximize the likelihood of a document alignment under the assumption that the item alignments are independent. Our objective is to maximize the log-likelihood of all  $p(S, T)$  with an L1 regularizer (with parameter  $\lambda$ ). After learning model parameters  $w$  by regularized maximum likelihood on training data, we introducing a threshold  $\tau$  on alignment probabilities to get a classifier. We perform line search on  $\tau$  and choose the value that maximizes F1 on dev data. Training was done using the Mallet toolkit (McCallum, 2002).

## 2.1 Features

The focus of PARMA is the integration of a diverse range of features based on existing lexical semantic resources. We built PARMA on a supervised framework to take advantage of this wide variety of features since they can describe many different correlated aspects of generation. The following features cover the spectrum from high-precision

<sup>2</sup>Note that type is not the same thing as part of speech: we allow nominal predicates like “death”.

to high-recall. Each feature has access to the proposed argument or predicate spans to be linked and the containing sentences as context. While we use supervised learning, some of the existing datasets for this task are very small. For extra training data, we pool material from different datasets and use the multi-domain split feature space approach to learn dataset specific behaviors (Daumé, 2007).

Features in general are defined over mention spans or head tokens, but we split these features to create separate feature-spaces for predicates and arguments.<sup>3</sup>

For argument coref chains we heuristically choose a canonical mention to represent each chain, and some features only look at this canonical mention. The canonical mention is chosen based on length,<sup>4</sup> information about the head word,<sup>5</sup> and position in the document.<sup>6</sup> In most cases, coref chains that are longer than one are proper nouns and the canonical mention is the first and longest mention (outranking pronominal references and other name shortenings).

**PPDB** We use lexical features from the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013). PPDB is a large set of paraphrases extracted from bilingual corpora using pivoting techniques. We make use of the English lexical portion which contains over 7 million rules for rewriting terms like “planet” and “earth”. PPDB offers a variety of conditional probabilities for each (synchronous context free grammar) rule, which we

<sup>3</sup>While conceptually cleaner, In practice we found this splitting to have no impact on performance.

<sup>4</sup>in tokens, not counting some words like determiners and auxiliary verbs

<sup>5</sup>like its part of speech tag and whether the it was tagged as a named entity

<sup>6</sup>mentions that appear earlier in the document and earlier in a given sentence are given preference



treat as independent experts. For each of these rule probabilities (experts), we find all rules that match the head tokens of a given alignment and have a feature for the max and harmonic mean of the log probabilities of the resulting rule set.

**FrameNet** FrameNet is a lexical database based on Charles Fillmore’s Frame Semantics (Fillmore, 1976; Baker et al., 1998). The database (and the theory) is organized around semantic frames that can be thought of as descriptions of events. Frames crucially include specification of the participants, or Frame Elements, in the event. The Destroying frame, for instance, includes frame elements `Destroyer` or `Cause Undergoer`. Frames are related to other frames through inheritance and perspectivization. For instance the frames `Commerce_buy` and `Commerce_sell` (with respective lexical realizations “buy” and “sell”) are both perspectives of `Commerce_goods-transfer` (no lexical realizations) which inherits from `Transfer` (with lexical realization “transfer”).

We compute a shortest path between headwords given edges (hypernym, hyponym, perspectivized parent and child) in FrameNet and bucket by distance to get features. We also have a binary feature for whether two tokens evoke the same frame.

**TED Alignments** Given two predicates or arguments in two sentences, we attempt to align the two sentences they appear in using a Tree Edit Distance (TED) model that aligns two dependency trees, based on the work described by (Yao et al., 2013). We represent a node in a dependency tree with three fields: lemma, POS tag and the type of dependency relation to the node’s parent. The TED model aligns one tree with the other using the dynamic programming algorithm of Zhang and Shasha (1989) with three predefined edits: deletion, insertion and substitution, seeking a solution yielding the minimum edit cost. Once we have built a tree alignment, we extract features for 1) whether the heads of the two phrases are aligned and 2) the count of how many tokens are aligned in both trees.

**WordNet** WordNet (Miller, 1995) is a database of information (synonyms, hypernyms, etc.) pertaining to words and short phrases. For each entry, WordNet provides a set of synonyms, hypernyms, etc. Given two spans, we use WordNet to determine semantic similarity by measuring how many synonym (or other) edges are needed to link two

terms. Similar words will have a short distance. For features, we find the shortest path linking the head words of two mentions using synonym, hypernym, hyponym, meronym, and holonym edges and bucket the length.

**String Transducer** To represent similarity between arguments that are names, we use a stochastic edit distance model. This stochastic string-to-string transducer has latent “edit” and “no edit” regions where the latent regions allow the model to assign high probability to contiguous regions of edits (or no edits), which are typical between variations of person names. In an edit region, parameters govern the relative probability of insertion, deletion, substitution, and copy operations. We use the transducer model of Andrews et al. (2012). Since in-domain name pairs were not available, we picked 10,000 entities at random from Wikipedia to estimate the transducer parameters. The entity labels were used as weak supervision during EM, as in Andrews et al. (2012).

For a pair of mention spans, we compute the conditional log-likelihood of the two mentions going both ways, take the max, and then bucket to get binary features. We duplicate these features with copies that only fire if both mentions are tagged as PER, ORG or LOC.

### 3 Evaluation

We consider three datasets for evaluating PARMA. For richer annotations that include lemmatizations, part of speech, NER, and in-doc coreference, we pre-processed each of the datasets using tools<sup>7</sup> similar to those used to create the Annotated Gigaword corpus (Napoles et al., 2012).

**Extended Event Coreference Bank** Based on the dataset of Bejan and Harabagiu (2010), Lee et al. (2012) introduced the Extended Event Coreference Bank (EECB) to evaluate cross-document event coreference. EECB provides document clusters, within which entities and events may corefer. Our task is different from Lee et al. but we can modify the corpus setup to support our task. To produce source and target document pairs, we select the first document within every cluster as the source and each of the remaining documents as target documents (i.e.  $N - 1$  pairs for a cluster of size  $N$ ). This yielded 437 document pairs.

**Roth and Frank** The only existing dataset for our task is from Roth and Frank (2012) (RF), who

<sup>7</sup><https://github.com/cnap/anno-pipeline>

annotated documents from the English Gigaword Fifth Edition corpus (Parker et al., 2011). The data was generated by clustering similar news stories from Gigaword using TF-IDF cosine similarity of their headlines. This corpus is small, containing only 10 document pairs in the development set and 60 in the test set. To increase the training size, we train PARMA with 150 randomly selected document pairs from both EECB and MTC, and the entire dev set from Roth and Frank using multi-domain feature splitting. We tuned the threshold  $\tau$  on the Roth and Frank dev set, but choose the regularizer  $\lambda$  based on a grid search on a 5-fold version of the EECB dataset.

**Multiple Translation Corpora** We constructed a new predicate argument alignment dataset based on the LDC Multiple Translation Corpora (MTC),<sup>8</sup> which consist of multiple English translations for foreign news articles. Since these multiple translations are semantically equivalent, they provide a good resource for aligned predicate argument pairs. However, finding good pairs is a challenge: we want pairs with significant overlap so that they have predicates and arguments that align, but not documents that are trivial rewrites of each other. Roth and Frank selected document pairs based on clustering, meaning that the pairs had high lexical overlap, often resulting in minimal rewrites of each other. As a result, despite ignoring all context, their baseline method (lemma-alignment) worked quite well.

To create a more challenging dataset, we selected document pairs from the multiple translations that minimize the lexical overlap (in English). Because these are translations, we know that there are equivalent predicates and arguments in each pair, and that any lexical variation preserves meaning. Therefore, we can select pairs with minimal lexical overlap in order to create a system that truly stresses lexically-based alignment systems.

Each document pair has a correspondence between sentences, and we run GIZA++ on these sentences to produce token-level alignments. We take all aligned nouns as arguments and all aligned verbs (excluding be-verbs, light verbs, and reporting verbs) as predicates. We then add negative examples by randomly substituting half of the sentences in one document with sentences from an

<sup>8</sup>LDC2010T10, LDC2010T11, LDC2010T12, LDC2010T14, LDC2010T17, LDC2010T23, LDC2002T01, LDC2003T18, and LDC2005T05

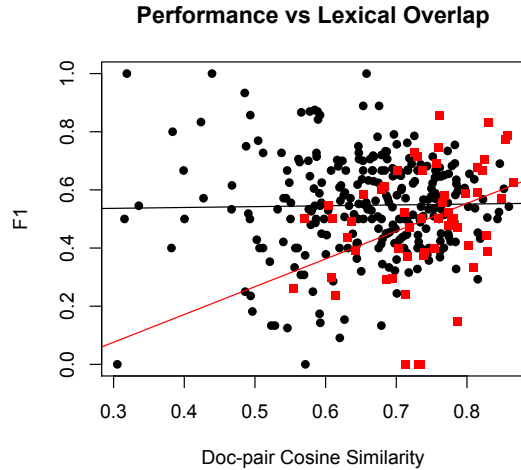


Figure 2: We plotted the PARMA’s performance on each of the document pairs. Red squares show the F1 for individual document pairs drawn from Roth and Frank’s data set, and black circles show F1 for our Multiple Translation Corpora test set. The x-axis represents the cosine similarity between the document pairs. On the RF data set, performance is correlated with lexical similarity. On our more lexically diverse set, this is not the case. This could be due to the fact that some of the documents in the RF sets are minor re-writes of the same newswire story, making them easy to align.

other corpus, guaranteed to be unrelated. The amount of substitutions we perform can vary the “relatedness” of the two documents in terms of the predicates and arguments that they talk about. This reflects our expectation of real world data, where we do not expect perfect overlap in predicates and arguments between a source and target document, as you would in translation data.

Lastly, we prune any document pairs that have more than 80 predicates or arguments or have a Jaccard index on bags of lemmas greater than 0.5, to give us a dataset of 328 document pairs.

**Metric** We use precision, recall, and F1. For the RF dataset, we follow Roth and Frank (2012) and Cohn et al. (2008) and evaluate on a version of F1 that considers SURE and POSSIBLE links, which are available in the RF data. Given an alignment to be scored  $A$  and a reference alignment  $B$  which contains SURE and POSSIBLE links,  $B_s$  and  $B_p$  respectively, precision and recall are:

$$P = \frac{|A \cap B_p|}{|A|} \quad R = \frac{|A \cap B_s|}{|B_s|} \quad (1)$$

|      |                | <b>F1</b>   | <b>P</b> | <b>R</b> |
|------|----------------|-------------|----------|----------|
| EECB | lemma          | 63.5        | 84.8     | 50.8     |
|      | PARMA          | <b>74.3</b> | 80.5     | 69.0     |
| RF   | lemma          | 48.3        | 40.3     | 60.3     |
|      | Roth and Frank | 54.8        | 59.7     | 50.7     |
|      | PARMA          | <b>57.6</b> | 52.4     | 64.0     |
| MTC  | lemma          | 42.1        | 51.3     | 35.7     |
|      | PARMA          | <b>59.2</b> | 73.4     | 49.6     |

Table 1: PARMA outperforms the baseline lemma matching system on the three test sets, drawn from the Extended Event Coreference Bank, Roth and Frank’s data, and our set created from the Multiple Translation Corpora. PARMA achieves a higher F1 and recall score than Roth and Frank’s reported result.

and F1 as the harmonic mean of the two. Results for EECB and MTC reflect 5-fold cross validation, and RF uses the given dev/test split.

**Lemma baseline** Following Roth and Frank we include a lemma baseline, in which two predicates or arguments align if they have the same lemma.<sup>9</sup>

## 4 Results

On every dataset PARMA significantly improves over the lemma baselines (Table 1). On RF, compared to Roth and Frank, the best published method for this task, we also improve, making PARMA the state of the art system for this task. Furthermore, we expect that the smallest improvements over Roth and Frank would be on RF, since there is little training data. We also note that compared to Roth and Frank we obtain much higher recall but lower precision.

We also observe that MTC was more challenging than the other datasets, with a lower lemma baseline. Figure 2 shows the correlation between document similarity and document F1 score for RF and MTC. While for RF these two measures are correlated, they are uncorrelated for MTC. Additionally, there is more data in the MTC dataset which has low cosine similarity than in RF.

## 5 Conclusion

PARMA achieves state of the art performance on three datasets for predicate argument alignment. It builds on the development of lexical semantic resources and provides a platform for learning to utilize these resources. Additionally, we show that

<sup>9</sup>We could not reproduce lemma from Roth and Frank (shown in Table 1) due to a difference in lemmatizers. We obtained 55.4; better than their system but worse than PARMA.

task difficulty can be strongly tied to lexical similarity if the evaluation dataset is not chosen carefully, and this provides an artificially high baseline in previous work. PARMA is robust to drops in lexical similarity and shows large improvements in those cases. PARMA will serve as a useful benchmark in determining the value of more sophisticated models of predicate-argument alignment, which we aim to address in future work.

While our system is fully supervised, and thus dependent on manually annotated examples, we observed here that this requirement may be relatively modest, especially for in-domain data.

## Acknowledgements

We thank JHU HLTCOE for hosting the winter MiniSCALE workshop that led to this collaborative work. This material is based on research sponsored by the NSF under grant IIS-1249516 and DARPA under agreement number FA8750-13-2-0017 (the DEFT program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of NSF, DARPA, or the U.S. Government.

## References

- Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*, pages 1–8. Association for Computational Linguistics.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL ’98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, pages 1412–1422, Stroudsburg, PA, USA. Association for Computational Linguistics.

- 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.
- Hal Daumé. 2007. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256.
- Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Conference on Computational Linguistics (Coling)*.
- Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280(1):20–32.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. Association for Computational Linguistics.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://www.cs.umass.edu/mccallum/mallet>.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *AKBC-WEKEX Workshop at NAACL 2012*, June.
- Robert Parker, David Graff, Jumbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition.
- Michael Roth and Anette Frank. 2012. Aligning predicate argument structures in monolingual comparable texts: A new corpus for a new task. In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 218–227, Montréal, Canada, 7-8 June. Association for Computational Linguistics.
- Xuchen Yao, Benjamin Van Durme, Peter Clark, and Chris Callison-Burch. 2013. Answer extraction as sequence tagging with tree edit distance. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, December.

# Aggregated Word Pair Features for Implicit Discourse Relation Disambiguation

**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

We present a reformulation of the word pair features typically used for the task of disambiguating implicit relations in the Penn Discourse Treebank. Our word pair features achieve significantly higher performance than the previous formulation when evaluated without additional features. In addition, we present results for a full system using additional features which achieves close to state of the art performance without resorting to gold syntactic parses or to context outside the relation.

## 1 Introduction

Discourse relations such as *contrast* and *causality* are part of what makes a text coherent. Being able to automatically identify these relations is important for many NLP tasks such as generation, question answering and textual entailment. In some cases, discourse relations contain an explicit marker such as *but* or *because* which makes it easy to identify the relation. Prior work (Pitler and Nenkova, 2009) showed that where explicit markers exist, the class of the relation can be disambiguated with f-scores higher than 90%.

Predicting the class of *implicit* discourse relations, however, is much more difficult. Without an explicit marker to rely on, work on this task initially focused on using lexical cues in the form of *word pairs* mined from large corpora where they appear around an explicit marker (Marcu and Echihiabi, 2002). The intuition is that these pairs will tend to represent semantic relationships which are related to the discourse marker (for example, word pairs often appearing around *but* may tend to be antonyms). While this approach showed some success and has been used extensively in later work, it has been pointed out by multiple authors that many of the most useful word pairs

are pairs of very common functional words, which contradicts the original intuition, and it is hard to explain why these are useful.

In this work we focus on the task of identifying and disambiguating implicit discourse relations which have no explicit marker. In particular, we present a reformulation of the word pair features that have most often been used for this task in the past, replacing the sparse lexical features with dense aggregated score features. This is the main contribution of our paper. We show that our formulation outperforms the original one while requiring less features, and that using a stop list of functional words does not significantly affect performance, suggesting that these features indeed represent semantically related content word pairs.

In addition, we present a system which combines these word pairs with additional features to achieve near state of the art performance without the use of syntactic parse features and of context outside the arguments of the relation. Previous work has attributed much of the achieved performance to these features, which are easy to get in the experimental setting but would be less reliable or unavailable in other applications.<sup>1</sup>

## 2 Related Work

This line of research began with (Marcu and Echihiabi, 2002), who used a small number of unambiguous explicit markers and patterns involving them, such as [Arg1, *but* Arg2] to collect sets of word pairs from a large corpus using the cross-product of the words in Arg1 and Arg2. The authors created a feature out of each pair and built a naive bayes model directly from the unannotated corpus, updating the priors and posteriors using maximum likelihood. While they demonstrated

<sup>1</sup>Reliable syntactic parses are not always available in domains other than newswire, and context (preceding relations, especially explicit relations) is not always available in some applications such as generation and question answering.

some success, their experiments were run on data that is unnatural in two ways. First, it is balanced. Second, it is constructed with the same unsupervised method they use to extract the word pairs - by assuming that the patterns correspond to a particular relation and collecting the arguments from an unannotated corpus. Even if the assumption is correct, these arguments are really taken from explicit relations with their markers removed, which as others have pointed out (Blair-Goldensohn et al., 2007; Pitler et al., 2009) may not look like true implicit relations.

More recently, implicit relation prediction has been evaluated on annotated implicit relations from the Penn Discourse Treebank (Prasad et al., 2008). PDTB uses hierarchical relation types which abstract over other theories of discourse such as RST (Mann and Thompson, 1987) and SDRT (Asher and Lascarides, 2003). It contains 40,600 annotated relations from the WSJ corpus. Each relation has two arguments, Arg1 and Arg2, and the annotators decide whether it is explicit or implicit.

The first to evaluate directly on PDTB in a realistic setting were Pitler et al. (2009). They used word pairs as well as additional features to train four binary classifiers, each corresponding to one of the high-level PDTB relation classes. Although other features proved to be useful, word pairs were still the major contributor to most of these classifiers. In fact, their best system for *comparison* included only the word pair features, and for all other classes other than *expansion* the word pair features alone achieved an f-score within 2 points of the best system. Interestingly, they found that training the word pair features on PDTB itself was more useful than training them on an external corpus like Marcu and Echihabi (2002), although in some cases they resort to information gain in the external corpus for filtering the word pairs.

Zhou et al. (2010) used a similar method and added features that explicitly try to predict the *implicit marker* in the relation, increasing performance. Most recently to the best of our knowledge, Park and Cardie (2012) achieved the highest performance by optimizing the feature set. Another work evaluating on PDTB is (Lin et al., 2009), who are unique in evaluating on the more fine-grained second-level relation classes.

## 3 Word Pairs

### 3.1 The Problem: Sparsity

While Marcu and Echihabi (2002)'s approach of training a classifier from an unannotated corpus provides a relatively large amount of training data, this data does not consist of true implicit relations. However, the approach taken by Pitler et al. (2009) and repeated in more recent work (training directly on PDTB) is problematic as well: when training a model with so many sparse features on a dataset the size of PDTB (there are 22,141 non-explicit relations overall), it is likely that many important word pairs will not be seen in training.

In fact, even the larger corpus of Marcu and Echihabi (2002) may not be quite large enough to solve the sparsity issue, given that the number of word pairs is quadratic in the vocabulary. Blair-Goldensohn et al. (2007) report that using even a very small stop list (25 words) significantly reduces performance, which is counter-intuitive. They attribute this finding to the sparsity of the feature space. An analysis in (Pitler et al., 2009) also shows that the top word pairs (ranked by information gain) all contain common functional words, and are not at all the semantically-related content words that were imagined. In the case of some reportedly useful word pairs (the-and; in-the; the-of...) it is hard to explain how they might affect performance except through overfitting.

### 3.2 The Solution: Aggregation

Representing each word pair as a single feature has the advantage of allowing the weights for each pair to be learned directly from the data. While powerful, this approach requires large amounts of data to be effective.

Another possible approach is to aggregate some of the pairs together and learn weights from the data only for the aggregated sets of words. For this approach to be effective, the pairs we choose to group together should have similar meaning with regard to predicting the relation.

Biran and Rambow (2011) is to our knowledge the only other work utilizing a similar approach. They used aggregated word pair set features to predict whether or not a sentence is argumentative. Their method is to group together word pairs that have been collected around the same explicit discourse marker: for every discourse marker such as *therefore* or *however*, they have a single feature whose value depends only on the word pairs

collected around that marker. This is reasonable given the intuition that the marker pattern is unambiguous and points at a particular relation. Using one feature per marker can be seen as analogous (yet complementary) to Zhou et al. (2010)’s approach of trying to predict the implicit connective by giving a score to each marker using a language model.

This work uses binary features which only indicate the appearance of one or more of the pairs. The original frequencies of the word pairs are not used anywhere. A more powerful approach is to use an informed function to weight the word pairs used inside each feature.

### 3.3 Our Approach

Our approach is similar in that we choose to aggregate word pairs that were collected around the same explicit marker. We first assembled a list of all 102 discourse markers used in PDTB, in both explicit and implicit relations.<sup>2</sup>

Next, we extract word pairs for each marker from the Gigaword corpus by taking the cross product of words that appear in a sentence around that marker. This is a simpler approach than using patterns - for example, the marker *because* can appear in two patterns: [Arg1 *because* Arg2] and [*because* Arg1, Arg2], and we only use the first. We leave the task of listing the possible patterns for each of the 102 markers to future work because of the significant manual effort required. Meanwhile, we rely on the fact that we use a very large corpus and hope that the simple pattern [Arg1 *marker* Arg2] is enough to make our features useful. There are, of course, markers for which this pattern does not normally apply, such as *by comparison* or *on one hand*. We expect these features to be down-weighted by the final classifier, as explained at the end of this section. When collecting the pairs, we stem the words and discard pairs which appear only once around the marker.

We can think of each discourse marker as having a corresponding unordered “document”, where each word pair is a term with an associated frequency. We want to create a feature for each marker such that for each data instance (that is, for each potential relation in the PDTB data) the value for the feature is the relevance of the marker document to the data instance.

<sup>2</sup>in implicit relations, there is no marker in the text but the implicit marker is provided by the human annotators

Each data instance in PDTB consists of two arguments, and can therefore also be represented as a set of word pairs extracted from the cross-product of the two arguments. To represent the relevance of the instance to each marker, we set the value of the marker feature to the cosine similarity of the data instance and the marker’s “document”, where each word pair is a dimension.

While the terms (i.e. word pairs) of the data instance are weighted by simple occurrence count, we weight the terms in each marker’s document with tf-idf, where tf is defined in one of two ways: normalized term frequency ( $\frac{\text{count}(t)}{\max\{\text{count}(s,d):s \in d\}}$ ) and pointwise mutual information ( $\log \frac{\text{count}(t)}{\text{count}(w_1) * \text{count}(w_2)}$ ), where  $w_1$  and  $w_2$  are the member words of the pair. Idf is calculated normally given that the set of all documents is defined as the 102 marker documents.

We then train a binary classifier (logistic regression) using these 102 features for each of the four high-level relations in PDTB: *comparison*, *contingency*, *expansion* and *temporal*. To make sure our results are comparable to previous work, we treat *EntRel* relations as instances of *expansion* and use sections 2-20 for training and sections 21-22 for testing. We use a ten fold stratified cross-validation of the training set for development. Explicit relations are excluded from all data sets.

As mentioned earlier, there are markers that do not fit the simple pattern we use. In particular, some markers always or often appear as the first term of a sentence. For these, we expect the list of word pairs to be empty or almost empty, since in most sentences there are no words on the left (and recall that we discard pairs that appear only once). Since the features created for these markers will be uninformative, we expect them to be weighted down by the classifier and have no significant effect on prediction.

## 4 Evaluation of Word Pairs

For our main evaluation, we evaluate the performance of word pair features when used with no additional features. Results are shown in Table 1. Our word pair features outperform the previous formulation (represented by the results reported by (Pitler et al., 2009), but used by virtually all previous work on this task). For most relation classes, tf is significantly better than pmi.<sup>3</sup>

<sup>3</sup>Significance was verified for our own results in all experiments shown in this paper with a standard t-test

|                        | Comparison           | Contingency        | Expansion            | Temporal             |
|------------------------|----------------------|--------------------|----------------------|----------------------|
| Pitler et al., 2009    | 21.96 (56.59)        | <b>45.6 (67.1)</b> | 63.84 (60.28)        | 16.21 (61.98)        |
| tf-idf, no stop list   | 23 (61.72)           | 44.03 (66.78)      | <b>66.48 (60.93)</b> | <b>19.54 (68.09)</b> |
| pmi-idf, no stop list  | <b>24.38 (61.72)</b> | 38.96 (61.52)      | 62.22 (57.26)        | 16 (65.53)           |
| tf-idf, with stop list | 23.77                | 44.33              | 65.33                | 16.98                |

Table 1: Main evaluation. F-measure (accuracy) for various implementations of the word pairs features

|                       | Comparison    | Contingency   | Expansion     | Temporal      |
|-----------------------|---------------|---------------|---------------|---------------|
| Best System           | 25.4 (63.36)  | 46.94 (68.09) | 75.87 (62.84) | 20.23 (68.35) |
| features used         | pmi+1,2,3,6   | tf+ALL        | tf+8          | tf+3,9        |
| Pitler et al., 2009   | 21.96 (56.59) | 47.13 (67.3)  | 76.42 (63.62) | 16.76 (63.49) |
| Zhou et al., 2010     | 31.79 (58.22) | 47.16 (48.96) | 70.11 (54.54) | 20.3 (55.48)  |
| Park and Cardie, 2012 | 31.32 (74.66) | 49.82 (72.09) | 79.22 (69.14) | 26.57 (79.32) |

Table 2: Secondary evaluation. F-measure (accuracy) for the best systems. *tf* and *pmi* refer to the word pair features used (by *tf* implementation), and the numbers refer to the indices of Table 3

|   |            | Comp. | Cont. | Exp.  | Temp. |
|---|------------|-------|-------|-------|-------|
| 1 | WordNet    | 20.07 | 34.07 | 52.96 | 11.58 |
| 2 | Verb Class | 14.24 | 24.84 | 49.6  | 10.04 |
| 3 | MPN        | 23.84 | 38.58 | 49.97 | 13.16 |
| 4 | Modality   | 17.49 | 28.92 | 13.84 | 10.72 |
| 5 | Polarity   | 16.46 | 26.36 | 65.15 | 11.58 |
| 6 | Affect     | 18.62 | 31.59 | 59.8  | 13.37 |
| 7 | Similarity | 20.68 | 34.5  | 43.16 | 12.1  |
| 8 | Negation   | 8.28  | 22.47 | 75.87 | 11.1  |
| 9 | Length     | 20.75 | 31.28 | 65.72 | 10.19 |

Table 3: F-measure for each feature category

We also show results using a stop list of 50 common functional words. The stop list has only a small effect on performance except in the *temporal* class. This may be because of functional words like *was* and *will* which have a temporal effect.

## 5 Other Features

For our secondary evaluation, we include additional features to complement the word pairs. Previous work has relied on features based on the gold parse trees of the Penn Treebank (which overlaps with PDTB) and on contextual information from relations preceding the one being disambiguated. We intentionally limit ourselves to features that do not require either so that our system can be readily used on arbitrary argument pairs.

**WordNet Features:** We define four features based on WordNet (Fellbaum, 1998) - *Synonyms*, *Antonyms*, *Hypernyms* and *Hyponyms*. The values are the counts of word pairs in the cross-product of the words in the arguments that have the particular relation (synonymy, antonymy etc) between them.

**Verb Class:** This is the count of pairs of verbs from Arg1 and Arg2 that share the same class, de-

finied as the highest level Levin verb class (Levin, 1993) from the LCS database (Dorr, 2001).

**Money, Percentages and Numbers (MPN):** The counts of currency symbols/abbreviations, percentage signs or cues (“percent”, “BPS”...) and numbers in each argument.

**Modality:** Presence or absence of each English modal in each argument.

**Polarity:** Based on MPQA (Wilson et al., 2005). We include the counts of positive and negative words according to the MPQA subjectivity lexicon for both arguments. Unlike Pitler et al. (2009), we do not use neutral polarity features. We also do not explicitly group negation with polarity (although we do have separate negation features).

**Affect:** Based on the Dictionary of Affect in Language (Whissell, 1989). Each word in the DAL gets a score for three dimensions - *pleasantness* (pleasant - unpleasant), *activation* (passive - active) and *imagery* (hard to imagine - easy to imagine). We use the average score for each dimension in each argument as a feature.

**Content Similarity:** We use the cosine similarity and word overlap of the arguments as features.

**Negation:** Presence or absence of negation terms in each of the arguments.

**Length:** The ratio between the lengths (counts of words) of the arguments.

## 6 Evaluation of Additional Features

For our secondary evaluation, we present results for each feature category on its own in Table 3 and for our best system for each of the relation classes in Table 2. We show results for the best systems from (Pitler et al., 2009), (Zhou et al., 2010) and



(Park and Cardie, 2012) for comparison.

## 7 Conclusion

We presented an aggregated approach to word pair features and showed that it outperforms the previous formulation for all relation types but *contingency*. This is our main contribution. With this approach, using a stop list does not have a major effect on results for most relation classes, which suggests most of the word pairs affecting performance are content word pairs which may truly be semantically related to the discourse structure.

In addition, we introduced the new and useful *WordNet*, *Affect*, *Length* and *Negation* feature categories. Our final system outperformed the best system from Pitler et al. (2009), who used mostly similar features, for *comparison* and *temporal* and is competitive with the most recent state of the art systems for *contingency* and *expansion* without using any syntactic or context features.

## 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

- Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in Natural Language Processing Series. Cambridge University Press.
- Or Biran and Owen Rambow. 2011. Identifying justifications in written dialog by classifying text as argumentative. *International Journal of Semantic Computing*, 5(4):363–381, December.
- Sasha Blair-Goldensohn, Kathleen McKeown, and Owen Rambow. 2007. Building and refining rhetorical-semantic relation models. In *HLT-NAACL*, pages 428–435. The Association for Computational Linguistics.
- Bonnie J. Dorr. 2001. *LCS Verb Database, Online Software Database of Lexical Conceptual Structures*

*and Documentation*. University Of Maryland College Park.

- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press.
- Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University Of Chicago Press.
- Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351.
- William C. Mann and Sandra A. Thompson. 1987. *Rhetorical Structure Theory: A theory of text organization*. Technical Report ISI/RS-87-190, ISI.
- Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *ACL*, pages 368–375. ACL.
- Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 108–112.
- Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *ACL/IJCNLP (Short Papers)*, pages 13–16. The Association for Computer Linguistics.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *ACL/IJCNLP*, pages 683–691. The Association for Computer Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *In Proceedings of LREC*.
- Cynthia M. Whissell. 1989. *The dictionary of affect in language*.
- 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*, pages 347–354.
- Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

# Implicatures and Nested Beliefs in Approximate Decentralized-POMDPs

Adam Vogel, Christopher Potts, and Dan Jurafsky

Stanford University

Stanford, CA, USA

{acvogel, cgpotts, jurafsky}@stanford.edu

## Abstract

Conversational implicatures involve reasoning about multiply nested belief structures. This complexity poses significant challenges for computational models of conversation and cognition. We show that agents in the multi-agent Decentralized-POMDP reach implicature-rich interpretations simply as a by-product of the way they reason about each other to maximize joint utility. Our simulations involve a reference game of the sort studied in psychology and linguistics as well as a dynamic, interactional scenario involving implemented artificial agents.

## 1 Introduction

Gricean conversational implicatures (Grice, 1975) are inferences that listeners make in order to reconcile the speaker’s linguistic behavior with the assumption that the speaker is cooperative. As Grice conceived of them, implicatures crucially involve reasoning about multiply-nested belief structures: roughly, for  $p$  to count as an implicature, the speaker must believe that the listener will infer that the speaker believes  $p$ . This complexity makes implicatures an important testing ground for models of conversation and cognition.

Implicatures have received considerable attention in the context of simple reference games in which the listener uses the speaker’s utterance to try to identify the speaker’s intended referent (Rosenberg and Cohen, 1964; Clark and Wilkes-Gibbs, 1986; Dale and Reiter, 1995; DeVault and Stone, 2007; Krahmer and van Deemter, 2012). Many implicature patterns can be embedded in these games using specific combinations of potential referents and message sets. The paradigm has proven fruitful not only for evaluating computational models (Golland et al., 2010; Degen and

Franke, 2012; Frank and Goodman, 2012; Rohde et al., 2012; Bergen et al., 2012) but also for studying children’s pragmatic abilities without implicitly assuming they have mastered challenging linguistic structures (Stiller et al., 2011).

In this paper, we extend these results beyond simple reference games to full decision-problems in which the agents reason about language and action together over time. To do this, we use the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) to implement agents that are capable of manipulating the multiply-nested belief structures required for implicature calculation. Optimal decision making in Dec-POMDPs is NEXP complete, so we employ the single-agent POMDP approximation of Vogel et al. (2013). We show that agents in the Dec-POMDP reach implicature-rich interpretations simply as a by-product of the way they reason about each other to maximize joint utility. Our simulations involve a reference game and a dynamic, interactional scenario involving implemented artificial agents.

## 2 Decision-Theoretic Communication

The Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Bernstein et al., 2002) is a multi-agent generalization of the POMDP, where agents act to maximize a shared utility function. Formally, a Dec-POMDP consists of a tuple  $(S, A, O, R, T, \Omega, b_0, \gamma)$ .  $S$  is a finite set of states,  $A$  is the set of actions,  $O$  is the set of observations, and  $T(s'|a_1, a_2, s)$  is the *transition distribution* which determines what effect the joint action  $(a_1, a_2)$  has on the state of the world. The true state  $s \in S$  is not observable to the agents, who must utilize observations  $o \in O$ , which are emitted after each action according to the *observation distribution*  $\Omega(o_1, o_2|s', a)$ . The *reward function*  $R(s, a_1, a_2)$  represents the goal of the agents, who act to maximize expected reward. Lastly,  $b_0 \in \Delta(S)$  is the initial belief state and

$\gamma \in [0, 1]$  is the discount factor.

The true state of the world  $s \in S$  is not observable to either agent. In single-agent POMDPs, agents maintain a *belief state*  $b(s) \in \Delta(S)$ , which is a distribution over states. Agents acting in Dec-POMDPs must take into account not only their beliefs about the state of the world, but also the beliefs of their partners, leading to *nested* belief states. In the model presented here, our agent models the other agent’s beliefs about the state of the world, and assumes that the other agent does not take into account our own beliefs, a common approach (Gmytrasiewicz and Doshi, 2005).

Agents make decisions according to a *policy*  $\pi_i : \Delta(S) \rightarrow A$  which maximizes the discounted expected reward  $\sum_{t=0}^{\infty} \gamma^t \mathbb{E}[R(s^t, a_1^t, a_2^t) | b_0, \pi_1, \pi_2]$ . Using the assumption that the other agent tracks one less level of belief, we can solve for the other agent’s policy  $\bar{\pi}$ , which allows us to estimate his actions and beliefs over time. To construct policies, we use Perseus (Spaan and Vlassis, 2005), a point-based value iteration algorithm.

Even tracking just one level of nested beliefs quickly leads to a combinatorial explosion in the number of belief states the other agent might have. This causes decision making in Dec-POMDPs to be NEXP complete, limiting their application to problems with only a handful of states (Bernstein et al., 2002). To ameliorate this difficulty, we use the method of Vogel et al. (2013), which creates a single-agent approximation to the full Dec-POMDP. To form this single-agent POMDP, we augment the state space to be  $S \times S$ , where the second set of state variables allows us to model the other agent’s beliefs. We maintain a *point estimate*  $\bar{b}$  of the other agent’s beliefs, which is formed by summing out observations  $O$  that the other player might have received. To accomplish this, we factor the transition distribution into two terms:  $T((s', \bar{s}') | a, \bar{\pi}(\bar{s}), (s, \bar{s})) = \bar{T}(\bar{s}' | s', a, \bar{\pi}(\bar{s}), (s, \bar{s})) T(s' | a, \bar{\pi}(\bar{s}), (s, \bar{s}))$ . This observation marginalization can be folded into the transition distribution  $\bar{T}(\bar{s}' | s', a, \bar{\pi}(\bar{s}), (s, \bar{s}))$ :

$$\begin{aligned} \bar{T}(\bar{s}' | s', a, \bar{\pi}(\bar{s}), (s, \bar{s})) &= \Pr(\bar{s}' | s', a, \bar{\pi}(\bar{s}), (s, \bar{s})) \\ &= \sum_{\bar{o} \in O} \left( \frac{\Omega(\bar{o} | s', a, \bar{\pi}(\bar{s})) T(\bar{s}' | a, \bar{\pi}(\bar{s}), \bar{s})}{\sum_{\bar{s}''} \Omega(\bar{o} | \bar{s}'', a, \bar{\pi}(\bar{s})) T(\bar{s}'' | a, \bar{\pi}(\bar{s}), \bar{s})} \right. \\ &\quad \left. \times \Omega(\bar{o} | s', a, \bar{\pi}(\bar{s})) \right) \end{aligned} \quad (1)$$

Communication is treated as another type of ob-

servation, with messages coming from a finite set  $M$ . Each message  $m \in M$  has the semantics  $\Pr(s|m)$ , which represents the probability that the world is in state  $s \in S$  given that  $m$  is true. Messages  $m$  received from a partner are combined with perceptual observations  $o \in O$ , to form a joint observation  $(m, o)$ .

A *literal listener*, denoted L, interprets messages according to this semantics, without taking into account the beliefs of the speaker. L assumes that the perceptual observations and messages are conditionally independent given the state of the world. Using Bayes’ rule, the literal listener’s joint observation/message distribution is

$$\begin{aligned} \Pr((o, m) | s, s', a) &= \Omega(o | s', a) \Pr(m | s) \\ &= \Omega(o | s', a) \frac{\Pr(s|m) \Pr(m)}{\sum_{m' \in M} \Pr(s|m') \Pr(m')} \end{aligned} \quad (2)$$

The  $\Pr(m)$  prior over messages can be estimated from corpus data, but we use a uniform prior for simplicity.

A *literal speaker*, denoted S, produces messages according to the most descriptive term:

$$\pi_S(s) = \arg \max_{m \in M} p(s|m). \quad (3)$$

The literal speaker does not model the beliefs of the listener.

To interpret implicatures, a *level-one listener*, denoted L(S), models the beliefs a literal speaker must have had to produce an utterance:  $\Pr(m|s) = \mathbb{1}[\bar{\pi}_S(s) = m]$ , where  $\bar{\pi}_S$  is the level-one listener’s estimate of the speaker’s policy. In this setting, we denote the level-one listener’s estimate of the speaker’s belief as  $\bar{s}$ , yielding the belief update equation

$$\begin{aligned} \Pr((o, m) | (s, \bar{s}), (s', \bar{s}'), a, \bar{\pi}_S(\bar{s})) &= \\ \Omega(o | s', a) \mathbb{1}[\bar{\pi}_S(\bar{s}) = m] \end{aligned} \quad (4)$$

The literal semantics of messages is not explicitly included in the level-one listener’s belief update. Instead, when he solves for the literal speaker’s policy  $\bar{\pi}_S$ , the meaning of a message is the set of beliefs that would lead the literal speaker to produce the utterance.

A *level-one speaker*, S(L), produces utterances to influence a literal listener, and a *level-two listener*, L(S(L)), uses two levels of belief nesting to interpret utterances as the beliefs that a level-one speaker might have to produce that utterance. At each level of nesting, we apply the marginalized



(a) Scenario.

| Message   | $r_1$         | $r_2$         | $r_3$         |
|-----------|---------------|---------------|---------------|
| moustache | $\frac{1}{2}$ | $\frac{1}{2}$ | 0             |
| glasses   | 0             | $\frac{1}{2}$ | $\frac{1}{2}$ |
| hat       | 0             | 0             | 1             |

(b) Literal interpretations.

| Message   | $r_1$ | $r_2$ | $r_3$ |
|-----------|-------|-------|-------|
| moustache | 1     | 0     | 0     |
| glasses   | 0     | 1     | 0     |
| hat       | 0     | 0     | 1     |

(c) Implicature-rich interpretations.

Figure 1: A simple reference game. The matrices give distributions  $\Pr(t = r_i | \text{utterance})$

belief-state approach of (Vogel et al., 2013), augmenting the state space with another copy of the underlying world state space, where the new copy represents the next level of belief. For instance, the  $L(S(L))$  agent will make decisions in the  $S \times S \times S$  space. For an  $L(S(L))$  state  $(s, \bar{s}, \hat{s})$ ,  $s$  is the true state of the world,  $\bar{s}$  is the speaker’s belief of the state of the world, and  $\hat{s}$  is the speaker’s belief of the listener’s beliefs. In the next two sections we show how a level-one and level-two listener infer implicatures.

### 3 Reference Game Implicatures

Fig. 1a is the scenario for a reference game of the sort pioneered by Rosenberg and Cohen (1964) and Dale and Reiter (1995). The potential referents are  $r_1$ ,  $r_2$ , and  $r_3$ . Speakers use a restricted vocabulary consisting of three messages: ‘moustache’, ‘glasses’, and ‘hat’. The speaker is assigned a referent  $r_i$  (hidden from the listener) and produces a message on that basis. The speaker and listener share the goal of having the listener identify the speaker’s intended referent  $r_i$ .

Fig. 1b depicts the literal interpretations for this game. It looks like the listener’s chances of success are low. Only ‘hat’ refers unambigu-

ously. However, the language and scenario facilitate *scalar implicature* (Horn, 1972; Harnish, 1979; Gazdar, 1979). Briefly, the scalar implicature pattern is that a speaker who is knowledgeable about the relevant domain will choose a communicatively weak utterance  $U$  over a communicatively stronger utterance  $U'$  iff  $U'$  is false (assuming  $U$  and  $U'$  are relevant). The required sense of communicative strength encompasses logical entailments as well as more particularized pragmatic partial orders (Hirschberg, 1985).

In our scenario, ‘hat’ is stronger than ‘glasses’: the referents wearing a hat are a proper subset of those wearing glasses. Thus, given the players’ goal, if the speaker says ‘glasses’, the listener should draw the scalar implicature that ‘hat’ is false. Thus, ‘glasses’ comes to unambiguously refer to  $r_2$  (Fig. 1c, line 2). Similarly, though ‘moustache’ and ‘glasses’ do not *literally* stand in the specific–general relationship needed for scalar implicature, they do with ‘glasses’ *pragmatically* associated with  $r_2$  (Fig. 1c, line 1).

Our implementation of these games as Dec-POMDPs mirrors their intuitive description and their treatment in iterated best response models (Jäger, 2007; Jäger, 2012; Franke, 2009; Frank and Goodman, 2012). The state space  $S$  encodes the attributes of the referents (e.g.,  $\mathbf{hat}(r_2) = \mathbf{T}$ ,  $\mathbf{glasses}(r_1) = \mathbf{F}$ ) and includes a target variable  $t$  identifying the speaker’s referent (hidden from the listener). The speaker has three speech actions, identified with the three messages. The listener has four actions: ‘listen’ plus a ‘choose’ action  $c_i$  for each referent  $r_i$ . The set of observations  $O$  is just the set of messages (construed as utterances). The agents receive a positive reward iff the listener action  $c_i$  corresponds to the speaker’s target  $t$ . Because this is a one-step reference game, the transition distribution  $T$  is the identity distribution.

The *literal listener*  $L$  interprets utterances as a truth-conditional speaker would produce them (Fig. 1b). The *level-one speaker*  $S(L)$  augments the state space with a variable ‘listener\_target’ and models  $L$ ’s beliefs  $\bar{b}$  using the approximate methods of Sec. 2. Crucially, the optimal speaker policy  $\pi_{S(L)}$  is such that  $\pi_{S(L)}(t=r_3) = \text{‘hat’}$  and  $\pi_{S(L)}(t=r_1) = \text{‘moustache’}$ . The *level-two listener*  $L(S(L))$  models  $S(L)$  via an estimate of the ‘listener\_target’ variable. For each speech action  $m$ ,  $L(S(L))$  considers all values of  $t$  and the likeli-

hood that  $S(L)$  would have produced  $m$ :

$$\Pr(t=r_i|m) \propto \mathbb{1}[\bar{\pi}_{S(L)}(t=r_i) = m]$$

Since  $S(L)$  uses ‘hat’ to describe  $r_3$  and ‘moustache’ to describe  $r_1$ ,  $L(S(L))$  correctly infers that ‘glasses’ refers to  $r_2$ , completing Fig. 1c’s full implicature-rich pattern of mutual exclusivity (Clark, 1987; Frank et al., 2009).

This basic pattern is robustly attested empirically in human data. The experimental data are, of course, invariably less crisp than our idealized model predicts, but many important sources of variation could be brought into our model, with the addition of strong salience priors (Frank and Goodman, 2012; Stiller et al., 2011), assumptions about bounded rationality (Camerer et al., 2004; Franke, 2009), and a ‘soft-max’ view of the listener (Frank et al., 2009).

#### 4 Cards World Implicatures

The Cards corpus<sup>1</sup> contains 1266 metadata-rich transcripts from a two-player chat-based game. The world is a simple maze in which a deck of cards has been distributed. The players’ goal is to find specific subsets of the cards, subject to a variety of constraints on what they can see and do. The Dec-POMDP-based agents of Vogel et al. (2013) play a simplified version in which the goal is to be co-located with a single card. Vogel et al. show that their agents’ linguistic behavior is broadly Gricean. However, their agents’ language is too simple to reveal implicatures. The present section remedies this shortcoming. Implicature-rich interpretations are an immediate consequence.

We implement the simplified Cards tasks as follows. The state space  $S$  is composed of the location of each player and the location of the card. The transition distribution  $T(s'|s, a_1, a_2)$  encodes the outcome of movement actions. Agents receive one of two sensor observations, indicating whether the card is at their current location. The players are rewarded when they are both located on the card. Each player begins knowing his own location, but not the location of the other player nor of the card.

The players have four movement actions (‘up’, ‘down’, ‘left’, ‘right’) and nine speech actions interpreted as identifying card locations. Fig. 2 depicts these utterances as a partial order determined by entailment. These general-to-specific relation-

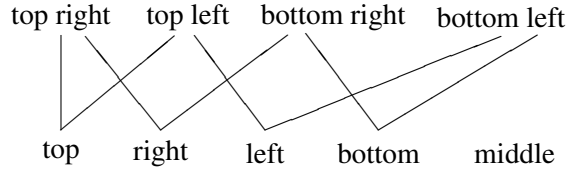


Figure 2: Cards world utterance actions.

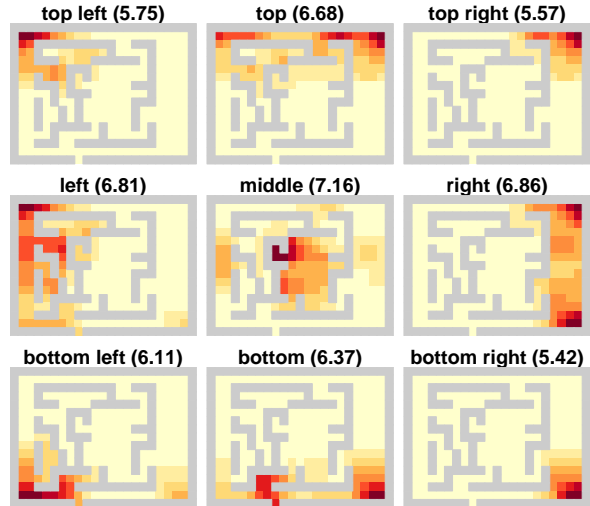


Figure 3: Literal interpretations derived from the Cards corpus. The entropy of each distribution is included in parentheses. Each term is estimated from all tokens that contain it, which washes out implicature-rich usage, thereby providing our model with an empirically-grounded literal start.

ships show that the language can support scalar conversational implicatures.<sup>2</sup>

Fig. 2 is not entirely appropriate in our setting, however. Our expressions are vague; there is no sharp boundary between, e.g., ‘top’ and ‘bottom’, nor is it clear where ‘top right’ begins. To model this vagueness, we analyze each message  $m$  as denoting a conditional distribution  $\Pr(x|m)$  over grid squares  $x$  in the gameboard. These distributions are derived from human–human Cards interactions using the data and methods of Potts (2012). Of course, there is a tension here: our model assumes that we begin with literal interpretations, but human–human data will reflect pragmatically-enriched usage. To get around this, we approximate literal interpretations by deriving each term’s distribution from all the corpus tokens that contain it. For example, the distribution for ‘top’ is

<sup>2</sup>Our agents cannot produce modified versions of ‘middle’ like ‘middle right’. These would be synonymous with implicature-enriched general terms. We work with a simple cost-function that treats all forms alike, but future versions of this work will incorporate more realistic form-based costs.

<sup>1</sup><http://cardscorpus.christopherpotts.net>

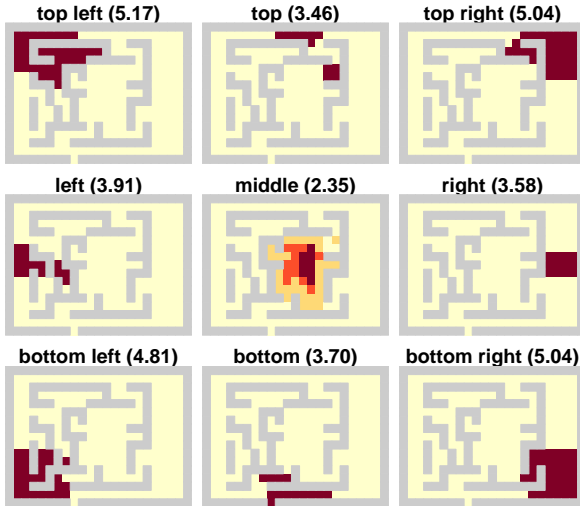


Figure 4: Implicature-rich interpretations, derived using the level-one listener L(S).

estimated not only from ‘top’ but also from ‘top right’, ‘middle right’, and so forth. The denotation for ‘top right’ excludes simple ‘top’ and ‘right’ utterances but includes expressions like ‘very top right’. This semantics washes out any implicature patterns, thereby giving us a proper literal starting point. Fig. 3 shows these denotations for the full set of expressions. The entailment relations from Fig. 2 are (fuzzily) evident. For example, the areas of high probability for ‘right’ properly contain the areas of high probability for ‘top right’.

To show how the Dec-POMDP model delivers implicatures, we begin with a *literal speaker*  $S$  who does not consider the location of the other player and instead searches the board until he finds the card. After finding it, he communicates the referring expression with highest literal probability for his location, using the distributions from Fig. 3. We denote the literal speaker’s policy by  $\pi_S$ . The *level-one listener* L(S) tracks an estimate of  $S$ ’s location and beliefs about the card location. Using the approximation defined in Sec. 2, L(S) interprets an utterance  $m$  as  $\Pr(m|s) = \mathbb{1}[\bar{\pi}_S(s) = m]$ . Thus, the meaning of each  $m$  is the set of beliefs that  $S$  might have to produce this utterance. Fig. 4 shows how L(S) interprets each message. The meaning of general terms like ‘top’ and ‘right’ now exclude their modified counterparts. This is evident in the lack of overlap between high-probability areas and in the lower entropy values.

Direct evaluation of this result against the corpus data is not possible, because the corpus does not encode interpretations. However, we expect

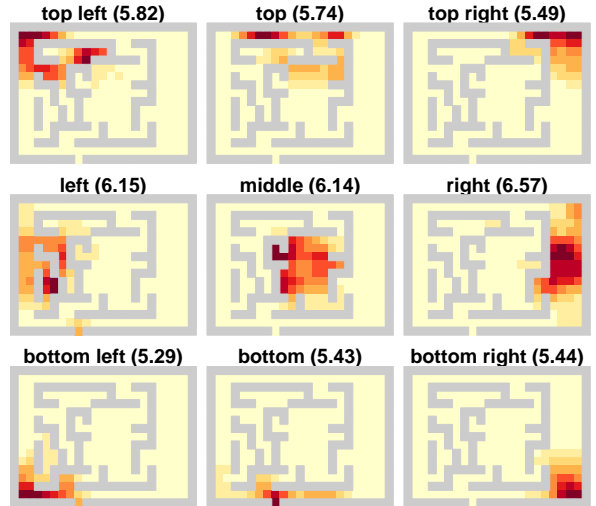


Figure 5: Distributions reflecting human speakers’ aggregate referential intentions. Each term is estimated only from tokens that exactly match it.

listener interpretations to align with speaker intentions, and we can gain insight into (aggregate) speaker intentions using our method for grounding referential terms. Whereas the literal interpretation for message  $m$  is obtained from all the tokens that contain it (Fig. 3), the speaker’s *intended interpretation* for  $m$  is obtained from all of the tokens that exactly match it. For instance, the meaning of ‘top’ now excludes tokens like ‘top left’. Fig. 5 shows these denotations, which mirror the distributions predicted by our model (Fig. 4). Thus, the L(S) model correctly infers the pragmatic meaning of referring expressions as used by human speakers, albeit in an idealized manner.

## 5 Future Work

We showed that implicatures arise in cooperative contexts from nested belief models. Our listener-centric implicatures must be combined with rational speaker behavior (Vogel et al., 2013) to produce general dialog agents. The computational complexity of Dec-POMDPs is prohibitive, and our approximations can be problematic for deep belief nesting. Future work will explore sampling-based approaches to belief update and decision making (Doshi and Gmytrasiewicz, 2009) to overcome these problems. These steps will move us closer to a computationally effective, unified theory of pragmatic enrichment and decision making.

**Acknowledgements** This research was supported in part by ONR grants N00014-10-1-0109 and N00014-13-1-0287 and ARO grant W911NF-07-1-0216.

## References

- Leon Bergen, Noah D. Goodman, and Roger Levy. 2012. That’s what she (could have) said: How alternative utterances affect language use. In *Proceedings of the Thirty-Fourth Annual Conference of the Cognitive Science Society*.
- Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840.
- Colin F. Camerer, Teck-Hua Ho, and Juin-Kuan Chong. 2004. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, August.
- Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.
- Eve V. Clark. 1987. The principle of contrast: A constraint on language acquisition. In Brian MacWhinney, editor, *Mechanisms of Language Acquisition*, pages 1–33. Erlbaum, Hillsdale, NJ.
- Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.
- Judith Degen and Michael Franke. 2012. Optimal reasoning about referential expressions. In *Proceedings of SemDIAL 2012*, Paris, September.
- David DeVault and Matthew Stone. 2007. Managing ambiguities across utterances in dialogue. In Ron Artstein and Laure Vieu, editors, *Proceedings of DECALOG 2007: Workshop on the Semantics and Pragmatics of Dialogue*.
- Prashant Doshi and Piotr J. Gmytrasiewicz. 2009. Monte carlo sampling methods for approximating interactive pomdps. *J. Artif. Int. Res.*, 34(1):297–337, March.
- Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998.
- Michael C. Frank, Noah D. Goodman, and Joshua B. Tenenbaum. 2009. Using speakers’ referential intentions to model early cross-situational word learning. *Psychological Science*, 20(5):579–585.
- Michael Franke. 2009. *Signal to Act: Game Theory in Pragmatics*. ILLC Dissertation Series. Institute for Logic, Language and Computation, University of Amsterdam.
- Gerald Gazdar. 1979. *Pragmatics: Implicature, Presupposition and Logical Form*. Academic Press, New York.
- Piotr J. Gmytrasiewicz and Prashant Doshi. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:24–49.
- Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, Cambridge, MA, October. ACL.
- H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Syntax and Semantics*, volume 3: Speech Acts, pages 43–58. Academic Press, New York.
- Robert M. Harnish. 1979. Logical form and implicature. In *Linguistic Communication and Speech Acts*, pages 313–391. MIT Press, Cambridge, MA.
- Julia Hirschberg. 1985. *A Theory of Scalar Implicature*. Ph.D. thesis, University of Pennsylvania.
- Laurence R Horn. 1972. *On the Semantic Properties of Logical Operators in English*. Ph.D. thesis, UCLA, Los Angeles.
- Gerhard Jäger. 2007. Game dynamics connects semantics and pragmatics. In Ahti-Veikko Pietarinen, editor, *Game Theory and Linguistic Meaning*, pages 89–102. Elsevier, Amsterdam.
- Gerhard Jäger. 2012. Game theory in semantics and pragmatics. In Maienborn et al. (Maienborn et al., 2012).
- Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.
- Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors. 2012. *Semantics: An International Handbook of Natural Language Meaning*, volume 3. Mouton de Gruyter, Berlin.
- Christopher Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In Nathan Arnett and Ryan Bennett, editors, *Proceedings of the 30th West Coast Conference on Formal Linguistics*, Somerville, MA. Cascadilla Press.
- Hannah Rohde, Scott Seyfarth, Brady Clark, Gerhard Jäger, and Stefan Kaufmann. 2012. Communicating with cost-based implicature: A game-theoretic approach to ambiguity. In *The 16th Workshop on the Semantics and Pragmatics of Dialogue*, Paris, September.
- Seymour Rosenberg and Bertram D. Cohen. 1964. Speakers’ and listeners’ processes in a word communication task. *Science*, 145:1201–1203.
- Matthijs T. J. Spaan and Nikos Vlassis. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24(1):195–220, August.

Alex Stiller, Noah D. Goodman, and Michael C. Frank. 2011. Ad-hoc scalar implicature in adults and children. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, Boston, July.

Adam Vogel, Max Bodoia, Dan Jurafsky, and Christopher Potts. 2013. Emergence of Gricean maxims from multi-agent decision theory. In *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Atlanta, Georgia, June. Association for Computational Linguistics.



# Domain-Specific Coreference Resolution with Lexicalized Features

Nathan Gilbert and Ellen Riloff

School of Computing

University of Utah

50 S. Central Campus Dr.

Salt Lake City, UT 84112

USA

{ngilbert, riloff}@cs.utah.edu

## Abstract

Most coreference resolvers rely heavily on string matching, syntactic properties, and semantic attributes of words, but they lack the ability to make decisions based on individual words. In this paper, we explore the benefits of lexicalized features in the setting of domain-specific coreference resolution. We show that adding lexicalized features to off-the-shelf coreference resolvers yields significant performance gains on four domain-specific data sets and with two types of coreference resolution architectures.

## 1 Introduction

Coreference resolvers are typically evaluated on collections of news articles that cover a wide range of topics, such as the ACE (ACE03, 2003; ACE04, 2004; ACE05, 2005) and OntoNotes (Pradhan et al., 2007) data sets. Many NLP applications, however, involve text analysis for specialized domains, such as clinical medicine (Gooch and Roudsari, 2012; Glinos, 2011), legal text analysis (Bouayad-Agha et al., 2009), and biological literature (Batista-Navarro and Ananiadou, 2011; Castaño et al., 2002). Learning-based coreference resolvers can be easily retrained for a specialized domain given annotated training texts for that domain. However, we found that retraining an off-the-shelf coreference resolver with domain-specific texts showed little benefit.

This surprising result led us to question the nature of the feature sets used by noun phrase (NP) coreference resolvers. Nearly all of the features employed by recent systems fall into three categories: string match and word overlap, syntactic properties (e.g., appositives, predicate nominals, parse features, etc.), and semantic matching (e.g., gender agreement, WordNet similarity, named entity classes, etc.). Conspicuously absent from most

systems are *lexical features* that allow the classifier to consider the specific words when making a coreference decision. A few researchers have experimented with lexical features, but they achieved mixed results in evaluations on broad-coverage corpora (Bengston and Roth, 2008; Björkelund and Nugues, 2011; Rahman and Ng, 2011a).

We hypothesized that lexicalized features can have a more substantial impact in domain-specific settings. Lexical features can capture domain-specific knowledge and subtle semantic distinctions that may be important within a domain. For example, based on the resolutions found in domain-specific training sets, our lexicalized features captured the knowledge that “tomcat” can be coreferent with “plane”, “UAW” can be coreferent with “union”, and “anthrax” can be coreferent with “diagnosis”. Capturing these types of domain-specific information is often impossible using only general-purpose resources. For example, WordNet defines “tomcat” only as an animal, does not contain an entry for “UAW”, and categorizes “anthrax” and “diagnosis” very differently.<sup>1</sup>

In this paper, we evaluate the impact of lexicalized features on 4 domains: management succession (MUC-6 data), vehicle launches (MUC-7 data), disease outbreaks (ProMed texts), and terrorism (MUC-4 data). We incorporate lexicalized feature sets into two different coreference architectures: Reconcile (Stoyanov et al., 2010), a pairwise coreference classifier, and Sieve (Raghuathan et al., 2010), a rule-based system. Our results show that lexicalized features significantly improve performance in all four domains and in both types of coreference architectures.

## 2 Related Work

We are not the first researchers to use lexicalized features for coreference resolution. However, pre-

<sup>1</sup>WordNet defines “anthrax” as a disease (condition/state) and “diagnosis” as an identification (discovery event).

| Train \ Test | MUC-6        |              |              | MUC-7        |              |              | Promed       |              |              | MUC-4        |              |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|              | P            | R            | F            | P            | R            | F            | P            | R            | F            | P            | R            | F            |
| MUC-6        | <b>80.79</b> | 62.71        | <b>70.61</b> | <b>84.33</b> | 61.74        | 71.29        | 83.54        | 70.34        | 76.37        | <b>80.22</b> | 60.81        | 69.18        |
| MUC-7        | 74.78        | 65.59        | 69.88        | 82.73        | 64.09        | <b>72.23</b> | <b>85.29</b> | 71.82        | <b>77.98</b> | 77.35        | 64.19        | <b>70.16</b> |
| Promed       | 73.60        | 64.20        | <b>68.60</b> | 82.88        | 63.37        | 71.82        | 80.31        | 72.66        | 76.29        | 74.52        | 65.65        | 69.80        |
| MUC-4        | 69.27        | <b>65.66</b> | 67.42        | 71.49        | <b>67.22</b> | 69.29        | 76.92        | <b>74.25</b> | 75.56        | 71.76        | <b>67.37</b> | 69.50        |

Table 1: Cross-domain  $B^3$  (Bagga and Baldwin, 1998) results for Reconcile with its general feature set. The Paired Permutation test (Pesarin, 2001) was used for statistical significance testing and gray cells represent results that are not significantly different from the best result.

vious work has evaluated the benefit of lexical features only for broad-coverage data sets.

Bengston and Roth (2008) incorporated a *memorization* feature to learn which entities can refer to one another. They created a binary feature for every pair of head nouns, including pronouns. They reported no significant improvement from these features on the ACE 2004 data.

Rahman and Ng (2011a) also utilized lexical features, going beyond strict memorization with methods to combat data sparseness and incorporating semantic information. They created a feature for every ordered pair of head nouns (for pronouns and nominals) or full NPs (for proper nouns). *Semi-lexical features* were also used when one NP was a Named Entity, and *unseen features* were used when the NPs were not in the training set. Their features did yield improvements on both the ACE 2005 and OntoNotes-2 data, but the semi-lexical features included Named Entity classes as well as word-based features.

Rahman and Ng (2011b) explored the use of lexical features in greater detail and showed their benefit on the ACE05 corpus independent of, and combined with, a conventional set of coreference features. The ACE05 corpus is drawn from six sources (Newswire, Broadcast News, Broadcast Conversations, Conversational Telephone Speech, Weblogs, and Usenet). The authors experimented with utilizing lexical information drawn from different sources. The results showed that the best performance came from training and testing with lexical knowledge drawn from the same source. Although our approach is similar, this paper focuses on learning lexical information from different *domains* as opposed to the different genres found in the six sources of the ACE05 corpus.

Björkelund and Nugues (2011) used lexical word pairs for the 2011 CoNLL Shared Task, showing significant positive impact on performance. They used over 2000 annotated documents from the broad-coverage OntoNotes corpus

for training. Our work aims to show the benefit of lexical features using much smaller training sets (< 50 documents) focused on specific domains.

Lexical features have also been used for slightly different purposes. Florian et al. (2004) utilized lexical information such as mention spelling and context for entity tracking in ACE. Ng (2007) used lexical information to assess the likelihood of a noun phrase being anaphoric, but this did not show clear improvements on ACE data.

There has been previous work on domain-specific coreference resolution for several domains, including biological literature (Castaño et al., 2002; Liang and Lin, 2005; Gasperin and Briscoe, 2008; Kim et al., 2011; Batista-Navarro and Ananiadou, 2011), clinical medicine (He, 2007; Zheng et al., 2011; Glinos, 2011; Gooch and Roudsari, 2012) and legal documents (Bouayad-Agha et al., 2009). In addition, BABAR (Bean and Riloff, 2004) used *contextual role knowledge* for coreference resolution in the domains of terrorism and natural disasters. But BABAR acquired and used lexical information to match the compatibility of contexts surrounding NPs, not the NPs themselves. To the best of our knowledge, our work is the first to examine the impact of lexicalized features for domain-specific coreference resolution.

### 3 Exploiting Lexicalized Features

Table 1 shows the performance of a learning-based coreference resolver, Reconcile (Stoyanov et al., 2010), with its default feature set using different combinations of training and testing data. Reconcile does not include any lexical features, but does contain over 60 general features covering semantic agreement, syntactic constraints, string match and recency.

Each row represents a training set, each column represents a test set, and each cell shows precision (P), recall (R), and F score results under the  $B^3$  metric when using the corresponding training and test data. The best results for each test set appear

|                  | MUC-6        |              |              | MUC-7        |              |              | ProMED       |              |              | MUC-4        |              |              |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                  | P            | R            | F            | P            | R            | F            | P            | R            | F            | P            | R            | F            |
| <b>Reconcile</b> | 80.79        | 62.71        | 70.61        | 82.73        | 64.09        | 72.23        | 80.31        | 72.66        | 76.29        | 71.76        | 67.37        | 69.50        |
| +LexLookup       | <b>87.01</b> | 63.40        | 73.35        | <b>87.39</b> | 62.86        | 73.12        | <b>86.66</b> | 70.95        | 78.02        | <b>82.89</b> | 67.53        | <b>74.42</b> |
| +LexSets         | 86.50        | <b>63.76</b> | <b>73.41</b> | 85.86        | <b>64.35</b> | <b>73.56</b> | 86.19        | <b>72.14</b> | <b>78.54</b> | 81.98        | <b>67.73</b> | 74.18        |
| <b>Sieve</b>     | <b>92.20</b> | 61.70        | 73.90        | <b>91.46</b> | 59.59        | 72.16        | <b>94.43</b> | 67.25        | 78.55        | <b>91.30</b> | 59.84        | 72.30        |
| +LexBegin        | 91.22        | 62.97        | 74.51        | 91.24        | 60.28        | 72.59        | 93.51        | <b>69.15</b> | <b>79.51</b> | 89.01        | 62.84        | 73.67        |
| +LexEnd          | 90.59        | <b>63.47</b> | <b>74.64</b> | 91.17        | <b>60.56</b> | <b>72.78</b> | 93.99        | 68.87        | 79.49        | 89.04        | <b>64.03</b> | <b>74.47</b> |

Table 2: B<sup>3</sup> results for baselines and lexicalized feature sets across four domains.

in **boldface**.

We performed statistical significance testing using the Paired Permutation test (Pesarin, 2001) and the gray cells represent results where there was not significant difference from the best results in the same column. If just one cell is gray in a column, that indicates the result was significantly better than the other results in the same column with  $p \leq 0.05$ .

Table 1 does not show much benefit from training on the same domain as the test set. Three different training sets produce F scores that are not significantly different for both the MUC-6 and MUC-4 test data. For ProMed, training on the MUC-7 data yields significantly better results than training on all the other data sets, including ProMed texts! Based on these results, it would seem that training on the MUC-7 texts is likely to yield the best results no matter what domain you plan to use the coreference resolver for. The goal of our work is to investigate whether lexical features can extract additional knowledge from domain-specific training texts to help tailor a coreference resolver to perform better for a specific domain.

### 3.1 Extracting Coreferent Training Pairs

We adopt the terminology introduced by Stoyanov et al. (2009) to define a coreference element (CE) as a noun phrase that can participate in a coreference relation based on the task definition.

Each training document has manually annotated gold coreference chains corresponding to the sets of CEs that are coreferent. For each CE in a gold chain, we pair that CE with all of the other CEs in the same chain. We consider the coreference relation to be bi-directional, so we don't retain information about which CE was the antecedent. We do not extract CE pairs that share the same head noun because they are better handled with string match. For nominal NPs, we retain only the head noun, but we use the entire NP for proper names. We discard pairs that include a pronoun, and nor-

malize strings to lower case for consistency.

### 3.2 Lexicalized Feature Sets

We explore two ways to capture lexicalized information as features. The first approach indicates whether two CEs have ever been coreferent in the training data. We create a single feature called  $\text{LEXLOOKUP}(x, y)$  that receives a value of 1 when  $x$  and  $y$  have been coreferent at least twice, or a value of 0 otherwise.<sup>2</sup>  $\text{LEXLOOKUP}(x, y)$  is a single feature that captures all CE pairs that were coreferent in the training data.

We also created *set-based* features that capture the set of terms that have been coreferent with a particular CE. The  $\text{CorefSet}(x)$  is the set of CEs that have appeared in the same coreference chain as mention  $x$  at least twice.

We create a set of binary-valued features  $\text{LEXSET}(x, y)$ , one for each CE  $x$  in the training data. Given a pair of CEs,  $x$  and  $y$ ,  $\text{LEXSET}(x, y) = 1$  if  $y \in \text{CorefSet}(x)$ , or 0 otherwise. The benefit of the set-based features over a single monolithic feature is that the classifier has one set-based feature for each mention found in the training data, so it can learn to handle individual terms differently.

We also tried encoding a separate feature for each distinct pair of words, analogous to the memorization feature in Bengston and Roth (2008). This did not improve performance as much as the other feature representations presented here.

## 4 Evaluation

### 4.1 Data Sets

We evaluated the performance of lexicalized features on 4 domain-specific corpora including two standard coreference benchmarks, the MUC-6 and MUC-7 data sets. The MUC-6 domain is management succession and consists of 30 training texts and 30 test texts. The MUC-7 domain is vehicle

<sup>2</sup>We require a frequency  $\geq 2$  to minimize overfitting because many cases occur only once in the training data.

launches and consists of 30 training texts and 20 test texts. We used these standard train/test splits to be consistent with previous work.

We also created 2 new coreference data sets which we will make freely available. We manually annotated 45 ProMed-mail articles ([www.promedmail.org](http://www.promedmail.org)) about disease outbreaks and 45 MUC-4 texts about terrorism, following the MUC guidelines (Hirschman, 1997). Inter-annotator agreement between two annotators was .77 ( $\kappa$ ) on ProMed and .84 (MUC F Score) (Villain et al., 1995) on both ProMed and MUC-4.<sup>3</sup> We performed 5-fold cross-validation on both data sets and report the micro-averaged results.

Gold CE spans were used in all experiments to factor out issues with markable identification and anaphoricity across the different domains.

## 4.2 Coreference Resolution Models

We conducted experiments using two coreference resolution architectures. Reconcile<sup>4</sup> (Stoyanov et al., 2010) is a freely available pairwise mention classifier. For classification, we chose Weka’s (Witten and Frank, 2005) Decision Tree learner inside Reconcile. Reconcile contains roughly 60 features (none lexical), largely modeled after Ng and Cardie (2002). We modified Reconcile’s Single Link clustering scheme to enforce an additional rule that non-overlapping proper names cannot be merged into the same chain.

We also conducted experiments with the Sieve coreference resolver, which applies high precision heuristic rules to incrementally build coreference chains. We implemented the LEXLOOKUP( $X, Y$ ) feature as an additional heuristic rule. We tried inserting this heuristic before Sieve’s other rules (LexBegin), and also after Sieve’s other rules (LexEnd).

## 4.3 Experimental Results

Table 2 presents results for Reconcile trained with and without lexical features and when adding a lexical heuristic with data drawn from same-domain texts to Sieve.

The first row shows the results without the lexicalized features (from Table 1). All F scores for Reconcile with lexicalized features are significantly better than without these features based on the Paired Permutation test (Pesarin, 2001) with

<sup>3</sup>We also computed  $\kappa$  on MUC-4, but unfortunately the score and original data were lost.

<sup>4</sup><http://www.cs.utah.edu/nlp/reconcile/>

$p \leq 0.05$ . MUC-4 showed the largest gain for Reconcile, with the F score increasing from 69.5 to over 74. For most domains, adding the lexical features to Reconcile substantially increased precision with comparable levels of recall.

The bottom half of Table 2 contains the results of adding a lexical heuristic to Sieve. The first row shows the default system with no lexical information. All F scores with the lexical heuristic are significantly better than without it. In Sieve’s high-precision coreference architecture, the lexical heuristic yields additional recall gains without sacrificing much precision.

|                  | ACE 2004     |              |              |
|------------------|--------------|--------------|--------------|
|                  | P            | R            | F            |
| <b>Reconcile</b> | 70.59        | 83.09        | 76.33        |
| +LexLookup       | 71.32        | 82.93        | 76.69        |
| +LexSets         | <b>71.44</b> | <b>83.45</b> | <b>76.98</b> |
| <b>Sieve</b>     | <b>90.09</b> | 74.23        | <b>81.39</b> |
| +LexBegin        | 86.54        | 75.43        | 80.61        |
| +LexEnd          | 87.00        | <b>75.45</b> | 80.82        |

Table 3: B<sup>3</sup> results for baselines and lexicalized feature sets on the broad-coverage ACE 2004 data set.

Table 3 shows the results for Reconcile and Sieve when training and testing on the ACE 2004 data. Here, we see little improvement from adding lexical information. For Reconcile, the small differences in F scores are not statistically significant. For Sieve, the unlexicalized system yields a significantly higher F score than when adding the lexical heuristic. These results support our hypothesis that lexicalized information can be beneficial for capturing domain-specific word associations, but may not be as helpful in a broad-coverage setting where the language covers a diverse set of topics.

Table 4 shows a re-evaluation of the cross-domain experiments from Table 1 for Reconcile with the LexSet features added. The bottom half of the table shows cross-domain experiments for Sieve using the lexical heuristic at the end of its rule set (LexEnd). Results are presented using both the B<sup>3</sup> metric and the MUC Score (Villain et al., 1995).

Training and testing on the same domain always produced the highest recall scores for MUC-7, ProMed, and MUC-4 when utilizing lexical features. In all cases, lexical features acquired from same-domain texts yield results that are either clearly the best or not significantly different from the best.

| Train \ Test                           | MUC-6        |              |              | MUC-7        |              |              | Promed       |              |              | MUC-4        |              |              |
|--|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|  | P            | R            | F            | P            | R            | F            | P            | R            | F            | P            | R            | F            |
| <b>Reconcile (B<sup>3</sup> Score)</b> |              |              |              |              |              |              |              |              |              |              |              |              |
| MUC-6                                  | <b>86.50</b> | <b>63.76</b> | <b>73.41</b> | <b>90.44</b> | 60.75        | <b>72.68</b> | 89.28        | 68.14        | 77.29        | 84.05        | 60.61        | 70.44        |
| MUC-7                                  | 80.65        | 63.42        | 71.01        | 85.86        | <b>64.46</b> | <b>73.56</b> | <b>89.41</b> | 70.05        | <b>78.55</b> | 80.61        | 63.26        | 70.89        |
| Promed                                 | 81.69        | 62.73        | 70.96        | 88.32        | 62.79        | 73.40        | 86.19        | <b>72.14</b> | 78.54        | <b>84.81</b> | 62.58        | 72.02        |
| MUC-4                                  | 81.20        | 62.34        | 70.53        | 87.23        | 63.13        | 73.25        | 87.52        | 71.11        | 78.46        | 81.98        | <b>67.73</b> | <b>74.18</b> |
| <b>Reconcile (MUC Score)</b>           |              |              |              |              |              |              |              |              |              |              |              |              |
| MUC-6                                  | <b>89.56</b> | 71.17        | <b>79.32</b> | 90.85        | 67.43        | 77.41        | 89.61        | 65.67        | 75.79        | 88.27        | 66.98        | 76.16        |
| MUC-7                                  | 86.14        | <b>72.22</b> | 78.57        | 89.56        | <b>72.01</b> | <b>79.83</b> | <b>89.34</b> | 68.08        | 77.27        | 87.30        | 70.22        | 77.83        |
| Promed                                 | 86.92        | 70.68        | 77.97        | <b>90.93</b> | 70.33        | 79.31        | 88.54        | <b>69.55</b> | <b>77.90</b> | <b>88.83</b> | 68.89        | 78.23        |
| MUC-4                                  | 85.72        | 70.50        | 77.37        | 88.78        | 71.24        | 79.05        | 88.24        | 68.18        | 77.55        | 87.89        | <b>74.18</b> | <b>80.45</b> |
| <b>Sieve (B<sup>3</sup> Score)</b>     |              |              |              |              |              |              |              |              |              |              |              |              |
| MUC-6                                  | 90.59        | 63.47        | 74.64        | 91.20        | 59.91        | 72.32        | 94.30        | 67.25        | 78.51        | <b>91.30</b> | 59.90        | 72.34        |
| MUC-7                                  | 91.62        | <b>63.67</b> | <b>75.13</b> | 91.17        | <b>60.56</b> | <b>72.78</b> | <b>94.43</b> | 67.35        | 78.62        | 91.14        | 60.44        | 72.68        |
| Promed                                 | <b>92.14</b> | 61.70        | 73.90        | <b>91.46</b> | 59.93        | 72.41        | 93.99        | <b>68.87</b> | <b>79.49</b> | 91.27        | 60.76        | 72.96        |
| MUC-4                                  | 91.76        | 61.88        | 73.91        | 91.26        | 59.93        | 72.34        | 94.30        | 67.35        | 78.58        | 89.04        | <b>64.03</b> | <b>74.47</b> |
| <b>Sieve (MUC Score)</b>               |              |              |              |              |              |              |              |              |              |              |              |              |
| MUC-6                                  | 91.80        | <b>70.87</b> | <b>79.99</b> | 91.38        | 65.52        | 76.32        | 92.08        | 64.71        | 76.01        | 90.38        | 66.98        | 77.10        |
| MUC-7                                  | <b>91.82</b> | 69.70        | 79.25        | <b>91.68</b> | <b>66.36</b> | <b>76.99</b> | <b>92.20</b> | 64.86        | 76.15        | 90.71        | 67.09        | 77.13        |
| Promed                                 | 91.99        | 69.15        | 78.95        | <b>91.68</b> | 65.52        | 76.42        | 91.70        | <b>66.33</b> | <b>76.98</b> | <b>90.85</b> | 67.09        | 77.18        |
| MUC-4                                  | 91.79        | 69.39        | 79.03        | 91.48        | 65.52        | 76.36        | 92.00        | 64.86        | 76.08        | 90.31        | <b>69.62</b> | <b>78.62</b> |

Table 4: Cross-domain B<sup>3</sup> and MUC results for Reconcile and Sieve with lexical features. Gray cells represent results that are not significantly different from the best results in the column at the 0.05 p-level.

For MUC-6 and MUC-7, the highest F score results almost always come from training on same-domain texts, although in some cases these results are not significantly different from training on other domains. Lexical features can yield improvements when training on a different domain if there is overlap in the vocabulary across the domains. For the ProMed domain, the Sieve system performs significantly better, under both metrics, with same-domain lexical features than with lexical features acquired from a different domain. For Reconcile, there is not a significant difference in the F score for ProMed when training on ProMed, MUC-4, or MUC-7. In the MUC-4 domain, using same-domain lexical information *always* produces the best F score, under both metrics and in both coreference systems.

## 5 Conclusions

We explored the use of lexical information for domain-specific coreference resolution using 4 domain-specific data sets and 2 coreference resolvers. Lexicalized features consistently improved performance for all of the domains and in both coreference architectures. We see benefits from lexicalized features in cross-domain training, but the gains are often more substantial when utilizing same-domain lexical knowledge.

In the future, we plan to explore additional types of lexical information to benefit domain-specific coreference resolution.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1018314 and the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the U.S. government.

## References

- ACE03. 2003. NIST ACE evaluation website. In <http://www.nist.gov/speech/tests/ace/2003>.
- ACE04. 2004. NIST ACE evaluation website. In <http://www.nist.gov/speech/tests/ace/2004>.
- ACE05. 2005. NIST ACE evaluation website. In <http://www.nist.gov/speech/tests/ace/2005>.
- Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreference using the Vector Space Model. *Proceedings of the 17th international conference on Computational Linguistics (COLING)*.
- Riza Theresa Batista-Navarro and Sophia Ananiadou. 2011. Building a coreference-annotated corpus from the domain of biochemistry. In *Proceedings of BioNLP 2011 Workshop*, BioNLP '11, pages 83–91.
- David Bean and Ellen Riloff. 2004. Unsupervised learning of Contextual Role Knowledge for coreference resolution. *Proceedings of the HLT/NAACL 2004*.

- Eric Bengston and Dan Roth. 2008. Understanding the value of features for coreference resolution. *Empirical Methods in Natural Language Processing*.
- Anders Björkelund and Pierre Nugues. 2011. Exploring lexicalized features for coreference resolution. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 45–50.
- Nadjet Bouayad-Agha, Gerard Casamayor, Gabriela Ferraro, Simon Mille, Vanesa Vidal, and Leo Wanner. 2009. Improving the comprehension of legal documentation: the case of patent claims. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 78–87.
- José Castaño, Jason Zhang, and James Pustejovsky. 2002. Anaphora resolution in biomedical literature. *International Symposium on Reference Resolution*.
- Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, Salim Roukos, and T Zhang. 2004. A statistical model for multilingual entity detection and tracking. *HLT-NAACL*.
- Caroline Gasperin and Ted Briscoe. 2008. Statistical anaphora resolution in biomedical texts. *Proceedings of the 22nd Annual Conference on Computational Linguistics*, pages 257–264.
- Demetrios G. Glinos. 2011. A search based method for clinical text coreference resolution. In *Proceedings of the Fifth i2b2/VA Track on Challenges in Natural Language Processing for Clinical Data (i2b2 2011)*.
- Phil Gooch and Abdul Roudsari. 2012. Lexical patterns, features and knowledge resources for coreference resolution in clinical notes. *Journal of Biomedical Informatics*, 45.
- Tian Ye He. 2007. *Coreference resolution on entities and events for hospital discharge summaries*. Ph.D. thesis, Massachusetts Institute of Technology.
- Lynette Hirschman. 1997. MUC-7 task definition. *Proceedings of MUC-7*.
- Youngjun Kim, Ellen Riloff, and Nathan Gilbert. 2011. The taming of Reconcile as a Biomedical coreference resolver. *ACL/HLT 2011 Workshop on Biomedical Natural Language Processing (BioNLP 2011) Shared Task Paper*.
- Tyne Liang and Yu-Hsiang Lin. 2005. Anaphora resolution for biomedical literature by exploiting multiple resources. *Natural Language Processing-IJCNLP 2005*, pages 742–753.
- Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. *Proceedings of the 40th Annual Meeting of the ACL*, pages 104–111.
- Vincent Ng. 2007. Shallow semantics for coreference resolution. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1689–1694.
- Fortunato Pesarin. 2001. *Multivariate permutation tests: with applications in biostatistics*, volume 240. Wiley Chichester.
- Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessice MacBride, and Linnea Micculla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. In *Proceedings of the International Conference on Semantic Computing*.
- Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A Multi-Pass Sieve for coreference resolution. *Empirical Methods in Natural Language Processing 2010*.
- Altaf Rahman and Vincent Ng. 2011a. Coreference resolution with world knowledge. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies (ACL-HLT)*, pages 814–824.
- Altaf Rahman and Vincent Ng. 2011b. Narrowing the modelling gap: A cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research*.
- Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the State-of-the-Art. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP (ACL-IJCNLP 2009)*.
- Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2010. Coreference resolution with Reconcile. *Proceedings of the Joint Conference of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*.
- Marc Villain, John Aberdeen, John Berger, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of the 6th conference on Message understanding*.
- Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition.
- Jiaping Zheng, Wendy Chapman, Rebecca Crowley, and Guergana Savova. 2011. Coreference resolution: A review of general methodologies and applications in the clinical domain. *Journal of Biomedical Informatics*, 44:1113–1122.

# Learning to Order Natural Language Texts

Jiwei Tan<sup>a, b</sup>, Xiaojun Wan<sup>a\*</sup> and Jianguo Xiao<sup>a</sup>

<sup>a</sup>Institute of Computer Science and Technology, The MOE Key Laboratory of Computational Linguistics, Peking University, China

<sup>b</sup>School of Information Science and Technology, Beijing Normal University, China

tanjiwei8@gmail.com, {wanxiaojun, jgxiao}@pku.edu.cn

## Abstract

Ordering texts is an important task for many NLP applications. Most previous works on summary sentence ordering rely on the contextual information (e.g. adjacent sentences) of each sentence in the source document. In this paper, we investigate a more challenging task of ordering a set of unordered sentences without any contextual information. We introduce a set of features to characterize the order and coherence of natural language texts, and use the learning to rank technique to determine the order of any two sentences. We also propose to use the genetic algorithm to determine the total order of all sentences. Evaluation results on a news corpus show the effectiveness of our proposed method.

## 1 Introduction

Ordering texts is an important task in many natural language processing (NLP) applications. It is typically applicable in the text generation field, both for concept-to-text generation and text-to-text generation (Lapata, 2003), such as multiple document summarization (MDS), question answering and so on. However, ordering a set of sentences into a coherent text is still a hard and challenging problem for computers.

Previous works on sentence ordering mainly focus on the MDS task (Barzilay et al., 2002; Okazaki et al., 2004; Nie et al., 2006; Ji and Pulman, 2006; Madnani et al., 2007; Zhang et al., 2010; He et al., 2006; Bollegala et al., 2005; Bollegala et al., 2010). In this task, each summary sentence is extracted from a source document. The timestamp of the source documents and the adjacent sentences in the source documents can be used as important clues for ordering summary sentences.

In this study, we investigate a more challenging and more general task of ordering a set of unordered sentences (e.g. randomly shuffle the

sentences in a text paragraph) without any contextual information. This task can be applied to almost all text generation applications without restriction.

In order to address this challenging task, we first introduce a few useful features to characterize the order and coherence of natural language texts, and then propose to use the learning to rank algorithm to determine the order of two sentences. Moreover, we propose to use the genetic algorithm to decide the overall text order. Evaluations are conducted on a news corpus, and the results show the prominence of our method. Each component technique or feature in our method has also been validated.

## 2 Related Work

For works taking no use of source document, Lapata (2003) proposed a probabilistic model which learns constraints on sentence ordering from a corpus of texts. Experimental evaluation indicated the importance of several learned lexical and syntactic features. However, the model only works well when using single feature, but unfortunately, it becomes worse when multiple features are combined. Barzilay and Lee (2004) investigated the utility of domain-specific content model for representing topic and topic shifts and the model performed well on the five selected domains. Nahnsen (2009) employed features which were based on discourse entities, shallow syntactic analysis, and temporal precedence relations retrieved from VerbOcean. However, the model does not perform well on datasets describing the consequences of events.

## 3 Our Proposed Method

### 3.1 Overview

The task of text ordering can be modeled like (Cohen et al., 1998), as measuring the coherence of a text by summing the association strength of any sentence pairs. Then the objective of a text ordering model is to find a permutation which can maximize the summation.

---

\* Xiaojun Wan is the corresponding author.

Formally, we define an association strength function  $\text{PREF}(u,v) \in \mathbb{R}$  to measure how strong it is that sentence  $u$  should be arranged before sentence  $v$  (denoted as  $u \succ v$ ). We then define function  $\text{AGREE}(\rho, \text{PREF})$  as:

$$\text{AGREE}(\rho, \text{PREF}) = \sum_{u,v:\rho(u) > \rho(v)} \text{PREF}(u,v) \quad (1)$$

where  $\rho$  denotes a sentence permutation and  $\rho(u) > \rho(v)$  means  $u \succ v$  in the permutation  $\rho$ . Then the objective of finding an overall order of the sentences becomes finding a permutation  $\rho$  to maximize  $\text{AGREE}(\rho, \text{PREF})$ .

The main framework is made up of two parts: defining a pairwise order relation and determining an overall order. Our study focuses on both the two parts by learning a better pairwise relation and proposing a better search strategy, as described respectively in next sections.

### 3.2 Pairwise Relation Learning

The goal for pairwise relation learning is defining the strength function  $\text{PREF}$  for any sentence pair. In our method we define the function  $\text{PREF}$  by combining multiple features.

**Method:** Traditionally, there are two main methods for defining a strength function: integrating features by a linear combination (He et al., 2006; Bollegala et al., 2005) or by a binary classifier (Bollegala et al., 2010). However, the binary classification method is very coarse-grained since it considers any pair of sentences either “positive” or “negative”. Instead we propose to use a better model of learning to rank to integrate multiple features.

In this study, we use Ranking SVM implemented in the  $\text{svm}^{\text{rank}}$  toolkit (Joachims, 2002; Joachims, 2006) as the ranking model. The examples to be ranked in our ranking model are sequential sentence pairs like  $u \succ v$ . The feature values for a training example are generated by a few feature functions  $f_i(u,v)$ , and we will introduce the features later. We build the training examples for  $\text{svm}^{\text{rank}}$  as follows:

For a training query, which is a paragraph with  $n$  sequential sentences as  $s_1 \succ s_2 \succ \dots \succ s_n$ , we can get  $A_n^2 = n(n-1)$  training examples. For pairs like  $s_a \succ s_{a+k}$  ( $k > 0$ ) the target rank values are set to  $n-k$ , which means that the longer the distance between the two sentences is, the smaller the target value is. Other pairs like  $s_{a+k} \succ s_a$  are all set to 0. In order to better capture the order information of each feature, for every sen-

tence pair  $u \succ v$ , we derive four feature values from each function  $f_i(u,v)$ , which are listed as follows:

$$V_{i,1} = f_i(u,v) \quad (2)$$

$$V_{i,2} = \begin{cases} 1/2, & \text{if } f_i(u,v) + f_i(v,u) = 0 \\ \frac{f_i(u,v)}{f_i(u,v) + f_i(v,u)}, & \text{otherwise} \end{cases} \quad (3)$$

$$V_{i,3} = \begin{cases} 1/|S|, & \text{if } \sum_{y \in S \cap y \neq u} f_i(u,y) = 0 \\ f_i(u,v) / \sum_{y \in S \cap y \neq u} f_i(u,y), & \text{otherwise} \end{cases} \quad (4)$$

$$V_{i,4} = \begin{cases} 1/|S|, & \text{if } \sum_{x \in S \cap x \neq v} f_i(x,v) = 0 \\ f_i(u,v) / \sum_{x \in S \cap x \neq v} f_i(x,v), & \text{otherwise} \end{cases} \quad (5)$$

where  $S$  is the set of all sentences in a paragraph and  $|S|$  is the number of sentences in  $S$ . The three additional feature values of (3) (4) (5) are defined to measure the priority of  $u \succ v$  to  $v \succ u$ ,  $u \succ v$  to  $u \succ \forall y \in S - \{u,v\}$  and  $u \succ v$  to  $\forall x \in S - \{u,v\} \succ v$  respectively, by calculating the proportion of  $f_i(u,v)$  in respective summations.

The learned model can be used to predict target values for new examples. A paragraph of unordered sentences is viewed as a test query, and the predicted target value for  $u \succ v$  is set as  $\text{PREF}(u,v)$ .

**Features:** We select four types of features to characterize text coherence. Every type of features is quantified with several functions distinguished by  $i$  in the formulation of  $f_i(u,v)$  and normalized to  $[0,1]$ . The features and definitions of  $f_i(u,v)$  are introduced in Table 1.

| Type              | Description  |
|-------------------|--|
| Similarity        | $\text{sim}(u,v)$  |
|                   | $\text{sim}(\text{latter}(u), \text{former}(v))$                                     |
| Overlap           | $\text{overlap}_j(u,v) / \min( u ,  v )$   |
|                   | $\frac{\text{overlap}_j(\text{latter}(u), \text{former}(v))}{\text{overlap}_j(u,v)}$ |
| Coreference       | Number of coreference chains   |
|                   | Number of coreference words  |
| Probability Model | Noun   |
|                   | Verb   |
|                   | Verb & noun dependency   |
|                   | Adjective & adverb   |

Table 1: Features used in our model.



As in Table 1, function  $\text{sim}(u, v)$  denotes the cosine similarity of sentence  $u$  and  $v$ ;  $\text{latter}(u)$  and  $\text{former}(v)$  denotes the latter half part of  $u$  and the former part of  $v$  respectively, which are separated by the most centered comma (if exists) or word (if no comma exists);  $\text{overlap}_j(u, v)$  denotes the number of mutual words of  $u$  and  $v$ , for  $j=1, 2, 3$  representing lemmatized noun, verb and adjective or adverb respectively;  $|u|$  is the number of words of sentence  $u$ . The value will be set to 0 if the denominator is 0.

For the coreference features we use the ARK-ref<sup>1</sup> tool. It can output the coreference chains containing words which represent the same entity for two sequential sentences  $u \succ v$ .

The probability model originates from (Lapata, 2003), and we implement the model with four features of lemmatized noun, verb, adjective or adverb, and verb and noun related dependency.

### 3.3 Overall Order Determination

Cohen et al. (1998) proved finding a permutation  $\rho$  to maximize  $\text{AGREE}(\rho, \text{PREF})$  is NP-complete. To solve this, they proposed a greedy algorithm for finding an approximately optimal order. Most later works adopted the greedy search strategy to determine the overall order.

However, a greedy algorithm does not always lead to satisfactory results, as our experiment shows in Section 4.2. Therefore, we propose to use the genetic algorithm (Holland, 1992) as the search strategy, which can lead to better results.

**Genetic Algorithm:** The genetic algorithm (GA) is an artificial intelligence algorithm for optimization and search problems. The key point of using GA is modeling the individual, fitness function and three operators of crossover, mutation and selection. Once a problem is modeled, the algorithm can be constructed conventionally.

In our method we set a permutation  $\rho$  as an individual encoded by a numerical path, for example a permutation  $s_2 \succ s_1 \succ s_3$  is encoded as (2 1 3). Then the function  $\text{AGREE}(\rho, \text{PREF})$  is just the fitness function. We adopt the order-based crossover operator which is described in (Davis, 1985). The mutation operator is a random inversion of two sentences. For selection operator we take a tournament selection operator which randomly selects two individuals to choose the one with the greater fitness value  $\text{AGREE}(\rho, \text{PREF})$ .

After several generations of evolution, the individual with the greatest fitness value will be a close solution to the optimal result.

## 4 Experiments

### 4.1 Experiment Setup

**Data Set and Evaluation Metric:** We conducted the experiments on the North American News Text Corpus<sup>2</sup>. We trained the model on 80 thousand paragraphs and tested with 200 shuffled paragraphs. We use Kendall’s  $\tau$  as the evaluation metric, which is based on the number of inversions in the rankings.

**Comparisons:** It is incomparable with other methods for summary sentence ordering based on special summarization corpus, so we implemented Lapata’s probability model for comparison, which is considered the state of the art for this task. In addition, we implemented a random ordering as a baseline. We also tried to use a classification model in place of the ranking model. In the classification model, sentence pairs like  $s_a \succ s_{a+1}$  were viewed as positive examples and all other pairs were viewed as negative examples. When deciding the overall order for either ranking or classification model we used three search strategies: greedy, genetic and exhaustive (or brutal) algorithms. In addition, we conducted a series of experiments to evaluate the effect of each feature. For each feature, we tested in two experiments, one of which only contained the single feature and the other one contained all the other features. For comparative analysis of features, we tested with an exhaustive search algorithm to determine the overall order.

### 4.2 Experiment Results

The comparison results in Table 2 show that our Ranking SVM based method improves the performance over the baselines and the classification based method with any of the search algorithms. We can also see the greedy search strategy does not perform well and the genetic algorithm can provide a good approximate solution to obtain optimal results.

| Method         | Greedy  | Exhaustive | Genetic |
|----------------|---------|------------|---------|
| Baseline       | -0.0127 |            |         |
| Probability    | 0.1859  |            |         |
| Classification | 0.5006  | 0.5360     | 0.5264  |
| Ranking        | 0.5191  | 0.5768     | 0.5747  |

Table 2: Average  $\tau$  of different methods.

<sup>1</sup> <http://www.ark.cs.cmu.edu/ARKref/>

<sup>2</sup> The corpus is available from <http://www ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC98T30>

**Ranking vs. Classification:** It is not surprising that the ranking model is better, because when using a classification model, an example should be labeled either positive or negative. It is not very reasonable to label a sentence pair like  $s_a \succ s_{a+k}$  ( $k > 1$ ) as a negative example, nor a positive one, because in some cases, it is easy to conclude one sentence should be arranged after another but hard to decide whether they should be adjacent. As we see in the function AGREE, the value of  $\text{PREF}(s_a, s_{a+k})$  also contributes to the summation. In a ranking model, this information can be quantified by the different priorities of sentence pairs with different distances.

**Single Feature Effect:** The effects of different types of features are shown in Table 3. *Prob* denotes Lapata’s probability model with different features.

| Feature                                    | Only   | Removed |
|--|--------|---------|
| <i>Similarity</i>                          | 0.0721 | 0.4614  |
| <i>Overlap</i>                             | 0.1284 | 0.4631  |
| <i>Coreference</i>                         | 0.0734 | 0.4704  |
| <i>Prob<sub>noun</sub></i>                 | 0.3679 | 0.3932  |
| <i>Prob<sub>verb</sub></i>                 | 0.0615 | 0.4544  |
| <i>Prob<sub>adjective&amp;adverb</sub></i> | 0.2650 | 0.4258  |
| <i>Prob<sub>dependency</sub></i>           | 0.2687 | 0.4892  |
| <i>All</i>                                 | 0.5768 |         |

Table 3: Effects of different features.

It can be seen in Table 3 that all these features contribute to the final result. The two features of noun probability and dependency probability play an important role as demonstrated in (Lapata, 2003). Other features also improve the final performance. A paragraph which is ordered entirely right by our method is shown in Figure 1.

- (1) *Vanunu, 43, is serving an 18-year sentence for treason.*
- (2) *He was kidnapped by Israel's Mossad spy agency in Rome in 1986 after giving The Sunday Times of London photographs of the inside of the Dimona reactor.*
- (3) *From the photographs, experts determined that Israel had the world's sixth largest stockpile of nuclear weapons.*
- (4) *Israel has never confirmed or denied that it has a nuclear capability.*

Figure 1: A right ordered paragraph.

Sentences which should be arranged together tend to have a higher similarity and overlap. Like sentence (3) and (4) in Figure 1, they have a highest cosine similarity of 0.2240 and most overlap words of “Israel” and “nuclear”. However, the similarity or overlap of the two sen-

tences does not help to decide which sentence should be arranged before another. In this case the overlap and similarity of half part of the sentences may help. For example latter((3)) and former((4)) share an overlap of “Israel” while there is no overlap for latter((4)) and former((3)).

Coreference is also an important clue for ordering natural language texts. When we use a pronoun to represent an entity, it always has occurred before. For example when conducting coreference resolution for (1)  $\succ$  (2), it will be found that “He” refers to “Vanunu”. Otherwise for (2)  $\succ$  (1), no coreference chain will be found.

### 4.3 Genetic Algorithm

There are three main parameters for GA including the crossover probability (PC), the mutation probability (PM) and the population size (PS). There is no definite selection for these parameters. In our study we experimented with a wide range of parameter values to see the effect of each parameter. It is hard to traverse all possible combinations so when testing a parameter we fixed the other two parameters. The results are shown in Table 4.

| Value Para | Avg    | Max    | Min    | Stddev |
|------------|--------|--------|--------|--------|
| <b>PS</b>  | 0.5731 | 0.5859 | 0.5606 | 0.0046 |
| <b>PC</b>  | 0.5733 | 0.5806 | 0.5605 | 0.0038 |
| <b>PM</b>  | 0.5741 | 0.5803 | 0.5337 | 0.0045 |

Table 4: Results of GA with different parameters.

As we can see in Table 4, when adjusting the three parameters the average  $\tau$  values are all close to the exhaustive result of 0.5768 and their standard deviations are low. Table 4 shows that in our case the genetic algorithm is not very sensible to the parameters. In the experiments, we set PS to 30, PC to 0.5 and PM to 0.05, and reached a value of 0.5747, which is very close to the theoretical upper bound of 0.5768.

## 5 Conclusion and Discussion

In this paper we propose a method for ordering sentences which have no contextual information by making use of Ranking SVM and the genetic algorithm. Evaluation results demonstrate the good effectiveness of our method.

In future work, we will explore more features such as semantic features to further improve the performance.

### Acknowledgments

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03) and National High-Tech R&D Program (2012AA011101).

## References

- Danushka Bollegala, Naoaki Okazaki, Mitsuru Ishizuka. 2005. A machine learning approach to sentence ordering for multi-document summarization and its evaluation. In *Proceedings of the Second international joint conference on Natural Language Processing (IJCNLP '05)*, 624-635.
- Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2010. A bottom-up approach to sentence ordering for multi-document summarization. *Inf. Process. Manage.* 46, 1 (January 2010), 89-109.
- John H. Holland. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
- Lawrence Davis. 1985. Applying adaptive algorithms to epistatic domains. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 1 (IJCAI'85)*, Aravind Joshi (Ed.), Vol. 1. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 162-164.
- Mirella Lapata. 2003. Probabilistic text structuring: experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL '03)*, Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 545-552.
- Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. 2004. Improving chronological sentence ordering by precedence relation. In *Proceedings of the 20th international conference on Computational Linguistics (COLING '04)*. Association for Computational Linguistics, Stroudsburg, PA, USA, , Article 750 .
- Nitin Madnani, Rebecca Passonneau, Necip Fazil Ayan, John M. Conroy, Bonnie J. Dorr, Judith L. Klavans, Dianne P. O'Leary, and Judith D. Schlesinger. 2007. Measuring variability in sentence ordering for news summarization. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG '07)*, Stephan Busemann (Ed.). Association for Computational Linguistics, Stroudsburg, PA, USA, 81-88.
- Paul D. Ji and Stephen Pulman. 2006. Sentence ordering with manifold-based classification in multi-document summarization. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 526-533.
- Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.
- Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL2004: Proceedings of the Main Conference*, pages 113–120.
- Renxian Zhang, Wenjie Li, and Qin Lu. 2010. Sentence ordering with event-enriched semantics and two-layered clustering for multi-document news summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1489-1497.
- Thade Nahnsen. 2009. Domain-independent shallow sentence ordering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium (SRWS '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 78-83.
- Thorsten Joachims. 2002. Optimizing search engines using click through data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*. ACM, New York, NY, USA, 133-142.
- 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)*. ACM, New York, NY, USA, 217-226.
- William W. Cohen, Robert E. Schapire, and Yoram Singer. 1998. Learning to order things. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10(NIPS '97)*, Michael I. Jordan, Michael J. Kearns, and Sara A. Solla (Eds.). MIT Press, Cambridge, MA, USA, 451-457.
- Yanxiang He, Dexi Liu, Hua Yang, Donghong Ji, Chong Teng, and Wenqing Qi. 2006. A hybrid sentence ordering strategy in multi-document summarization. In *Proceedings of the 7th international conference on Web Information Systems (WISE'06)*, Karl Aberer, Zhiyong Peng, Elke A. Rundensteiner, Yanchun Zhang, and Xuhui Li (Eds.). Springer-Verlag, Berlin, Heidelberg, 339-349.
- Yu Nie, Donghong Ji, and Lingpeng Yang. 2006. An adjacency model for sentence ordering in multi-document summarization. In *Proceedings of the Third Asia conference on Information Retrieval Technology (AIRS'06)*, 313-322.

# Universal Dependency Annotation for Multilingual Parsing

Ryan McDonald<sup>†</sup> Joakim Nivre<sup>†\*</sup> Yvonne Quirnbach-Brundage<sup>‡</sup> Yoav Goldberg<sup>†\*</sup>  
Dipanjan Das<sup>†</sup> Kuzman Ganchev<sup>†</sup> Keith Hall<sup>†</sup> Slav Petrov<sup>†</sup> Hao Zhang<sup>†</sup>  
Oscar Täckström<sup>†\*</sup> Claudia Bedini<sup>‡</sup> Núria Bertomeu Castelló<sup>‡</sup> Jungmee Lee<sup>‡</sup>  
Google, Inc.<sup>†</sup> Uppsala University\* Appen-Butler-Hill<sup>‡</sup> Bar-Ilan University\*  
Contact: ryanmcd@google.com

## Abstract

We present a new collection of treebanks with homogeneous syntactic dependency annotation for six languages: German, English, Swedish, Spanish, French and Korean. To show the usefulness of such a resource, we present a case study of cross-lingual transfer parsing with more reliable evaluation than has been possible before. This ‘universal’ treebank is made freely available in order to facilitate research on multilingual dependency parsing.<sup>1</sup>

## 1 Introduction

In recent years, syntactic representations based on head-modifier dependency relations between words have attracted a lot of interest (Kübler et al., 2009). Research in dependency parsing – computational methods to predict such representations – has increased dramatically, due in large part to the availability of dependency treebanks in a number of languages. In particular, the CoNLL shared tasks on dependency parsing have provided over twenty data sets in a standardized format (Buchholz and Marsi, 2006; Nivre et al., 2007).

While these data sets are standardized in terms of their formal representation, they are still heterogeneous treebanks. That is to say, despite them all being dependency treebanks, which annotate each sentence with a dependency tree, they subscribe to different annotation schemes. This can include superficial differences, such as the renaming of common relations, as well as true divergences concerning the analysis of linguistic constructions. Common divergences are found in the

analysis of coordination, verb groups, subordinate clauses, and multi-word expressions (Nilsson et al., 2007; Kübler et al., 2009; Zeman et al., 2012).

These data sets can be sufficient if one’s goal is to build monolingual parsers and evaluate their quality without reference to other languages, as in the original CoNLL shared tasks, but there are many cases where heterogeneous treebanks are less than adequate. First, a homogeneous representation is critical for multilingual language technologies that require consistent cross-lingual analysis for downstream components. Second, consistent syntactic representations are desirable in the evaluation of unsupervised (Klein and Manning, 2004) or cross-lingual syntactic parsers (Hwa et al., 2005). In the cross-lingual study of McDonald et al. (2011), where delexicalized parsing models from a number of source languages were evaluated on a set of target languages, it was observed that the best target language was frequently not the closest typologically to the source. In one stunning example, Danish was the worst source language when parsing Swedish, solely due to greatly divergent annotation schemes.

In order to overcome these difficulties, some cross-lingual studies have resorted to heuristics to homogenize treebanks (Hwa et al., 2005; Smith and Eisner, 2009; Ganchev et al., 2009), but we are only aware of a few systematic attempts to create homogeneous syntactic dependency annotation in multiple languages. In terms of automatic construction, Zeman et al. (2012) attempt to harmonize a large number of dependency treebanks by mapping their annotation to a version of the Prague Dependency Treebank scheme (Hajič et al., 2001; Böhmová et al., 2003). Additionally, there have been efforts to manually or semi-manually construct resources with common syn-

<sup>1</sup>Downloadable at <https://code.google.com/p/uni-dep-tb/>.

tactic analyses across multiple languages using alternate syntactic theories as the basis for the representation (Butt et al., 2002; Helmreich et al., 2004; Hovy et al., 2006; Erjavec, 2012).

In order to facilitate research on multilingual syntactic analysis, we present a collection of data sets with uniformly analyzed sentences for six languages: German, English, French, Korean, Spanish and Swedish. This resource is freely available and we plan to extend it to include more data and languages. In the context of part-of-speech tagging, universal representations, such as that of Petrov et al. (2012), have already spurred numerous examples of improved empirical cross-lingual systems (Zhang et al., 2012; Gelling et al., 2012; Täckström et al., 2013). We aim to do the same for syntactic dependencies and present cross-lingual parsing experiments to highlight some of the benefits of cross-lingually consistent annotation. First, results largely conform to our expectations of which target languages should be useful for which source languages, unlike in the study of McDonald et al. (2011). Second, the evaluation scores in general are significantly higher than previous cross-lingual studies, suggesting that most of these studies underestimate true accuracy. Finally, unlike all previous cross-lingual studies, we can report full labeled accuracies and not just unlabeled structural accuracies.

## 2 Towards A Universal Treebank

The Stanford typed dependencies for English (De Marneffe et al., 2006; de Marneffe and Manning, 2008) serve as the point of departure for our ‘universal’ dependency representation, together with the tag set of Petrov et al. (2012) as the underlying part-of-speech representation. The Stanford scheme, partly inspired by the LFG framework, has emerged as a de facto standard for dependency annotation in English and has recently been adapted to several languages representing different (and typologically diverse) language groups, such as Chinese (Sino-Tibetan) (Chang et al., 2009), Finnish (Finno-Ugric) (Haverinen et al., 2010), Persian (Indo-Iranian) (Seraji et al., 2012), and Modern Hebrew (Semitic) (Tsarfaty, 2013). Its widespread use and proven adaptability makes it a natural choice for our endeavor, even though additional modifications will be needed to capture the full variety of grammatical structures in the world’s languages.

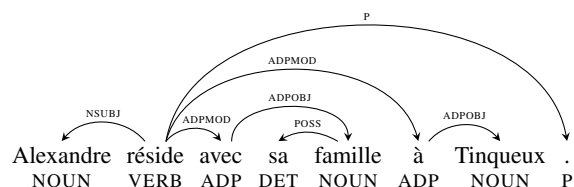


Figure 1: A sample French sentence.

We use the so-called *basic* dependencies (with punctuation included), where every dependency structure is a tree spanning all the input tokens, because this is the kind of representation that most available dependency parsers require. A sample dependency tree from the French data set is shown in Figure 1. We take two approaches to generating data. The first is traditional manual annotation, as previously used by Helmreich et al. (2004) for multilingual syntactic treebank construction. The second, used only for English and Swedish, is to automatically convert existing treebanks, as in Zeman et al. (2012).

### 2.1 Automatic Conversion

Since the Stanford dependencies for English are taken as the starting point for our universal annotation scheme, we begin by describing the data sets produced by automatic conversion. For English, we used the Stanford parser (v1.6.8) (Klein and Manning, 2003) to convert the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993) to basic dependency trees, including punctuation and with the copula verb as head in copula constructions. For Swedish, we developed a set of deterministic rules for converting the Talbanken part of the Swedish Treebank (Nivre and Megyesi, 2007) to a representation as close as possible to the Stanford dependencies for English. This mainly consisted in relabeling dependency relations and, due to the fine-grained label set used in the Swedish Treebank (Teleman, 1974), this could be done with high precision. In addition, a small number of constructions required structural conversion, notably coordination, which in the Swedish Treebank is given a Prague style analysis (Nilsson et al., 2007). For both English and Swedish, we mapped the language-specific part-of-speech tags to universal tags using the mappings of Petrov et al. (2012).

### 2.2 Manual Annotation

For the remaining four languages, annotators were given three resources: 1) the English Stanford

guidelines; 2) a set of English sentences with Stanford dependencies and universal tags (as above); and 3) a large collection of unlabeled sentences randomly drawn from newswire, weblogs and/or consumer reviews, automatically tokenized with a rule-based system. For German, French and Spanish, contractions were split, except in the case of clitics. For Korean, tokenization was more coarse and included particles within token units. Annotators could correct this automatic tokenization.

The annotators were then tasked with producing language-specific annotation guidelines with the expressed goal of keeping the label and construction set as close as possible to the original English set, only adding labels for phenomena that do not exist in English. Making fine-grained label distinctions was discouraged. Once these guidelines were fixed, annotators selected roughly an equal amount of sentences to be annotated from each domain in the unlabeled data. As the sentences were already randomly selected from a larger corpus, annotators were told to view the sentences in order and to discard a sentence only if it was 1) fragmented because of a sentence splitting error; 2) not from the language of interest; 3) incomprehensible to a native speaker; or 4) shorter than three words. The selected sentences were pre-processed using cross-lingual taggers (Das and Petrov, 2011) and parsers (McDonald et al., 2011).

The annotators modified the pre-parsed trees using the TrEd<sup>2</sup> tool. At the beginning of the annotation process, double-blind annotation, followed by manual arbitration and consensus, was used iteratively for small batches of data until the guidelines were finalized. Most of the data was annotated using single-annotation and full review: one annotator annotating the data and another reviewing it, making changes in close collaboration with the original annotator. As a final step, all annotated data was semi-automatically checked for annotation consistency.

### 2.3 Harmonization

After producing the two converted and four annotated data sets, we performed a harmonization step, where the goal was to maximize consistency of annotation across languages. In particular, we wanted to eliminate cases where the same label was used for different linguistic relations in different languages and, conversely, where one and

the same relation was annotated with different labels, both of which could happen accidentally because annotators were allowed to add new labels for the language they were working on. Moreover, we wanted to avoid, as far as possible, labels that were only used in one or two languages.

In order to satisfy these requirements, a number of language-specific labels were merged into more general labels. For example, in analogy with the *nn* label for (element of a) noun-noun compound, the annotators of German added *aa* for compound adjectives, and the annotators of Korean added *vv* for compound verbs. In the harmonization step, these three labels were merged into a single label *compmo* for modifier in compound.

In addition to harmonizing language-specific labels, we also renamed a small number of relations, where the name would be misleading in the universal context (although quite appropriate for English). For example, the label *prep* (for a modifier headed by a preposition) was renamed *adpmod*, to make clear the relation to other modifier labels and to allow postpositions as well as prepositions.<sup>3</sup> We also eliminated a few distinctions in the original Stanford scheme that were not annotated consistently across languages (e.g., merging *complm* with *mark*, *number* with *num*, and *purpcl* with *advcl*).

The final set of labels is listed with explanations in Table 1. Note that relative to the universal part-of-speech tagset of Petrov et al. (2012) our final label set is quite rich (40 versus 12). This is due mainly to the fact that the former is based on deterministic mappings from a large set of annotation schemes and therefore reduced to the granularity of the greatest common denominator. Such a reduction may ultimately be necessary also in the case of dependency relations, but since most of our data sets were created through manual annotation, we could afford to retain a fine-grained analysis, knowing that it is always possible to map from finer to coarser distinctions, but not vice versa.<sup>4</sup>

### 2.4 Final Data Sets

Table 2 presents the final data statistics. The number of sentences, tokens and tokens/sentence vary

<sup>3</sup>Consequently, *pobj* and *pcomp* were changed to *adpobj* and *adpcomp*.

<sup>4</sup>The only two data sets that were created through conversion in our case were English, for which the Stanford dependencies were originally defined, and Swedish, where the native annotation happens to have a fine-grained label set.

<sup>2</sup>Available at <http://ufal.mff.cuni.cz/tred/>.

| Label   | Description               | Label     | Description             | Label     | Description              |
|---------|---------------------------|-----------|-------------------------|-----------|--------------------------|
| acompl  | adjectival complement     | compmod   | compound modifier       | nmod      | noun modifier            |
| adp     | adposition                | conj      | conjunct                | nsubj     | nominal subject          |
| adpcomp | complement of adposition  | cop       | copula                  | nsubjpass | passive nominal subject  |
| adpmo   | adpositional modifier     | csubj     | clausal subject         | num       | numeric modifier         |
| adpobj  | object of adposition      | csubjpass | passive clausal subject | p         | punctuation              |
| advcl   | adverbial clause modifier | dep       | generic                 | parataxis | parataxis                |
| advmo   | adverbial modifier        | det       | determiner              | partmo    | participial modifier     |
| amod    | adjectival modifier       | doj       | direct object           | poss      | possessive               |
| appos   | appositive                | expl      | expletive               | prt       | verb particle            |
| attr    | attribute                 | infmo     | infinitival modifier    | rcmo      | relative clause modifier |
| aux     | auxiliary                 | ioj       | indirect object         | rel       | relative                 |
| auxpass | passive auxiliary         | mark      | marker                  | xcomp     | open clausal complement  |
| cc      | conjunction               | mwe       | multi-word expression   |           |                          |
| ccomp   | clausal complement        | neg       | negation                |           |                          |

Table 1: Harmonized label set based on Stanford dependencies (De Marneffe et al., 2006).

|    | source(s) | # sentences | # tokens  |
|----|-----------|-------------|-----------|
| DE | N, R      | 4,000       | 59,014    |
| EN | PTB*      | 43,948      | 1,046,829 |
| SV | STB†      | 6,159       | 96,319    |
| ES | N, B, R   | 4,015       | 112,718   |
| FR | N, B, R   | 3,978       | 90,000    |
| KO | N, B      | 6,194       | 71,840    |

Table 2: Data set statistics. \*Automatically converted WSJ section of the PTB. The data release includes scripts to generate this data, not the data itself. †Automatically converted Talbanken section of the Swedish Treebank. N=News, B=Blogs, R=Consumer Reviews.

due to the source and tokenization. For example, Korean has 50% more sentences than Spanish, but ~40k less tokens due to a more coarse-grained tokenization. In addition to the data itself, annotation guidelines and harmonization rules are included so that the data can be regenerated.

### 3 Experiments

One of the motivating factors in creating such a data set was improved cross-lingual transfer evaluation. To test this, we use a cross-lingual transfer parser similar to that of McDonald et al. (2011). In particular, it is a perceptron-trained shift-reduce parser with a beam of size 8. We use the features of Zhang and Nivre (2011), except that all lexical identities are dropped from the templates during training and testing, hence inducing a ‘delexicalized’ model that employs only ‘universal’ properties from source-side treebanks, such as part-of-speech tags, labels, head-modifier distance, etc.

We ran a number of experiments, which can be seen in Table 3. For these experiments we ran-

domly split each data set into training, development and testing sets.<sup>5</sup> The one exception is English, where we used the standard splits. Each row in Table 3 represents a source training language and each column a target evaluation language. We report both unlabeled attachment score (UAS) and labeled attachment score (LAS) (Buchholz and Marsi, 2006). This is likely the first reliable cross-lingual parsing evaluation. In particular, previous studies could not even report LAS due to differences in treebank annotations.

We can make several interesting observations. Most notably, for the Germanic and Romance target languages, the best source language is from the same language group. This is in stark contrast to the results of McDonald et al. (2011), who observe that this is rarely the case with the heterogeneous CoNLL treebanks. Among the Germanic languages, it is interesting to note that Swedish is the best source language for both German and English, which makes sense from a typological point of view, because Swedish is intermediate between German and English in terms of word order properties. For Romance languages, the cross-lingual parser is approaching the accuracy of the supervised setting, confirming that for these languages much of the divergence is lexical and not structural, which is not true for the Germanic languages. Finally, Korean emerges as a very clear outlier (both as a source and as a target language), which again is supported by typological considerations as well as by the difference in tokenization.

With respect to evaluation, it is interesting to compare the absolute numbers to those reported in McDonald et al. (2011) for the languages com-

<sup>5</sup>These splits are included in the release of the data.

| Source Training Language | Target Test Language             |              |              |              |              |              |                                |              |              |              |              |              |
|--------------------------|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------------------------|--------------|--------------|--------------|--------------|--------------|
|                          | Unlabeled Attachment Score (UAS) |              |              |              |              |              | Labeled Attachment Score (LAS) |              |              |              |              |              |
|                          | Germanic                         |              |              | Romance      |              |              | Germanic                       |              |              | Romance      |              |              |
|                          | DE                               | EN           | SV           | ES           | FR           | KO           | DE                             | EN           | SV           | ES           | FR           | KO           |
| DE                       | 74.86                            | 55.05        | 65.89        | 60.65        | 62.18        | 40.59        | 64.84                          | 47.09        | 53.57        | 48.14        | 49.59        | <b>27.73</b> |
| EN                       | 58.50                            | 83.33        | <b>70.56</b> | 68.07        | 70.14        | <b>42.37</b> | 48.11                          | 78.54        | <b>57.04</b> | 56.86        | 58.20        | 26.65        |
| SV                       | <b>61.25</b>                     | <b>61.20</b> | 80.01        | 67.50        | 67.69        | 36.95        | <b>52.19</b>                   | <b>49.71</b> | 70.90        | 54.72        | 54.96        | 19.64        |
| ES                       | 55.39                            | 58.56        | 66.84        | 78.46        | <b>75.12</b> | 30.25        | 45.52                          | 47.87        | 53.09        | 70.29        | <b>63.65</b> | 16.54        |
| FR                       | 55.05                            | 59.02        | 65.05        | <b>72.30</b> | 81.44        | 35.79        | 45.96                          | 47.41        | 52.25        | <b>62.56</b> | 73.37        | 20.84        |
| KO                       | 33.04                            | 32.20        | 27.62        | 26.91        | 29.35        | 71.22        | 26.36                          | 21.81        | 18.12        | 18.63        | 19.52        | 55.85        |

Table 3: Cross-lingual transfer parsing results. Bolded are the best per target cross-lingual result.

mon to both studies (DE, EN, SV and ES). In that study, UAS was in the 38–68% range, as compared to 55–75% here. For Swedish, we can even measure the difference exactly, because the test sets are the same, and we see an increase from 58.3% to 70.6%. This suggests that most cross-lingual parsing studies have underestimated accuracies.

#### 4 Conclusion

We have released data sets for six languages with consistent dependency annotation. After the initial release, we will continue to annotate data in more languages as well as investigate further automatic treebank conversions. This may also lead to modifications of the annotation scheme, which should be regarded as preliminary at this point. Specifically, with more typologically and morphologically diverse languages being added to the collection, it may be advisable to consistently enforce the principle that content words take function words as dependents, which is currently violated in the analysis of adpositional and copula constructions. This will ensure a consistent analysis of functional elements that in some languages are not realized as free words or are not obligatory, such as adpositions which are often absent due to case inflections in languages like Finnish. It will also allow the inclusion of language-specific functional or morphological markers (case markers, topic markers, classifiers, etc.) at the leaves of the tree, where they can easily be ignored in applications that require a uniform cross-lingual representation. Finally, this data is available on an open source repository in the hope that the community will commit new data and make corrections to existing annotations.

#### Acknowledgments

Many people played critical roles in the process of creating the resource. At Google, Fer-

nando Pereira, Alfred Spector, Kannan Pashupathy, Michael Riley and Corinna Cortes supported the project and made sure it had the required resources. Jennifer Bahk and Dave Orr helped coordinate the necessary contracts. Andrea Held, Supreet Chinnan, Elizabeth Hewitt, Tu Tsao and Leigha Weinberg made the release process smooth. Michael Ringgaard, Andy Golding, Terry Koo, Alexander Rush and many others provided technical advice. Hans Uszkoreit gave us permission to use a subsample of sentences from the Tiger Treebank (Brants et al., 2002), the source of the news domain for our German data set. Annotations were additionally provided by Sulki Kim, Patrick McCrae, Laurent Alamarguy and Héctor Fernández Alcalde.

#### References

- Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Kluwer.
- Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.
- Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*.
- Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*.
- Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*.



- Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL-HLT*.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*.
- Marie-Catherine De Marneffe, Bill MacCartney, and Chris D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.
- Tomaz Erjavec. 2012. MULTEXT-East: Morphosyntactic resources for Central and Eastern European languages. *Language Resources and Evaluation*, 46:131–142.
- Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of ACL-IJCNLP*.
- Douwe Gelling, Trevor Cohn, Phil Blunsom, and Joao Graça. 2012. The pascal challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*.
- Jan Hajič, Barbora Vidova Hladka, Jarmila Panevová, Eva Hajičová, Petr Sgall, and Petr Pajas. 2001. Prague Dependency Treebank 1.0. LDC, 2001T10.
- Katri Haverinen, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Filip Ginter, and Tapio Salakoski. 2010. Treebanking finnish. In *Proceedings of The Ninth International Workshop on Treebanks and Linguistic Theories (TLT9)*.
- Stephen Helmsreich, David Farwell, Bonnie Dorr, Nizar Habash, Lori Levin, Teruko Mitamura, Florence Reeder, Keith Miller, Eduard Hovy, Owen Rambow, and Advait Siddharthan. 2004. Interlingual annotation of multilingual text corpora. In *Proceedings of the HLT-EACL Workshop on Frontiers in Corpus Annotation*.
- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of NAACL*.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(03):311–325.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*.
- Dan Klein and Chris D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of ACL*.
- Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Morgan and Claypool.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330.
- Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of EMNLP*.
- Jens Nilsson, Joakim Nivre, and Johan Hall. 2007. Generalizing tree transformations for inductive dependency parsing. In *Proceedings of ACL*.
- Joakim Nivre and Beáta Megyesi. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL*.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.
- Mojgan Seraji, Beáta Megyesi, and Nivre Joakim. 2012. Bootstrapping a Persian dependency treebank. *Linguistic Issues in Language Technology*, 7(18):1–10.
- David A. Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*.
- Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the ACL*.
- Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur.
- Reut Tsarfaty. 2013. A unified morpho-syntactic scheme of stanford dependencies. *Proceedings of ACL*.
- Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. 2012. Hamlet: To parse or not to parse. In *Proceedings of LREC*.
- Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL-HLT*.
- Yuan Zhang, Roi Reichart, Regina Barzilay, and Amir Globerson. 2012. Learning to map into a universal pos tagset. In *Proceedings of EMNLP*.

# An Empirical Examination of Challenges in Chinese Parsing

Jonathan K. Kummerfeld<sup>†</sup> Daniel Tse<sup>‡</sup> James R. Curran<sup>‡</sup> Dan Klein<sup>†</sup>

<sup>†</sup>Computer Science Division  
University of California, Berkeley  
Berkeley, CA 94720, USA  
{jkk,klein}@cs.berkeley.edu

<sup>‡</sup>School of Information Technology  
University of Sydney  
Sydney, NSW 2006, Australia  
{dtse6695,james}@it.usyd.edu.au

## Abstract

Aspects of Chinese syntax result in a distinctive mix of parsing challenges. However, the contribution of individual sources of error to overall difficulty is not well understood. We conduct a comprehensive automatic analysis of error types made by Chinese parsers, covering a broad range of error types for large sets of sentences, enabling the first empirical ranking of Chinese error types by their performance impact. We also investigate which error types are resolved by using gold part-of-speech tags, showing that improving Chinese tagging only addresses certain error types, leaving substantial outstanding challenges.

## 1 Introduction

A decade of Chinese parsing research, enabled by the Penn Chinese Treebank (PCTB; Xue et al., 2005), has seen Chinese parsing performance improve from 76.7  $F_1$  (Bikel and Chiang, 2000) to 84.1  $F_1$  (Qian and Liu, 2012). While recent advances have focused on understanding and reducing the errors that occur in segmentation and part-of-speech tagging (Qian and Liu, 2012; Jiang et al., 2009; Forst and Fang, 2009), a range of substantial issues remain that are purely syntactic.

Early work by Levy and Manning (2003) presented modifications to a parser motivated by a manual investigation of parsing errors. They noted substantial differences between Chinese and English parsing, attributing some of the differences to treebank annotation decisions and others to meaningful differences in syntax. Based on this analysis they considered how to modify their parser to capture the information necessary to model the syntax within the PCTB. However, their manual analysis was limited in scope, covering only part of the parser output, and was unable to characterize the relative impact of the issues they uncovered.

This paper presents a more comprehensive analysis of errors in Chinese parsing, building on the technique presented in Kummerfeld et al. (2012), which characterized the error behavior of English parsers by quantifying how often they make errors such as PP attachment and coordination scope. To accommodate error classes that are absent in English, we augment the system to recognize Chinese-specific parse errors.<sup>1</sup> We use the modified system to show the relative impact of different error types across a range of Chinese parsers.

To understand the impact of tagging errors on different error types, we performed a part-of-speech ablation experiment, in which particular confusions are introduced in isolation. By analyzing the distribution of errors in the system output with and without gold part-of-speech tags, we are able to isolate and quantify the error types that can be resolved by improvements in tagging accuracy.

Our analysis shows that improvements in tagging accuracy can only address a subset of the challenges of Chinese syntax. Further improvement in Chinese parsing performance will require research addressing other challenges, in particular, determining coordination scope.

## 2 Background

The closest previous work is the detailed manual analysis performed by Levy and Manning (2003). While their focus was on issues faced by their factored PCFG parser (Klein and Manning, 2003b), the error types they identified are general issues presented by Chinese syntax in the PCTB. They presented several Chinese error types that are rare or absent in English, including noun/verb ambiguity, NP-internal structure and coordination ambiguity due to *pro*-drop, suggesting that closing the English-Chinese parsing gap demands techniques

<sup>1</sup>The system described in this paper is available from <http://code.google.com/p/berkeley-parser-analyser/>

beyond those currently used for English. However, as noted in their final section, their manual analysis of parse errors in 100 sentences only covered a portion of a single parser’s output, limiting the conclusions they could reach regarding the distribution of errors in Chinese parsing.

## 2.1 Automatic Error Analysis

Our analysis builds on Kummerfeld et al. (2012), which presented a system that automatically classifies English parse errors using a two stage process. First, the system finds the shortest path from the system output to the gold annotations, where each step in the path is a tree transformation, fixing at least one bracket error. Second, each transformation step is classified into one of several error types.

When directly applied to Chinese parser output, the system placed over 27% of the errors in the catch-all ‘Other’ type. Many of these errors clearly fall into one of a small set of error types, motivating an adaptation to Chinese syntax.

## 3 Adapting error analysis to Chinese

To adapt the Kummerfeld et al. (2012) system to Chinese, we developed a new version of the second stage of the system, which assigns an error category to each tree transformation step.

To characterize the errors the original system placed in the ‘Other’ category, we looked through one hundred sentences, identifying error types generated by Chinese syntax that the existing system did not account for. With these observations we were able to implement new rules to catch the previously missed cases, leading to the set shown in Table 1. To ensure the accuracy of our classifications, we alternated between refining the classification code and looking at affected classifications to identify issues. We also periodically changed the sentences from the development set we manually checked, to avoid over-fitting.

Where necessary, we also expanded the information available during classification. For example, we use the structure of the final gold standard tree when classifying errors that are a byproduct of sense disambiguation errors.

## 4 Chinese parsing errors

Table 1 presents the errors made by the Berkeley parser. Below we describe the error types that are

| Error Type              | Brackets | % of total |
|-------------------------|----------|------------|
| NP-internal*            | 6019     | 22.70%     |
| Coordination            | 2781     | 10.49%     |
| Verb taking wrong args* | 2310     | 8.71%      |
| Unary                   | 2262     | 8.53%      |
| Modifier Attachment     | 1900     | 7.17%      |
| One Word Span           | 1560     | 5.88%      |
| Different label         | 1418     | 5.35%      |
| Unary A-over-A          | 1208     | 4.56%      |
| Wrong sense/bad attach* | 1018     | 3.84%      |
| Noun boundary error*    | 685      | 2.58%      |
| VP Attachment           | 626      | 2.36%      |
| Clause Attachment       | 542      | 2.04%      |
| PP Attachment           | 514      | 1.94%      |
| Split Verb Compound*    | 232      | 0.88%      |
| Scope error*            | 143      | 0.54%      |
| NP Attachment           | 109      | 0.41%      |
| Other                   | 3186     | 12.02%     |

Table 1: Errors made when parsing Chinese. Values are the number of bracket errors attributed to that error type. The values shown are for the Berkeley parser, evaluated on the development set. \* indicates error types that were added or substantially changed as part of this work.

either new in this analysis, have had their definition altered, or have an interesting distribution.<sup>2</sup>

In all of our results we follow Kummerfeld et al. (2012), presenting the number of bracket errors (missing or extra) attributed to each error type. Bracket counts are more informative than a direct count of each error type, because the impact on EVALB F-score varies between errors, e.g. a single attachment error can cause 20 bracket errors, while a unary error causes only one.

**NP-internal.** (Figure 1a). Unlike the Penn Treebank (Marcus et al., 1993), the PCTB annotates some NP-internal structure. We assign this error type when a transformation involves words whose parts of speech in the gold tree are one of: CC, CD, DEG, ETC, JJ, NN, NR, NT and OD.

We investigated the errors that fall into the NP-internal category and found that 49% of the errors involved the creation or deletion of a single preterminal phrasal bracket. These errors arise when a parser proposes a tree in which POS tags (for instance, JJ or NN) occur as siblings of phrasal tags (such as NP), a configuration used by the PCTB bracketing guidelines to indicate complementation as opposed to adjunction (Xue et al., 2005).

<sup>2</sup>For an explanation of the English error types, see Kummerfeld et al. (2012).

**Verb taking wrong args.** (Figure 1b). This error type arises when a verb (e.g. 扭转 *reverse*) is hypothesized to take an incorrect argument (布什 *Bush* instead of 地位 *position*). Note that this also covers some of the errors that Kummerfeld et al. (2012) classified as NP Attachment, changing the distribution for that type.

**Unary.** For mis-application of unary rules we separate out instances in which the two brackets in the production have the the same label (A-over-A). This cases is created when traces are eliminated, a standard step in evaluation. More than a third of unary errors made by the Berkeley parser are of the A-over-A type. This can be attributed to two factors: (i) the PCTB annotates non-local dependencies using traces, and (ii) Chinese syntax generates more traces than English syntax (Guo et al., 2007). However, for parsers that do not return traces they are a benign error.

**Modifier attachment.** (Figure 1c). Incorrect modifier scope caused by modifier phrase attachment level. This is less frequent in Chinese than in English: while English VP modifiers occur in pre- and post-verbal positions, Chinese only allows pre-verbal modification.

**Wrong sense/bad attach.** (Figure 1d). This applies when the head word of a phrase receives the wrong POS, leading to an attachment error. This error type is common in Chinese because of POS fluidity, e.g. the well-known Chinese verb/noun ambiguity often causes mis-attachments that are classified as this error type.

In Figure 1d, the word 投资 *invest* has both noun and verb senses. While the gold standard interpretation is the relative clause *firms that Macau invests in*, the parser returned an NP interpretation *Macau investment firms*.

**Noun boundary error.** In this error type, a span is moved to a position where the POS tags of its new siblings all belong to the list of NP-internal structure tags which we identified above, reflecting the inclusion of additional material into an NP.

**Split verb compound.** The PCTB annotations recognize several Chinese verb compounding strategies, such as the serial verb construction (规划建设 *plan [and] build*) and the resultative construction (煮熟 *cook [until] done*), which join a bare verb to another lexical item. We introduce an error type specific to Chinese, in which such verb compounds are split, with the two halves of the compound placed in different phrases.

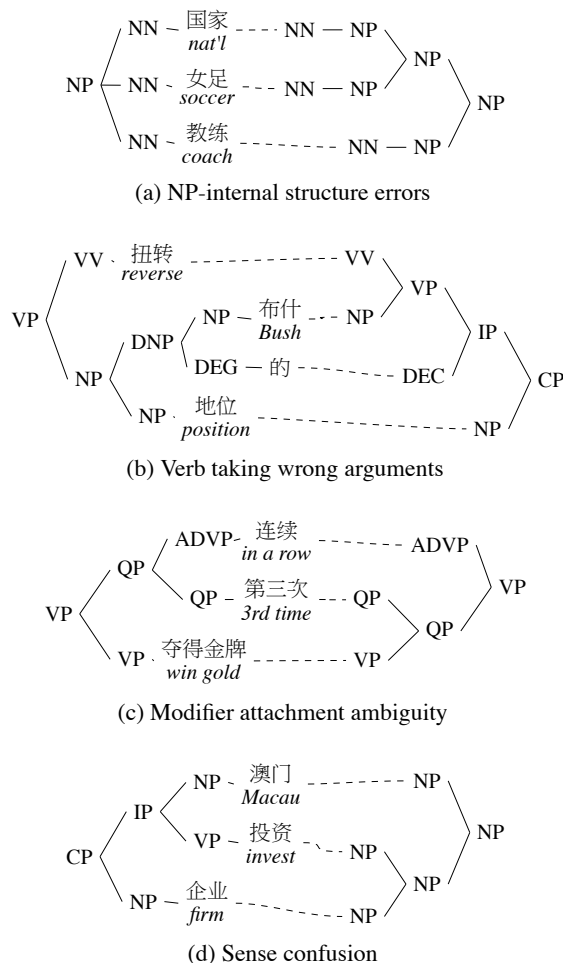


Figure 1: Prominent error types in Chinese parsing. The left tree is the gold structure; the right is the parser hypothesis.

**Scope error.** These are cases in which a new span must be added to more closely bind a modifier phrase (ADVP, ADJP, and PP).

**PP attachment.** This error type is rare in Chinese, as adjunct PPs are pre-verbal. It does occur near coordinated VPs, where ambiguity arises about which of the conjuncts the PP has scope over. Whether this particular case is PP attachment or coordination is debatable; we follow Kummerfeld et al. (2012) and label it PP attachment.

#### 4.1 Chinese-English comparison

It is difficult to directly compare error analysis results for Chinese and English parsing because of substantial changes in the classification method, and differences in treebank annotations.

As described in the previous section, the set of error categories considered for Chinese is very different to the set of categories for English. Even for some of the categories that were not substantially changed, errors may be classified differently because of cross-over between categories between

| System       | F <sub>1</sub> | NP   |       | Verb |       | Mod.   | 1-Word | Diff  | Wrong | Noun | VP     | Clause | PP     | Other |
|--------------|----------------|------|-------|------|-------|--------|--------|-------|-------|------|--------|--------|--------|-------|
|              |                | Int. | Coord | Args | Unary | Attach | Span   | Label | Sense | Edge | Attach | Attach | Attach |       |
| <i>Best</i>  |                | 1.54 | 1.25  | 1.01 | 0.76  | 0.72   | 0.21   | 0.30  | 0.05  | 0.21 | 0.26   | 0.22   | 0.18   | 1.87  |
| Berk-G       | 86.8           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| Berk-2       | 81.8           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| Berk-1       | 81.1           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| ZPAR         | 78.1           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| Bikel        | 76.1           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| Stan-F       | 76.0           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| Stan-P       | 70.0           |      |       |      |       |        |        |       |       |      |        |        |        |       |
| <i>Worst</i> |                | 3.94 | 1.75  | 1.73 | 1.48  | 1.68   | 1.06   | 1.02  | 0.88  | 0.55 | 0.50   | 0.44   | 0.44   | 4.11  |

Table 2: Error breakdown for the development set of PCTB 6. The area filled in for each bar indicates the average number of bracket errors per sentence attributed to that error type, where an empty bar is no errors and a full bar has the value indicated in the bottom row. The parsers are: the Berkeley parser with gold POS tags as input (Berk-G), the Berkeley product parser with two grammars (Berk-2), the Berkeley parser (Berk-1), the parser of Zhang and Clark (2009) (ZPAR), the Bikel parser (Bikel), the Stanford Factored parser (Stan-F), and the Stanford Unlexicalized PCFG parser (Stan-P).

two categories (e.g. between Verb taking wrong args and NP Attachment).

Differences in treebank annotations also present a challenge for cross-language error comparison. The most common error type in Chinese, NP-internal structure, is rare in the results of Kummerfeld et al. (2012), but the datasets are not comparable because the PTB has very limited NP-internal structure annotated. Further characterization of the impact of annotation differences on errors is beyond the scope of this paper.

Three conclusions that can be made are that (i) coordination is a major issue in both languages, (ii) PP attachment is a much greater problem in English, and (iii) a higher frequency of trace-generating syntax in Chinese compared to English poses substantial challenges.

## 5 Cross-parser analysis

The previous section described the error types and their distribution for a single Chinese parser. Here we confirm that these are general trends, by showing that the same pattern is observed for several different parsers on the PCTB 6 dev set.<sup>3</sup> We include results for a transition-based parser (ZPAR; Zhang and Clark, 2009), a split-merge PCFG parser (Petrov et al., 2006; Petrov and Klein, 2007; Petrov, 2010), a lexicalized parser (Bikel and Chiang, 2000), and a factored PCFG and dependency parser (Levy and Manning, 2003; Klein and Manning, 2003a,b).<sup>4</sup>

Comparing the two Stanford parsers in Table 2, the factored model provides clear improvements

<sup>3</sup>We use the standard data split suggested by the PCTB 6 file manifest. As a result, our results differ from those previously reported on other splits. All analysis is on the dev set, to avoid revealing specific information about the test set.

<sup>4</sup>These parsers represent a variety of parsing methods, though exclude some recently developed parsers that are not publicly available (Qian and Liu, 2012; Xiong et al., 2005).

on sense disambiguation, but performs slightly worse on coordination.

The Berkeley product parser we include uses only two grammars because we found, in contrast to the English results (Petrov, 2010), that further grammars provided limited benefits. Comparing the performance with the standard Berkeley parser it seems that the diversity in the grammars only assists certain error types, with most of the improvement occurring in four of the categories, while there is no improvement, or a slight decrease, in five categories.

## 6 Tagging Error Impact

The challenge of accurate POS tagging in Chinese has been a major part of several recent papers (Qian and Liu, 2012; Jiang et al., 2009; Forst and Fang, 2009). The Berk-G row of Table 2 shows the performance of the Berkeley parser when given gold POS tags.<sup>5</sup> While the F<sub>1</sub> improvement is unsurprising, for the first time we can clearly show that the gains are only in a subset of the error types. In particular, tagging improvement will not help for two of the most significant challenges: coordination scope errors, and verb argument selection.

To see which tagging confusions contribute to which error reductions, we adapt the POS ablation approach of Tse and Curran (2012). We consider the POS tag pairs shown in Table 3. To isolate the effects of each confusion we start from the gold tags and introduce the output of the Stanford tagger whenever it returns one of the two tags being considered.<sup>6</sup> We then feed these “semi-gold” tags

<sup>5</sup>We used the Berkeley parser as it was the best of the parsers we considered. Note that the Berkeley parser occasionally prunes all of the parses that use the gold POS tags, and so returns the best available alternative. This leads to a POS accuracy of 99.35%, which is still well above the parser’s standard POS accuracy of 93.66%.

<sup>6</sup>We introduce errors to gold tags, rather than removing er-

| Confused tags |     | Errors | $\Delta F_1$ |
|---------------|-----|--------|--------------|
| VV            | NN  | 1055   | -2.72        |
| DEC           | DEG | 526    | -1.72        |
| JJ            | NN  | 297    | -0.57        |
| NR            | NN  | 320    | -0.05        |

Table 3: The most frequently confused POS tag pairs. Each  $\Delta F_1$  is relative to Berk-G.

to the Berkeley parser, and run the fine-grained error analysis on its output.

**VV/NN.** This confusion has been consistently shown to be a major contributor to parsing errors (Levy and Manning, 2003; Tse and Curran, 2012; Qian and Liu, 2012), and we find a drop of over 2.7  $F_1$  when the output of the tagger is introduced. We found that while most error types have contributions from a range of POS confusions, verb/noun confusion was responsible for virtually all of the noun boundary errors corrected by using gold tags.

**DEG/DEC.** This confusion between the relativizer and subordinator senses of the particle 的 *de* is the primary source of improvements on modifier attachment when using gold tags.

**NR/NN and JJ/NN.** Despite their frequency, these confusions have little effect on parsing performance. Even within the NP-internal error type their impact is limited, and almost all of the errors do not change the logical form.

## 7 Conclusion

We have quantified the relative impacts of a comprehensive set of error types in Chinese parsing. Our analysis has also shown that while improvements in Chinese POS tagging can make a substantial difference for some error types, it will not address two high-frequency error types: incorrect verb argument attachment and coordination scope. The frequency of these two error types is also unimproved by the use of products of latent variable grammars. These observations suggest that resolving the core challenges of Chinese parsing will require new developments that suit the distinctive properties of Chinese syntax.

## Acknowledgments

We extend our thanks to Yue Zhang for helping us train new ZPAR models. We would also like to thank the anonymous reviewers for their helpful suggestions. This research was supported by a General Sir John Monash Fellowship to the first

errors from automatic tags, isolating the effect of a single confusion by eliminating interaction between tagging decisions.

author, the Capital Markets CRC under ARC Discovery grant DP1097291, and the NSF under grant 0643742.

## References

- Daniel M. Bikel and David Chiang. 2000. Two Statistical Parsing Models Applied to the Chinese Treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 1–6. Hong Kong, China.
- Martin Forst and Ji Fang. 2009. TBL-improved non-deterministic segmentation and POS tagging for a Chinese parser. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 264–272. Athens, Greece.
- Yuqing Guo, Haifeng Wang, and Josef van Genabith. 2007. Recovering Non-Local Dependencies for Chinese. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 257–266. Prague, Czech Republic.
- Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging – A Case Study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1, pages 522–530. Suntec, Singapore.
- Dan Klein and Christopher D. Manning. 2003a. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430. Sapporo, Japan.
- Dan Klein and Christopher D. Manning. 2003b. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10. MIT Press, Cambridge, MA.
- Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser Showdown at the Wall Street Corral: An Empirical Investigation of Error Types in Parser Output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059. Jeju Island, South Korea.

- Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 439–446. Sapporo, Japan.
- 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.
- Slav Petrov. 2010. Products of Random Latent Variable Grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Los Angeles, California.
- 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 Association for Computational Linguistics*, pages 433–440. Sydney, Australia.
- Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411. Rochester, New York, USA.
- Xian Qian and Yang Liu. 2012. Joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 501–511. Jeju Island, Korea.
- Daniel Tse and James R. Curran. 2012. The Challenges of Parsing Chinese with Combinatory Categorical 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.
- Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the Penn Chinese Treebank with semantic knowledge. In *Proceedings of the Second international joint conference on Natural Language Processing*, pages 70–81. Jeju Island, Korea.
- 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.

# Joint Inference for Heterogeneous Dependency Parsing

Guangyou Zhou and Jun Zhao

National Laboratory of Pattern Recognition  
Institute of Automation, Chinese Academy of Sciences  
95 Zhongguancun East Road, Beijing 100190, China  
{gyzhou, jzhao}@nlpr.ia.ac.cn

## Abstract

This paper is concerned with the problem of heterogeneous dependency parsing. In this paper, we present a novel joint inference scheme, which is able to leverage the consensus information between heterogeneous treebanks in the parsing phase. Different from stacked learning methods (Nivre and McDonald, 2008; Martins et al., 2008), which process the dependency parsing in a pipelined way (e.g., a second level uses the first level outputs), in our method, multiple dependency parsing models are coordinated to exchange consensus information. We conduct experiments on Chinese Dependency Treebank (CDT) and Penn Chinese Treebank (CTB), experimental results show that joint inference can bring significant improvements to all state-of-the-art dependency parsers.

## 1 Introduction

Dependency parsing is the task of building dependency links between words in a sentence, which has recently gained a wide interest in the natural language processing community and has been used for many problems ranging from machine translation (Ding and Palmer, 2004) to question answering (Zhou et al., 2011a). Over the past few years, supervised learning methods have obtained state-of-the-art performance for dependency parsing (Yamada and Matsumoto, 2003; McDonald et al., 2005; McDonald and Pereira, 2006; Hall et al., 2006; Zhou et al., 2011b; Zhou et al., 2011c). These methods usually rely heavily on the manually annotated treebanks for training the dependency models. However, annotating syntac-

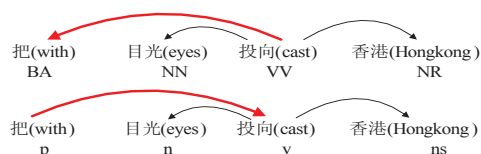


Figure 1: Different grammar formalisms of syntactic structures between CTB (upper) and CDT (below). CTB is converted into dependency grammar based on the head rules of (Zhang and Clark, 2008).

tic structure, either phrase-based or dependency-based, is both time consuming and labor intensive. Making full use of the existing manually annotated treebanks would yield substantial savings in data-annotation costs.

In this paper, we present a joint inference scheme for heterogeneous dependency parsing. This scheme is able to leverage consensus information between heterogeneous treebanks during the inference phase instead of using individual output in a pipelined way, such as stacked learning methods (Nivre and McDonald, 2008; Martins et al., 2008). The basic idea is very simple: although heterogeneous treebanks have different grammar formalisms, they share some consensus information in dependency structures for the same sentence. For example in Figure 1, the dependency structures actually share some partial agreements for the same sentence, the two words “eyes” and “Hongkong” depend on “cast” in both Chinese Dependency Treebank (CDT) (Liu et al., 2006) and Penn Chinese Treebank (CTB) (Xue et al., 2005). Therefore, we would like to train the dependency parsers on individual heterogeneous treebank and jointly parse the same sentences with consensus information exchanged between them.

The remainder of this paper is divided as fol-



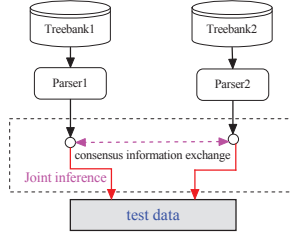


Figure 2: General joint inference scheme of heterogeneous dependency parsing.

lows. Section 2 gives a formal description of the joint inference for heterogeneous dependency parsing. In section 3, we present the experimental results. Finally, we conclude with ideas for future research.

## 2 Our Approach

The general joint inference scheme of heterogeneous dependency parsing is shown in Figure 2. Here, heterogeneous treebanks refer to two Chinese treebanks: CTB and CDT, therefore we have only two parsers, but the framework is generic enough to integrate more parsers. For easy explanation of the joint inference scheme, we regard a parser without consensus information as a *baseline parser*, a parser incorporates consensus information called a *joint parser*. Joint inference provides a framework that accommodates and coordinates multiple dependency parsing models. Similar to Li et al. (2009) and Zhu et al. (2010), the joint inference for heterogeneous dependency parsing consists of four components: (1) Joint Inference Model; (2) Parser Coordination; (3) Joint Inference Features; (4) Parameter Estimation.

### 2.1 Joint Inference Model

For a given sentence  $x$ , a joint dependency parsing model finds the best dependency parsing tree  $y^*$  among the set of possible candidate parses  $\mathcal{Y}(x)$  based on a scoring function  $F_s$ :

$$y^* = \arg \max_{y \in \mathcal{Y}(x)} F_s(x, y) \quad (1)$$

Following (Li et al., 2009), we will use  $d_k$  to denote the  $k$ th joint parser, and also use the notation  $\mathcal{H}_k(x)$  for a list of parse candidates of sentence  $x$  determined by  $d_k$ . The  $s$ th joint parser can be written as:

$$F_s(x, y) = P_s(x, y) + \sum_{k, k \neq s} \Psi_k(y, \mathcal{H}_k(x)) \quad (2)$$

where  $P_s(x, y)$  is the score function of the  $s$ th baseline model, and each  $\Psi_k(y, \mathcal{H}_k(x))$  is a partial

consensus score function with respect to  $d_k$  and is defined over  $y$  and  $\mathcal{H}_k(x)$ :

$$\Psi_k(y, \mathcal{H}_k(x)) = \sum_l \lambda_{k,l} f_{k,l}(y, \mathcal{H}_k(x)) \quad (3)$$

where each  $f_{k,l}(y, \mathcal{H}_k(x))$  is a feature function based on a consensus measure between  $y$  and  $\mathcal{H}_k(x)$ , and  $\lambda_{k,l}$  is the corresponding weight parameter. Feature index  $l$  ranges over all consensus-based features in equation (3).

### 2.2 Parser Coordination

Note that in equation (2), though the baseline score function  $P_s(x, y)$  can be computed individually, the case of  $\Psi_k(y, \mathcal{H}_k(x))$  is more complicated. It is not feasible to enumerate all parse candidates for dependency parsing. In this paper, we use a bootstrapping method to solve this problem. The basic idea is that we can use baseline models’  $n$ -best output as seeds, and iteratively refine joint models’  $n$ -best output with joint inference. The joint inference process is shown in Algorithm 1.

---

#### Algorithm 1 Joint inference for multiple parsers

---

**Step1:** For each joint parser  $d_k$ , perform inference with a baseline model, and memorize all dependency parsing candidates generated during inference in  $\mathcal{H}_k(x)$ ;

**Step2:** For each candidate in  $\mathcal{H}_k(x)$ , we extract subtrees and store them in  $\mathcal{H}'_k(x)$ . First, we extract bigram-subtrees that contain two words. If two words have a dependency relation, we add these two words as a subtree into  $\mathcal{H}'_k(x)$ . Similarly, we can extract trigram-subtrees. Note that the dependency direction is kept. Besides, we also store the “ROOT” word of each candidate in  $\mathcal{H}'_k(x)$ ;

**Step3:** Use joint parsers to re-parse the sentence  $x$  with the baseline features and joint inference features (see subsection 2.3). For joint parser  $d_k$ , consensus-based features of any dependency parsing candidate are computed based on current setting of  $\mathcal{H}'_s(x)$  for all  $s$  but  $k$ . New dependency parsing candidates generated by  $d_k$  in re-parsing are cached in  $\mathcal{H}''_k(x)$ ;

**Step4:** Update all  $\mathcal{H}_k(x)$  with  $\mathcal{H}''_k(x)$ ;

**Step5:** Iterate from Step2 to Step4 until a preset iteration limit is reached.

---

In Algorithm 1, dependency parsing candidates of different parsers can be mutually improved. For example, given two parsers  $d_1$  and  $d_2$  with candidates  $\mathcal{H}_1$  and  $\mathcal{H}_2$ , improvements on  $\mathcal{H}_1$  enable  $d_2$  to improve  $\mathcal{H}_2$ , and  $\mathcal{H}_1$  benefits from improved  $\mathcal{H}_2$ , and so on.

We can see that a joint parser does not enlarge the search space of its baseline model, the only change is parse scoring. By running a complete inference process, joint model can be applied to re-parsing all candidates explored by a parser.

Thus Step3 can be viewed as full-scale candidates reranking because the reranking scope is beyond the limited  $n$ -best output currently cached in  $\mathcal{H}_k$ .

### 2.3 Joint Inference Features

In this section we introduce the consensus-based feature functions  $f_{k,l}(y, \mathcal{H}_k(x))$  introduced in equation (3). The formulation can be written as:

$$f_{k,l}(y, \mathcal{H}_k(x)) = \sum_{y' \in \mathcal{H}_k(x)} P(y'|d_k) I_l(y, y') \quad (4)$$

where  $y$  is a dependency parse of  $x$  by using parser  $d_s$  ( $s \neq k$ ),  $y'$  is a dependency parse in  $\mathcal{H}_k(x)$  and  $P(y'|d_k)$  is the posterior probability of dependency parse  $y'$  parsed by parser  $d_k$  given sentence  $x$ .  $I_l(y, y')$  is a consensus measure defined on  $y$  and  $y'$  using different feature functions.

Dependency parsing model  $P(y'|d_k)$  can be predicted by using the global linear models (GLMs) (e.g., McDonald et al. (2005); McDonald and Pereira (2006)). The consensus-based score functions  $I_l(y, y')$  include the following parts:

(1) *head-modifier dependencies*. Each head-modifier dependency (denoted as “*edge*”) is a tuple  $t = \langle h, m, h \rightarrow m \rangle$ , so  $I_{edge}(y, y') = \sum_{t \in y} \delta(t, y')$ .

(2) *sibling dependencies*: Each sibling dependency (denoted as “*sib*”) is a tuple  $t = \langle i, h, m, h \leftarrow i \rightarrow m \rangle$ , so  $I_{sib}(y, y') = \sum_{t \in y} \delta(t, y')$ .

(3) *grandparent dependencies*: Each grandparent dependency (denoted as “*gp*”) is a tuple  $t = \langle h, i, m, h \rightarrow i \rightarrow m \rangle$ , so  $I_{gp}(y, y') = \sum_{\langle h, i, m, h \rightarrow i \rightarrow m \rangle \in y} \delta(t, y')$ .

(4) *root feature*: This feature (denoted as “*root*”) indicates whether the multiple dependency parsing trees share the same “*ROOT*”, so  $I_{root}(y, y') = \sum_{\langle ROOT \rangle \in y} \delta(\langle ROOT \rangle, y')$ .

$\delta(\cdot, \cdot)$  is an indicator function— $\delta(t, y')$  is 1 if  $t \in y'$  and 0 otherwise, feature index  $l \in \{edge, sib, gp, root\}$  in equation (4). Note that  $\langle h, m, h \rightarrow m \rangle$  and  $\langle m, h, m \rightarrow h \rangle$  are two different edges.

In our joint model, we extend the baseline features of (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007) by conjoining with the consensus-based features, so that we can learn in which kind of contexts the different parsers agree/disagree. For the third-order features (e.g., grand-siblings and tri-siblings) described in (Koo et al., 2010), we will discuss it in future work.

### 2.4 Parameter Estimation

The parameters are tuned to maximize the dependency parsing performance on the development set, using an algorithm similar to the average perceptron algorithm due to its strong performance and fast training (Koo et al., 2008). Due to limited space, we do not present the details. For more information, please refer to (Koo et al., 2008).

## 3 Experiments

In this section, we describe the experiments to evaluate our proposed approach by using CTB4 (Xue et al., 2005) and CDT (Liu et al., 2006). For the former, we adopt a set of head-selection rules (Zhang and Clark, 2008) to convert the phrase structure syntax of treebank into a dependency tree representation. The standard data split of CTB4 from Wang et al. (2007) is used. For the latter, we randomly select 2,000 sentences for test set, another 2,000 sentences for development set, and others for training set.

We use two baseline parsers, one trained on CTB4, and another trained on CDT in the experiments. We choose the  $n$ -best size of 16 and the best iteration time of four on the development set since these settings empirically give the best performance. CTB4 and CDT use two different POS tag sets and transforming from one tag set to another is difficult (Niu et al., 2009). To overcome this problem, we use Stanford POS Tagger<sup>1</sup> to train a universal POS tagger on the People’s Daily corpus,<sup>2</sup> a large-scale Chinese corpus (approximately 300 thousand sentences and 7 million words) annotated with word segmentation and POS tags. Then the POS tagger produces a universal layer of POS tags for both the CTB4 and CDT. Note that the word segmentation standards of these corpora (CTB4, CDT and People’s Daily) slightly differs; however, we do not consider this problem and leave it for future research.

The performance of the parsers is evaluated using the following metrics: UAS, DA, and CM, which are defined by (Hall et al., 2006). All the metrics except CM are calculated as mean scores per word, and punctuation tokens are consistently excluded.

We conduct experiments incrementally to evaluate the joint features used in our first-order and second-order parsers. **The first-order parser**

<sup>1</sup><http://nlp.stanford.edu/software/tagger.shtml>

<sup>2</sup><http://www.icl.pku.edu.cn>

| -    | Features      | CTB4                     |                          | CDT                      |                          |
|------|---------------|--------------------------|--------------------------|--------------------------|--------------------------|
|      |               | UAS                      | CM                       | UAS                      | CM                       |
| dep1 | baseline      | 86.6                     | 42.5                     | 75.4                     | 16.6                     |
|      | + <i>edge</i> | 88.01 ( $\uparrow$ 1.41) | 44.28 ( $\uparrow$ 1.78) | 77.10 ( $\uparrow$ 1.70) | 17.82 ( $\uparrow$ 1.22) |
|      | + <i>root</i> | 87.22 ( $\uparrow$ 0.62) | 43.03 ( $\uparrow$ 0.53) | 75.83 ( $\uparrow$ 0.43) | 16.81 ( $\uparrow$ 0.21) |
|      | + both        | 88.19 ( $\uparrow$ 1.59) | 44.54 ( $\uparrow$ 2.04) | 77.16 ( $\uparrow$ 1.76) | 17.90 ( $\uparrow$ 1.30) |
|      | CTB4 + CDT    | 87.32                    | 43.08                    | 75.91                    | 16.89                    |
| dep2 | baseline      | 88.38                    | 48.81                    | 77.52                    | 19.70                    |
|      | + <i>edge</i> | 89.17 ( $\uparrow$ 0.79) | 49.73 ( $\uparrow$ 0.92) | 78.44 ( $\uparrow$ 0.92) | 20.85 ( $\uparrow$ 1.15) |
|      | + <i>sib</i>  | 88.94 ( $\uparrow$ 0.56) | 49.26 ( $\uparrow$ 0.45) | 78.02 ( $\uparrow$ 0.50) | 20.13 ( $\uparrow$ 0.43) |
|      | + <i>gp</i>   | 88.90 ( $\uparrow$ 0.52) | 49.11 ( $\uparrow$ 0.30) | 77.97 ( $\uparrow$ 0.45) | 20.06 ( $\uparrow$ 0.36) |
|      | + <i>root</i> | 88.61 ( $\uparrow$ 0.23) | 48.88 ( $\uparrow$ 0.07) | 77.65 ( $\uparrow$ 0.13) | 19.88 ( $\uparrow$ 0.18) |
|      | + all         | 89.62 ( $\uparrow$ 1.24) | 50.15 ( $\uparrow$ 1.34) | 79.01 ( $\uparrow$ 1.49) | 21.11 ( $\uparrow$ 1.41) |
|      | CTB4 + CDT    | 88.91                    | 49.13                    | 78.03                    | 20.12                    |

Table 1: Dependency parsing results on the test set with different joint inference features. Abbreviations: dep1/dep2 = first-order parser and second-order parser; baseline = dep1 without considering any joint inference features; +\* = the baseline features conjoined with the joint inference features derived from the heterogeneous treebanks; CTB4 + CDT = we simply concatenate the two corpora and train a dependency parser, and then test on CTB4 and CDT using this single model. Improvements of joint models over baseline models are shown in parentheses.

| Type | Systems                 | $\leq 40$ | Full  |
|------|-------------------------|-----------|-------|
| D    | dep2                    | 90.86     | 88.38 |
|      | MaltParser              | 87.1      | 85.8  |
|      | Wang et al. (2007)      | 86.6      | -     |
| C    | MST <sub>Malt</sub> †   | 90.55     | 88.82 |
|      | Martins et al. (2008)†  | 90.63     | 88.84 |
|      | Surdeanu et al. (2010)† | 89.40     | 86.63 |
| H    | Zhao et al. (2009)      | 88.9      | 86.1  |
|      | Ours                    | 91.48     | 89.62 |
| S    | Yu et al. (2008)        | -         | 87.26 |
|      | Chen et al. (2009)      | 92.34     | 89.91 |
|      | Chen et al. (2012)      | -         | 91.59 |

Table 2: Comparison of different approach on CTB4 test set using UAS metric. MaltParser = Hall et al. (2006); MST<sub>Malt</sub>=Nivre and McDonald (2008). Type D = discriminative dependency parsers without using any external resources; C = combined parsers (stacked and ensemble parsers); H = discriminative dependency parsers using external resources derived from heterogeneous treebanks, S = discriminative dependency parsers using external unlabeled data. † The results on CTB4 were not directly reported in these papers, we implemented the experiments in this paper.

(dep1) only incorporates head-modifier dependency part (McDonald et al., 2005). The second-order parser (dep2) uses the head-modifier and sibling dependency parts (McDonald and Pereira, 2006), as well as the grandparent dependency part (Carreras, 2007; Koo et al., 2008). Table 1 shows the experimental results.

As shown in Table 1, we note that adding more joint inference features incrementally, the dependency parsing performance is improved consis-

tently, for both treebanks (CTB4 or CDT). As a final note, all comparisons between joint models and baseline models in Table 1 are statistically significant.<sup>3</sup> Furthermore, we also present a baseline method called “CTB4 + CDT” for comparison. This method first tags both CTB4 and CDT with the universal POS tagger trained on the People’s Daily corpus, then simply concatenates the two corpora and trains a dependency parser, and finally tests on CTB4 and CDT using this single model. The comparisons in Table 1 tell us that very limited information is obtained without consensus features by simply taking a union of the dependencies and their contexts from the two treebanks.

To put our results in perspective, we also compare our second-order joint parser with other best-performing systems. “ $\leq 40$ ” refers to the sentence with the length up to 40 and “Full” refers to all the sentences in test set. The results are shown in Table 2, our approach significantly outperforms many systems evaluated on this data set. Chen et al. (2009) and Chen et al. (2012) reported a very high accuracy using subtree-based features and dependency language model based features derived from large-scale data. Our systems did not use such knowledge. Moreover, their technique is orthogonal to ours, and we suspect that combining their subtree-based features into our systems might get an even better performance. We do not present the comparison of our proposed approach

<sup>3</sup>We use the sign test at the sentence level. All the comparisons are significant at  $p < 0.05$ .

| Type | Systems                | UAS         | DA           |
|------|------------------------|-------------|--------------|
| D    | Duan et al. (2007)     | 83.88       | 84.36        |
|      | Huang and Sagae (2010) | 85.20       | 85.52        |
|      | Zhang and Nivre (2011) | 86.0        | -            |
| C    | Zhang and Clark (2008) | -           | 86.21        |
|      | Bohnet and Kuhn (2012) | <b>87.5</b> | -            |
| H    | Li et al. (2012)       | 86.44       | -            |
|      | <i>Ours</i>            | 85.88       | 86.52        |
| S    | Chen et al. (2009)     | -           | <b>86.70</b> |

Table 3: Comparison of different approaches on CTB5 test set. Abbreviations D, C, H and S are as in Table 2.

| Treebanks | #Sen  | # Better | # NoChange | # Worse |
|-----------|-------|----------|------------|---------|
| CTB4      | 355   | 74       | 255        | 26      |
| CDT       | 2,000 | 341      | 1,562      | 97      |

Table 4: Statistics on joint inference output on CTB4 and CDT development set.

with the state-of-the-art methods on CDT because there is little work conducted on this treebank.

Some researchers conducted experiments on CTB5 with a different data split: files 1-815 and files 1,001-1,136 for training, files 886-931 and 1,148-1,151 for development, files 816-885 and files 1,137-1,147 for testing. The development and testing sets were also performed using gold-standard assigned POS tags. We report the experimental results on CTB5 test set in Table 4. Our results are better than most systems on this data split, except Zhang and Nivre (2011), Li et al. (2012) and Chen et al. (2009).

### 3.1 Additional Results

To obtain further information about how dependency parsers benefit from the joint inference, we conduct an initial experiment on CTB4 and CDT. From Table 4, we find that out of 355 sentences on the development set of CTB4, 74 sentences benefit from the joint inference, while 26 sentences suffer from it. For CDT, we also find that out of 2,000 sentences on the development set, 341 sentences benefit from the joint inference, while 97 sentences suffer from it. Although the overall dependency parsing results is improved, joint inference worsens dependency parsing result for some sentences. In order to obtain further information about the error sources, it is necessary to investigate why joint inference gives negative results, we will leave it for future work.

## 4 Conclusion and Future Work

We proposed a novel framework of joint inference, in which multiple dependency parsing mod-

els were coordinated to search for better dependency parses by leveraging the consensus information between heterogeneous treebanks. Experimental results showed that joint inference significantly outperformed the state-of-the-art baseline models.

There are some ways in which this research could be continued. First, recall that the joint inference scheme involves an iterative algorithm by using bootstrapping. Intuitively, there is a lack of formal guarantee. A natural avenue for further research would be the use of more powerful algorithms that provide certificates of optimality; e.g., dual decomposition that aims to develop decoding algorithms with formal guarantees (Rush et al., 2010). Second, we would like to combine our heterogeneous treebank annotations into a unified representation in order to make dependency parsing results comparable across different annotation guidelines (e.g., Tsarfaty et al. (2011)).

## 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). We thank the anonymous reviewers and the prior reviewers of ACL-2012 and AACL-2013 for their insightful comments. We also thank Dr. Li Cai for providing and preprocessing the data set used in this paper.

## References

- B. Bohnet and J. Kuhn. 2012. The best of both worlds—a graph-based completion model for transition-based parsers. In *Proceedings of EACL*.
- X. Carreras. 2007. Experiments with a Higher-order Projective Dependency Parser. In *Proceedings of EMNLP-CoNLL*, pages 957-961.
- W. Chen, D. Kawahara, K. Uchimoto, and Torisawa. 2009. Improving Dependency Parsing with Subtrees from Auto-Parsed Data. In *Proceedings of EMNLP*, pages 570-579.
- W. Chen, M. Zhang, and H. Li. 2012. Utilizing dependency language models for graph-based dependency parsing models. In *Proceedings of ACL*.
- Y. Ding and M. Palmer. 2004. Synchronous dependency insertion grammars: a grammar formalism for syntax based statistical MT. In *Proceedings of*

- the Workshop on Recent Advances in Dependency Grammar*, pages 90-97.
- X. Duan, J. Zhao, and B. Xu. 2007. Probabilistic Models for Action-based Chinese Dependency Parsing. In *Proceedings of ECML/PKDD*.
- J. M. Eisner. 2000. Bilexical Grammars and Their Cubic-Time Parsing Algorithm. Advanced in Probabilistic and Other Parsing Technologies, pages 29-62.
- J. Hall, J. Nivre, and J. Nilsson. 2006. Discriminative Classifier for Deterministic Dependency Parsing. In *Proceedings of ACL*, pages 316-323.
- L. Huang and K. Sagae. 2010. Dynamic Programming for Linear-Time Incremental Parsing. In *Proceedings of ACL*, pages 1077-1086.
- T. Koo, X. Carreras, and M. Collins. 2008. Simple Semi-Supervised Dependency Parsing. In *Proceedings of ACL*.
- T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. 2010. Dual Decomposition for Parsing with Non-Projective Head Automata. In *Proceedings of EMNLP*.
- M. Li, N. Duan, D. Zhang, C.-H. Li, and M. Zhou. 2009. Collaborative Decoding: Partial Hypothesis Re-ranking Using Translation Consensus Between Decoders. In *Proceedings of ACL*, pages 585-592.
- Z. Li, T. Liu, and W. Che. 2012. Exploiting multiple treebanks for parsing with Quasi-synchronous grammars. In *Proceedings of ACL*.
- T. Liu, J. Ma, and S. Li. 2006. Building a Dependency Treebank for Improving Chinese Parser. *Journal of Chinese Languages and Computing*, 16(4):207-224.
- A. F. T. Martins, D. Das, N. A. Smith, and E. P. Xing. 2008. Stacking Dependency Parsers. In *Proceedings of EMNLP*, pages 157-166.
- R. McDonald and F. Pereira. 2006. Online Learning of Approximate Dependency Parsing Algorithms. In *Proceedings of EACL*, pages 81-88.
- R. McDonald, K. Crammer, and F. Pereira. 2005. Online Large-margin Training of Dependency Parsers. In *Proceedings of ACL*, pages 91-98.
- Z. Niu, H. Wang, and H. Wu. 2009. Exploiting Heterogeneous Treebanks for Parsing. In *Proceedings of ACL*, pages 46-54.
- J. Nivre and R. McDonld. 2008. Integrating Graph-based and Transition-based Dependency Parsing. In *Proceedings of ACL*, pages 950-958.
- A. M. Rush, D. Sontag, M. Collins, and T. Jaakkola. 2010. On Dual Decomposition and Linear Programming Relation for Natural Language Processing. In *Proceedings of EMNLP*.
- M. Surdeanu and C. D. Manning. 2010. Ensemble Models for Dependency Parsing: Cheap and Good? In *Proceedings of NAACL*.
- R. Tsarfaty, J. Nivre, and E. Andersson. 2011. Evaluating Dependency Parsing: Robust and Heuristics-Free Cross-Annotation Evaluation. In *Proceedings of EMNLP*.
- J.-N Wang, J.-S. Chang, and K.-Y. Su. 1994. An Automatic Treebank Conversion Algorithm for Corpus Sharing. In *Proceedings of ACL*, pages 248-254.
- Q. I. Wang, D. Lin, and D. Schuurmans. 2007. Simple Training of Dependency Parsers via Structured Boosting. In *Proceedings of IJCAI*, pages 1756-1762.
- N. Xue, F. Xia, F.-D. Chiou, and M. Palmer. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 10(4):1-30.
- Yamada and Matsumoto. 2003. Statistical Sependency Analysis with Support Vector Machines. In *Proceedings of IWPT*, pages 195-206.
- D. H. Younger. 1967. Recognition and Parsing of Context-Free Languages in Time  $n^3$ . *Information and Control*, 12(4):361-379, 1967.
- K. Yu, D. Kawahara, and S. Kurohashi. 2008. Chinese Dependency Parsing with Large Scale Automatically Constructed Case Structures. In *Proceedings of COLING*, pages 1049-1056.
- Y. Zhang and S. Clark. 2008. A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing Using Beam-Search. In *Proceedings of EMNLP*, pages 562-571.
- Y. Zhang and J. Nivre. 2011. Transition-based Dependency Parsing with Rich Non-local Features. In *Proceedings of ACL*, pages 188-193.
- H. Zhao, Y. Song, C. Kit, and G. Zhou. 2009. Cross Language Dependency Parsing Using a Bilingual Lexicon. In *Proceedings of ACL*, pages 55-63.
- G. Zhou, L. Cai, J. Zhao, and K. Liu. 2011. Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives. In *Proceedings of ACL*, pages 653-662.
- 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.
- G. Zhou, L. Cai, K. Liu, and J. Zhao. 2011. Improving Dependency Parsing with Fined-Grained Features. In *Proceedings of IJCNLP*, pages 228-236.
- M. Zhu, J. Zhu, and T. Xiao. 2010. Heterogeneous Parsing via Collaborative Decoding. In *Proceedings of COLING*, pages 1344-1352.

# Easy-First POS Tagging and Dependency Parsing with Beam Search

Ji Ma<sup>†</sup> JingboZhu<sup>†</sup> Tong Xiao<sup>†</sup> Nan Yang<sup>‡</sup>

<sup>†</sup>Natural Language Processing Lab., Northeastern University, Shenyang, China

<sup>‡</sup>MOE-MS Key Lab of MCC, University of Science and Technology of China, Hefei, China

majineu@outlook.com

{zhujingbo, xiaotong}@mail.neu.edu.cn

nyang.ustc@gmail.com

## Abstract

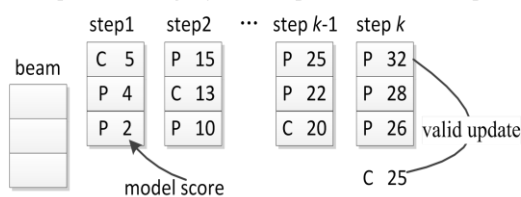
In this paper, we combine easy-first dependency parsing and POS tagging algorithms with beam search and structured perceptron. We propose a simple variant of “early-update” to ensure valid update in the training process. The proposed solution can also be applied to combine beam search and structured perceptron with other systems that exhibit spurious ambiguity. On CTB, we achieve 94.01% tagging accuracy and 86.33% unlabeled attachment score with a relatively small beam width. On PTB, we also achieve state-of-the-art performance.

## 1 Introduction

The easy-first dependency parsing algorithm (Goldberg and Elhadad, 2010) is attractive due to its good accuracy, fast speed and simplicity. The easy-first parser has been applied to many applications (Seeker et al., 2012; Søggard and Wulff, 2012). By processing the input tokens in an easy-to-hard order, the algorithm could make use of structured information on *both sides* of the hard token thus making more indicative predictions. However, rich structured information also causes exhaustive inference intractable. As an alternative, greedy search which only explores a tiny fraction of the search space is adopted (Goldberg and Elhadad, 2010).

To enlarge the search space, a natural extension to greedy search is beam search. Recent work also shows that beam search together with perceptron-based global learning (Collins, 2002) enable the use of non-local features that are helpful to improve parsing performance without overfitting (Zhang and Nivre, 2012). Due to these advantages, beam search and global learning has been applied to many NLP tasks (Collins and

No spurious ambiguity: one unique correct action sequence



Spurious ambiguity: multiple correct action sequences

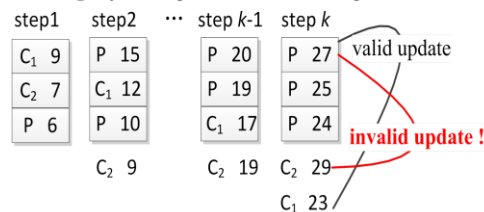


Figure 1: Example of cases without/with spurious ambiguity. The  $3 \times 1$  table denotes a beam. “C/P” denotes correct/predicted action sequence. The numbers following C/P are model scores.

Roark 2004; Zhang and Clark, 2007). However, to the best of our knowledge, no work in the literature has ever applied the two techniques to easy-first dependency parsing.

While applying beam-search is relatively straightforward, the main difficulty comes from combining easy-first dependency parsing with perceptron-based global learning. In particular, one needs to guarantee that each parameter update is *valid*, i.e., the correct action sequence has lower model score than the predicted one<sup>1</sup>. The difficulty in ensuring validity of parameter update for the easy-first algorithm is caused by its spurious ambiguity, i.e., the same result might be derived by more than one action sequences.

For algorithms which do not exhibit spurious ambiguity, “*early update*” (Collins and Roark 2004) is always valid: at the  $k$ -th step when the *single* correct action sequence falls off the beam,

<sup>1</sup> As shown by (Huang et al., 2012), only valid update guarantees the convergence of any perceptron-based training. Invalid update may lead to bad learning or even make the learning not converge at all.

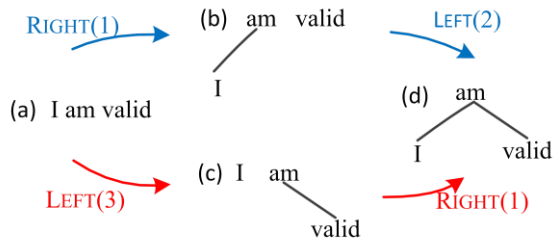


Figure 2: An example of parsing “I am valid”. Spurious ambiguity: (d) can be derived by both [RIGHT(1), LEFT(2)] and [LEFT(3), RIGHT(1)].

its model score must be lower than those still in the beam (as illustrated in figure 1, also see the proof in (Huang et al., 2012)). While for easy-first dependency parsing, there could be multiple action sequences that yield the gold result ( $C_1$  and  $C_2$  in figure 1). When all correct sequences fall off the beam, some may indeed have higher model score than those still in the beam ( $C_2$  in figure 1), causing invalid update.

For the purpose of valid update, we present a simple solution which is based on early update. The basic idea is to use one of the correct action sequences that were pruned right at the  $k$ -th step ( $C_1$  in figure 1) for parameter update.

The proposed solution is general and can also be applied to other algorithms that exhibit spurious ambiguity, such as easy-first POS tagging (Ma et al., 2012) and transition-based dependency parsing with dynamic oracle (Goldberg and Nivre, 2012). In this paper, we report experimental results on both easy-first dependency parsing and POS tagging (Ma et al., 2012). We show that both easy-first POS tagging and dependency parsing can be improved significantly from beam search and global learning. Specifically, on CTB we achieve 94.01% tagging accuracy which is the best result to date<sup>2</sup> for a single tagging model. With a relatively small beam, we achieve 86.33% unlabeled score (assume gold tags), better than state-of-the-art transition-based parsers (Huang and Sagae, 2010; Zhang and Nivre, 2011). On PTB, we also achieve good results that are comparable to the state-of-the-art.

## 2 Easy-first dependency parsing

The easy-first dependency parsing algorithm (Goldberg and Elhadad, 2010) builds a dependency tree by performing two types of actions LEFT( $i$ ) and RIGHT( $i$ ) to a list of sub-tree structures  $p_1, \dots, p_r$ .  $p_i$  is initialized with the  $i$ -th word

<sup>2</sup> Joint tagging-parsing models achieve higher accuracy, but those models are not directly comparable to ours.

---

### Algorithm 1: Easy-first with beam search

---

Input: sentence  $x$  of  $n$  words, beam width  $s$

Output: one best dependency tree

---

$$\text{BEST}_s(x, \beta, \mathbf{w}) \triangleq \text{argtop}_{y' \in \mathcal{Y}_{\text{EXTEN}(y)}}^s \mathbf{w} \cdot \boldsymbol{\varphi}(y')$$

// top  $s$  extensions from the beam

- 1  $\beta_0 \leftarrow []$  // initially, empty beam
  - 2 **for**  $k \in 1 \dots n - 1$  **do**
  - 3      $\beta_k \leftarrow \text{BEST}_s(x, \beta_{k-1}, \mathbf{w})$
  - 4 **return**  $\beta_{n-1}[0](x)$  // tree built by the best sequence
- 

of the input sentence. Action LEFT( $i$ )/RIGHT( $i$ ) attaches  $p_i$  to its left/right neighbor and then removes  $p_i$  from the sub-tree list. The algorithm proceeds until only one sub-tree left which is the dependency tree of the input sentence (see the example in figure 2). Each step, the algorithm chooses the highest score action to perform according to the linear model:

$$\text{Score}(x) = \mathbf{w} \cdot \boldsymbol{\varphi}(x)$$

Here,  $\mathbf{w}$  is the weight vector and  $\boldsymbol{\varphi}$  is the feature representation. In particular,  $\boldsymbol{\varphi}(\text{LEFT}(i)/\text{RIGHT}(i))$  denotes features extracted from  $p_i$ .

The parsing algorithm is greedy which explores a tiny fraction of the search space. Once an incorrect action is selected, it can never yield the correct dependency tree. To enlarge the search space, we introduce the beam-search extension in the next section.

## 3 Easy-first with beam search

In this section, we introduce easy-first with beam search in our own notations that will be used throughout the rest of this paper.

For a sentence  $x$  of  $n$  words, let  $y$  be the action (sub-)sequence that can be applied, in sequence, to  $x$  and the result sub-tree list is denoted by  $y(x)$ . For example, suppose  $x$  is “I am valid” and  $y$  is [RIGHT(1)], then  $y(x)$  yields figure 2(b). Let  $A_l$  to be LEFT( $i$ )/RIGHT( $i$ ) actions where  $i \in [1, l]$ . Thus, the set of all possible one-action extension of  $y$  is:

$$\text{EXTEN}(y) \triangleq \{y \circ a \mid a \in A_{|y(x)|}\}$$

Here, ‘ $\circ$ ’ means insert  $a$  to the end of  $y$ . Following (Huang et al., 2012), in order to formalize beam search, we also use the  $\text{argtop}_{y \in \mathcal{Y}}^s \mathbf{w} \cdot \boldsymbol{\varphi}(y)$  operation which returns the top  $s$  action sequences in  $\mathcal{Y}$  according to  $\mathbf{w} \cdot \boldsymbol{\varphi}(y)$ . Here,  $\mathcal{Y}$  denotes a set of action sequences,  $\boldsymbol{\varphi}(y)$  denotes the sum of feature vectors of each action in  $y$ .

Pseudo-code of easy-first with beam search is shown in algorithm 1. Beam search grows  $s$  (beam width) action sequences in parallel using a

---

**Algorithm 2:** Perceptron-based training over one training sample  $(x, t)$ 

---

Input:  $(x, t)$ ,  $s$ , parameter  $\mathbf{w}$ Output: new parameter  $\mathbf{w}$ 

---

$$\text{TOPC}(x, \beta, \mathbf{w}, \mathcal{C}) \triangleq \operatorname{argmax}_{y \in \mathcal{C} \cap (\cup_{y \in \text{EXTEN}(y)})} \mathbf{w} \cdot \boldsymbol{\varphi}(y')$$

// top correct extension from the beam

```
1  $\beta_0 \leftarrow []$ 
2 for  $k \in 1 \dots n - 1$  do
3    $\hat{y} = \text{TOPC}(x, \beta_{k-1}, \mathbf{w}, \mathcal{C})$ 
4    $\beta_k \leftarrow \text{BEST}_s(x, \beta_{k-1}, \mathbf{w})$ 
5   if  $\beta_k \cap \mathcal{C} = \emptyset$  // all correct seq. falls off the beam
6      $\mathbf{w} \leftarrow \mathbf{w} + \boldsymbol{\varphi}(\hat{y}) - \boldsymbol{\varphi}(\beta_k[0])$ 
7     break
8   if  $\beta_{n-1}[0](x) \neq t$  // full update
9      $\mathbf{w} \leftarrow \mathbf{w} + \boldsymbol{\varphi}(\hat{y}) - \boldsymbol{\varphi}(\beta_{n-1}[0])$ 
10 return  $\mathbf{w}$ 
```

---

beam  $\beta$ , (sequences in  $\beta$  are sorted in terms of model score, i.e.,  $\mathbf{w} \cdot \boldsymbol{\varphi}(\beta[0]) > \mathbf{w} \cdot \boldsymbol{\varphi}(\beta[1]) \dots$ ). At each step, the sequences in  $\beta$  are expanded in all possible ways and then  $\beta$  is filled up with the top  $s$  newly expanded sequences (line 2 ~ line 3). Finally, it returns the dependency tree built by the top action sequence in  $\beta_{n-1}$ .

## 4 Training

To learn the weight vector  $\mathbf{w}$ , we use the perceptron-based global learning<sup>3</sup> (Collins, 2002) which updates  $\mathbf{w}$  by rewarding the feature weights fired in the correct action sequence and punish those fired in the predicted incorrect action sequence. Current work (Huang et al., 2012) rigorously explained that only valid update ensures convergence of any perceptron variants. They also justified that the popular “early update” (Collins and Roark, 2004) is valid for the systems that do not exhibit spurious ambiguity<sup>4</sup>.

However, for the easy-first algorithm or more generally, systems that exhibit spurious ambiguity, even “early update” could fail to ensure validity of update (see the example in figure 1). For validity of update, we propose a simple solution which is based on “early update” and which can accommodate spurious ambiguity. The basic idea is to use the correct action sequence which was

---

<sup>3</sup> Following (Zhang and Nivre, 2012), we say the training algorithm is global if it optimizes the score of an entire action sequence. A local learner trains a classifier which distinguishes between single actions.

<sup>4</sup> As shown in (Goldberg and Nivre 2012), most transition-based dependency parsers (Nivre et al., 2003; Huang and Sagae 2010; Zhang and Clark 2008) ignores spurious ambiguity by using a static oracle which maps a dependency tree to a single action sequence.

---

**Features of (Goldberg and Elhadad, 2010)**

---

|   |   |
|---|---|
| for $p$ in $p_{i-1}, p_i, p_{i+1}$  | $w_p-vl_p, w_p-vr_p, t_p-vl_p,$<br>$t_p-vr_p, tlc_p, trc_p, wlc_p, wrc_p$                       |
| for $p$ in $p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}$  | $t_p-tlc_p, t_p-trc_p, t_p-tlc_p-trc_p$   |
| for $p, q, r$ in $(p_{i-2}, p_{i-1}, p_i), (p_{i-1}, p_i, p_{i+1}), (p_i, p_{i+1}, p_{i+2}), (p_{i+1}, p_{i+2}, p_{i+3})$ | $t_p-t_q-t_r, t_p-t_q-w_r$  |
| for $p, q$ in $(p_{i-1}, p_i)$  | $t_p-tlc_p-t_q, t_p-trc_p-t_q, t_p-tlc_p-w_q,$<br>$t_p-trc_p-w_q, t_p-w_q-tlc_q, t_p-w_q-trc_q$ |

---

Table 1: Feature templates for English dependency parsing.  $w_p$  denotes the head word of  $p$ ,  $t_p$  denotes the POS tag of  $w_p$ .  $vl_p/vr_p$  denotes the number  $p$ 's of left/right child.  $lc_p/rc_p$  denotes  $p$ 's leftmost/rightmost child.  $p_i$  denotes partial tree being considered.

pruned right at the step when all correct sequence falls off the beam (as  $C_1$  in figure 1).

Algorithm 2 shows the pseudo-code of the training procedure over one training sample  $(x, t)$ , a sentence-tree pair. Here we assume  $\mathcal{C}$  to be the set of all correct action sequences/sub-sequences. At step  $k$ , the algorithm constructs a correct action sequence  $\hat{y}$  of length  $k$  by extending those in  $\beta_{k-1}$  (line 3). It also checks whether  $\beta_k$  no longer contains any correct sequence. If so,  $\hat{y}$  together with  $\beta_k[0]$  are used for parameter update (line 5 ~ line 6). It can be easily verified that each update in line 6 is valid. Note that both “TOPC” and the operation in line 5 use  $\mathcal{C}$  to check whether an action sequence  $y$  is correct or not. This can be efficiently implemented (without explicitly enumerating  $\mathcal{C}$ ) by checking if each LEFT( $i$ )/RIGHT( $i$ ) in  $y$  are compatible with  $(x, t)$ :  $p_i$  already collected all its dependents according to  $t$ ;  $p_i$  is attached to the correct neighbor suggested by  $t$ .

## 5 Experiments

For English, we use PTB as our data set. We use the standard split for dependency parsing and the split used by (Ratnaparkhi, 1996) for POS tagging. Penn2Malt<sup>5</sup> is used to convert the bracketed structure into dependencies. For dependency parsing, POS tags of the training set are generated using 10-fold jack-knifing.

For Chinese, we use CTB 5.1 and the split suggested by (Duan et al., 2007) for both tagging and dependency parsing. We also use Penn2Malt and the head-finding rules of (Zhang and Clark 2008) to convert constituency trees into dependencies. For dependency parsing, we assume gold segmentation and POS tags for the input.

---

<sup>5</sup> <http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html>



Features used in English dependency parsing are listed in table 1. Besides the features in (Goldberg and Elhadad, 2010), we also include some trigram features and valency features which are useful for transition-based dependency parsing (Zhang and Nivre, 2011). For English POS tagging, we use the same features as in (Shen et al., 2007). For Chinese POS tagging and dependency parsing, we use the same features as (Ma et al., 2012). All of our experiments are conducted on a Core i7 (2.93GHz) machine, both the tagger and parser are implemented using C++.

### 5.1 Effect of beam width

Tagging/parsing performances with different beam widths on the development set are listed in table 2 and table 3. We can see that Chinese POS tagging, dependency parsing as well as English dependency parsing greatly benefit from beam search. While tagging accuracy on English only slightly improved. This may be because that the accuracy of the greedy baseline tagger is already very high and it is hard to get further improvement. Table 2 and table 3 also show that the speed of both tagging and dependency parsing drops linearly with the growth of beam width.

### 5.2 Final results

Tagging results on the test set together with some previous results are listed in table 4. Dependency parsing results on CTB and PTB are listed in table 5 and table 6, respectively.

On CTB, tagging accuracy of our greedy baseline is already comparable to the state-of-the-art. As the beam size grows to 5, tagging accuracy increases to 94.01% which is 2.3% error reduction. This is also the best tagging accuracy comparing with previous single tagging models (For limited space, we do not list the performance of joint tagging-parsing models).

Parsing performances on both PTB and CTB are significantly improved with a relatively small beam width ( $s = 8$ ). In particular, we achieve 86.33% uas on CTB which is 1.54% uas improvement over the greedy baseline parser. Moreover, the performance is better than the best transition-based parser (Zhang and Nivre, 2011) which adopts a much larger beam width ( $s = 64$ ).

## 6 Conclusion and related work

This work directly extends (Goldberg and Elhadad, 2010) with beam search and global learning. We show that both the easy-first POS tagger and dependency parser can be significantly impr-

| $s$ | PTB   | CTB   | speed |
|-----|-------|-------|-------|
| 1   | 97.17 | 93.91 | 1350  |
| 3   | 97.20 | 94.15 | 560   |
| 5   | 97.22 | 94.17 | 385   |

Table 2: Tagging accuracy vs beam width vs. Speed is evaluated using the number of sentences that can be processed in one second

| $s$ | PTB   |       | CTB   |       | speed |
|-----|-------|-------|-------|-------|-------|
|     | uas   | compl | uas   | compl |       |
| 1   | 91.77 | 45.29 | 84.54 | 33.75 | 221   |
| 2   | 92.29 | 46.28 | 85.11 | 34.62 | 124   |
| 4   | 92.50 | 46.82 | 85.62 | 37.11 | 71    |
| 8   | 92.74 | 48.12 | 86.00 | 35.87 | 39    |

Table 3: Parsing accuracy vs beam width. ‘uas’ and ‘compl’ denote unlabeled score and complete match rate respectively (all excluding punctuations).

| PTB                  |       | CTB                   |                    |
|----------------------|-------|-----------------------|--------------------|
| (Collins, 2002)      | 97.11 | (Hatori et al., 2012) | 93.82              |
| (Shen et al., 2007)  | 97.33 | (Li et al., 2012)     | 93.88              |
| (Huang et al., 2012) | 97.35 | (Ma et al., 2012)     | 93.84              |
| this work $s = 1$    | 97.22 | this work $s = 1$     | 93.87              |
| this work $s = 4$    | 97.28 | this work $s = 5$     | 94.01 <sup>†</sup> |

Table 4: Tagging results on the test set. ‘<sup>†</sup>’ denotes statistically significant over the greedy baseline by McNemar’s test ( $p < 0.05$ )

| Systems                 | $s$ | uas                | compl |
|-------------------------|-----|--------------------|-------|
| (Huang and Sagae, 2010) | 8   | 85.20              | 33.72 |
| (Zhang and Nivre, 2011) | 64  | 86.00              | 36.90 |
| (Li et al., 2012)       | —   | 86.55              | —     |
| this work               | 1   | 84.79              | 32.98 |
| this work               | 8   | 86.33 <sup>†</sup> | 36.13 |

Table 5: Parsing results on CTB test set.

| Systems                 | $s$ | uas                | compl |
|-------------------------|-----|--------------------|-------|
| (Huang and Sagae, 2010) | 8   | 92.10              | —     |
| (Zhang and Nivre, 2011) | 64  | 92.90              | 48.50 |
| (Koo and Collins, 2010) | —   | 93.04              | —     |
| this work               | 1   | 91.72              | 44.04 |
| this work               | 8   | 92.47 <sup>†</sup> | 46.07 |

Table 6: Parsing results on PTB test set.

oved using beam search and global learning.

This work can also be considered as applying (Huang et al., 2012) to the systems that exhibit spurious ambiguity. One future direction might be to apply the training method to transition-based parsers with dynamic oracle (Goldberg and Nivre, 2012) and potentially further advance performances of state-of-the-art transition-based parsers.

Shen et al., (2007) and (Shen and Joshi, 2008) also proposed bi-directional sequential classification with beam search for POS tagging and LTAG dependency parsing, respectively. The main difference is that their training method aims to learn a classifier which distinguishes between each local action while our training method aims to distinguish between action sequences. Our method can also be applied to their framework.

### Acknowledgments

We would like to thank Yue Zhang, Yoav Goldberg and Zhenghua Li for discussions and suggestions on earlier draft of this paper. We would also like to thank the three anonymous reviewers for their suggestions. This work was supported in part by the National Science Foundation of China (61073140; 61272376), Specialized Research Fund for the Doctoral Program of Higher Education (20100042110031) and the Fundamental Research Funds for the Central Universities (N100204002).

### References

- Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.
- Duan, X., Zhao, J., , and Xu, B. 2007. Probabilistic models for action-based Chinese dependency parsing. In *Proceedings of ECML/ECPPKDD*.
- Goldberg, Y. and Elhadad, M. 2010. An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing. In *Proceedings of NAACL*
- Huang, L. and Sagae, K. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL*.
- Huang, L. Fayong, S. and Guo, Y. 2012. Structured Perceptron with Inexact Search. In *Proceedings of NAACL*.
- Koo, T. and Collins, M. 2010. Efficient third-order dependency parsers. In *Proceedings of ACL*.
- Li, Z., Zhang, M., Che, W., Liu, T. and Chen, W. 2012. A Separately Passive-Aggressive Training Algorithm for Joint POS Tagging and Dependency Parsing. In *Proceedings of COLING*
- Ma, J., Xiao, T., Zhu, J. and Ren, F. 2012. Easy-First Chinese POS Tagging and Dependency Parsing. In *Proceedings of COLING*
- Rataparkhi, A. (1996) A Maximum Entropy Part-Of-Speech Tagger. In *Proceedings of EMNLP*
- Shen, L., Satt, G. and Joshi, A. K. (2007) Guided Learning for Bidirectional Sequence Classification. In *Proceedings of ACL*.
- Shen, L. and Josh, A. K. 2008. LTAG Dependency Parsing with Bidirectional Incremental Construction. In *Proceedings of EMNLP*.
- Seeker, W., Farkas, R. and Bohnet, B. 2012. Data-driven Dependency Parsing With Empty Heads. In *Proceedings of COLING*
- Søggard, A. and Wulff, J. 2012. An Empirical Study of Non-lexical Extensions to Delexicalized Transfer. In *Proceedings of COLING*
- Yue Zhang and Stephen Clark. 2007. Chinese Segmentation Using a Word-based Perceptron Algorithm. In *Proceedings of ACL*.
- Zhang, Y. and Clark, S. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL*.
- Zhang, Y. and Nivre, J. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL*.
- Zhang, Y. and Nivre, J. 2012. Analyzing the Effect of Global Learning and Beam-Search for Transition-Based Dependency Parsing. In *Proceedings of COLING*.

# Arguments and Modifiers from the Learner’s Perspective

Leon Bergen

MIT

Brain and Cognitive Science

bergen@mit.edu

Edward Gibson

MIT

Brain and Cognitive Science

egibson@mit.edu

Timothy J. O’Donnell

MIT

Brain and Cognitive Science

timod@mit.edu

## Abstract

We present a model for inducing sentential argument structure, which distinguishes arguments from optional modifiers. We use this model to study whether representing an argument/modifier distinction helps in learning argument structure, and whether a linguistically-natural argument/modifier distinction can be induced from distributional data alone. Our results provide evidence for both hypotheses.

## 1 Introduction

A fundamental challenge facing the language learner is to determine the content and structure of the stored units in the lexicon. This problem is made more difficult by the fact that many lexical units have *argument structure*. Consider the verb *put*. The sentence, *John put the socks* is incomplete; when hearing such an utterance, a speaker of English will expect a location to also be specified: *John put the socks in the drawer*. Facts such as these can be captured if the lexical entry for *put* also specifies that the verb has three required arguments: (i) who is doing the putting (ii) what is being put (iii) and the destination of the putting.

The problem of acquiring argument structure is further complicated by the fact that not all phrases in a sentence fill an argument role. Instead, many are *modifiers*. Consider the sentence *John put the socks in the drawer at 5 o’clock*. The phrase *at 5 o’clock* occurs here with the verb *put*, but it is not an argument. Removing this phrase does not change the core structure of the PUTTING event, nor is the sentence incomplete without this phrase.

The distinction between arguments and modifiers has a long history in traditional grammar and is leveraged in many modern theories of syntax (Haegeman, 1994; Steedman, 2001; Sag et al., 2003). Despite the ubiquity of the distinc-

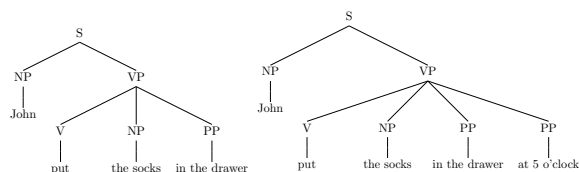


Figure 1: The VP’s in these sentences only share structure if we separate arguments from modifiers.

tion in syntax, however, there is a lack of consensus on the necessary and sufficient conditions for argumenthood (Schütze, 1995; Schütze and Gibson, 1999). It remains unclear whether the argument/modifier distinction is purely semantic or is also represented in syntax, whether it is binary or graded, and what effects argument/modifierhood have on the distribution of linguistic forms.

In this work, we take a new approach to these problems. We propose that the argument/modifier distinction is inferred on a phrase-by-phrase basis using probabilistic inference. Crucially, allowing the learner to separate the core argument structure of phrases from peripheral modifier content increases the generalizability of argument constructions. For example, the two sentences in Figure 1 intuitively share the same argument structures, but this overlap can only be identified if the prepositional phrase, “at 5 o’clock,” is treated as a modifier. Thus representing the argument/modifier distinction can help the learner find useful argument structures which generalize robustly.

Although, like the majority of theorists, we agree that the argument/adjunct distinction is fundamentally semantic, in this work we focus on its distributional correlates. Does the optionality of modifier phrases help the learner acquire lexical items with the right argument structure?

## 2 Approach

We adopt an approach where the lexicon consists of an inventory of stored tree fragments. These

tree fragments encode the necessary phrase types (i.e., arguments) that must be present in a structure before it is complete. In this system, sentences are generated by recursive *substitution* of tree fragments at the frontier argument nodes of other tree fragments. This approach extends work on learning probabilistic Tree-Substitution Grammars (TSGs) (Post and Gildea, 2009; Cohn et al., 2010; O’Donnell, 2011; O’Donnell et al., 2011).<sup>1</sup>

To model modification, we introduce a second structure-building operation, *adjunction*. While substitution must be licensed by the existence of an argument node, adjunction can insert constituents into well-formed trees. Many syntactic theories have made use of an adjunction operation to model modification. Here, we adopt the variant known as *sister-adjunction* (Rambow et al., 1995; Chiang and Bikel, 2002) which can insert a constituent as the sister to any node in an existing tree.

In order to derive the complete tree for a sentence, starting from an S root node, we recursively sample arguments and modifiers as follows.<sup>2</sup> For every nonterminal node on the frontier of our derivation, we sample an elementary tree from our lexicon to substitute into this node. As already noted, these elementary trees represent the argument structure of our tree. Then, for each argument nonterminal on the tree’s interior, we sister-adjoin one or more modifier nodes, which themselves are built by the same recursive process.

Figure 2 illustrates two derivations of the same tree, one in standard TSG without sister-adjunction, and one in our model. In the TSG derivation, at top, an elementary tree with four arguments – including the intuitively optional temporal PP – is used as the backbone for the derivation. The four phrases filling these arguments are then substituted into the elementary tree, as indicated by arrows. In the bottom derivation, which uses sister-adjunction, an elementary tree with only three arguments is used as the backbone. While the right-most temporal PP needed to be an argument of the elementary tree in the TSG derivation, the bottom derivation uses sister-adjunction to insert this PP as a child of the VP. Sister-adjunction therefore allows us to use an ar-

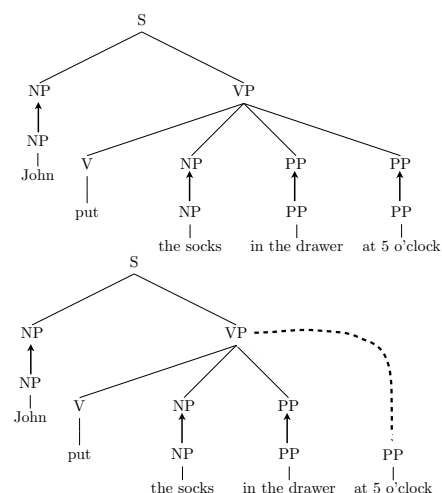


Figure 2: The first part of the figure shows how to derive the tree in TSG, while the second part shows how to use sister-adjunction to derive the same tree in our model.

gument structure that matches the true argument structure of the verb “put.”

This figure illustrates how derivations in our model can have a greater degree of generalizability than those in a standard TSG. Sister-adjunction will be used to derive children which are not part of the core argument structure, meaning that a greater variety of structures can be derived by a combination of common argument structures and sister-adjoined modifiers. Importantly, this makes the learning problem for our model less sparse than for TSGs; our model can derive the trees in a corpus using fewer types of elementary trees than a TSG. As a result, the distribution over these elementary trees is easier to estimate.

To understand what role modifiers play during learning, we will develop a learning model that can induce the lexicon and modifier contexts used by our generative model.

### 3 Model

Our model extends earlier work on induction of Bayesian TSGs (Post and Gildea, 2009; O’Donnell, 2011; Cohn et al., 2010). The model uses a Bayesian non-parametric distribution—the Pitman-Yor Process, to place a prior over the lexicon of elementary trees. This distribution allows the complexity of the lexicon to grow to arbitrary size with the input, while still enforcing a bias for more compact lexicons.

<sup>1</sup>Note that we depart from many discussions of argument structure in that we do not require that every stored fragment has a head word. In effect, we allow completely abstract phrasal constructions to also have argument structures.

<sup>2</sup>Our generative model is related to the generative model for Tree-Adjoining Grammars proposed in (Chiang, 2000)

For each nonterminal  $c$ , we define:

$$G_c|a_c, b_c, P_E \sim \text{PYP}(a_c, b_c, P_E(\cdot|c)) \quad (1)$$

$$e|c, G_c \sim G_c, \quad (2)$$

where  $P_E(\cdot|c)$  is a context free distribution over elementary trees rooted at  $c$ , and  $e$  is an elementary tree.

The context-free distribution over elementary trees  $P_E(e|c)$  is defined by:

$$P_E(e|c) = \prod_{i \in I(e)} (1-s_{c_i}) \prod_{f \in F(e)} s_{c_f} \prod_{c' \rightarrow \alpha \in e} P_{c'}(\alpha|c'), \quad (3)$$

where  $I(e)$  is the set of internal nodes in  $e$ ,  $F(e)$  is the set of frontier nodes,  $c_i$  is the nonterminal category associated with node  $i$ , and  $s_c$  is the probability that we stop expanding at a node  $c$ . For this paper, the parameters  $s_c$  are set to 0.5.

In addition to defining a distribution over elementary trees, we also define a distribution which governs modification via sister–adjunction. To sample a modifier, we first decide whether or not to sister–adjoin into location  $l$  in a tree. Following this step, we sample a modifier category (e.g., a PP) conditioned on the location  $l$ 's *context*: its parent and left siblings. Because contexts are sparse, we use a backoff scheme based on hierarchical Dirichlet processes similar to the ngram backoff schemes defined in (Teh, 2006; Goldwater et al., 2006). Let  $c$  be a nonterminal node in a tree derived by substitution into argument positions. The node  $c$  will have  $n \geq 1$  children derived by argument substitution:  $d_0, \dots, d_n$ . In order to sister–adjoin between two of these children  $d_i, d_{i+1}$ , we recursively sample nonterminals  $s_{i,1}, \dots, s_{i,k}$  until we hit a STOP symbol:

$$\begin{aligned} &P_a(s_{i,1}, \dots, s_{i,k}, \text{STOP}|C_0) \quad (4) \\ &= \prod_{j=1}^k P_a(s_{i,j}|C_j) \cdot (1 - P_{C_j}(\text{STOP})) \\ &\quad \cdot P_{C_{k+1}}(\text{STOP}) \end{aligned}$$

where  $C_j = d_1, s_{1,1}, \dots, d_i, s_{i,1}, \dots, s_{i,j-1}$ ,  $c$  is the context for the  $j$ 'th modifier between these children. The distribution over sister–adjoined nonterminals is defined using a hierarchical Dirichlet process to implement backoff in a prefix tree over contexts. We define the distribution  $G(q_l, \dots, q_1)$  over sister–adjoined nonterminals  $s_{i,j}$  given the context  $q_l, \dots, q_1$  by:

$$G(q_l, \dots, q_1) \sim \text{DP}(\alpha, G(q_{l-1}, \dots, q_1)). \quad (5)$$

The distribution  $G$  at the root of the hierarchy is not conditioned on any prior context. We define  $G$  by:

$$G \sim \text{DP}(\alpha, \text{Multinomial}(\mathbf{m})) \quad (6)$$

where  $\mathbf{m}$  is a vector with entries for each nonterminal, and where we sample  $\mathbf{m} \sim \text{Dir}(1, \dots, 1)$ .

To perform inference, we developed a local Gibbs sampler which generalizes the one proposed by (Cohn et al., 2010).

## 4 Results

We evaluate our model in two ways. First, we examine whether representing the argument/modifier distinction increases the ability of the model to learn highly generalizable elementary trees that can be used as argument structures across a variety of sentences. Second, we ask whether our model is able to induce the correct argument/modifier distinction according to a linguistic gold–standard. We trained our model on sections 2–21 of the WSJ part of the Penn Treebank (Marcus et al., 1999). The model was trained on the trees in this corpus, without any further annotations for substitution or modification.

To address the first question, we compared the structure of the grammar learned by our model to a grammar learned by a version of our model without sister–adjunction (i.e., a TSG similar to the one used in Cohn et al.). Our model should find more common structure among the trees in the input corpus, and therefore it should learn a set of elementary trees which are more complex and more widely shared across sentences. We evaluated this hypothesis by analyzing the average complexity of the most probable elementary trees learned by these models. As Table 1 shows, our model discovers elementary trees that have greater depth and more nodes than those found by the TSG. In addition, our model accounts for a larger portion of the corpus with fewer rules: the top 50, 100, and 200 most common elementary trees in our model's lexicon account for a greater portion of the corpus than the corresponding sets in the TSG.

Figure 3 illustrates a representative example from the corpus. By using sister–adjunction to separate the ADVP node from the rest of the sentence's derivation, our model was able to use a common depth-3 elementary tree to derive the backbone of the sentence. In contrast, the TSG cannot give the same derivation, as it needs to include the ADVP

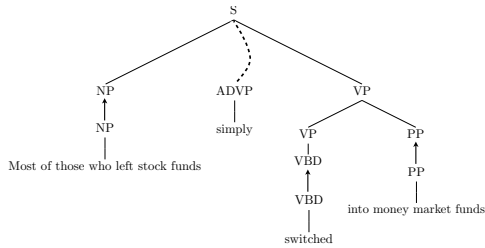


Figure 3: Part of a derivation found by our model.

| Model    | Rank | Avg tree depth | Avg tree size | #Tokens |
|----------|------|----------------|---------------|---------|
| Modifier | 50   | 1.59           | 3.42          | 97282   |
| TSG      | 50   | 1.38           | 2.98          | 88023   |
| Modifier | 100  | 1.84           | 3.98          | 134205  |
| TSG      | 100  | 1.58           | 3.38          | 116404  |
| Modifier | 200  | 1.97           | 4.27          | 170524  |
| TSG      | 200  | 1.77           | 3.84          | 146040  |

Table 1: This table shows the average depth and node count for elementary trees in our model and the TSG. The results are shown for the 50, 100, and 200 most frequent types of elementary trees.

node in the elementary tree; this wider elementary tree is much less common in the corpus.

We next examined whether our model learned to correctly identify modifiers in the corpus. Unfortunately, marking for argument/modifiers in the Penn Treebank is incomplete, and is limited to certain adverbials, e.g. locative and temporal PP’s. To supplement this markup, we made use of the corpus of (Kaeshammer and Demberg, 2012). This corpus adds annotations indicating, for each node in the Penn Treebank, whether that node is a modifier. This corpus was compiled by combining information from Propbank (Palmer et al., 2005) with a set of heuristics, as well as the NP-branching structures proposed in (Vadas and Curran, 2007). It is important to note that this corpus can only serve as a rough benchmark for evaluation of our model, as the heuristics used in its development did not always follow the correct linguistic analysis; the corpus was originally constructed for an alternative application in computational linguistics, for which non-linguistically-natural analyses were sometimes convenient. Our model was trained on this corpus, after it had been stripped of argument/modifier annotations.

We compare our model’s performance to a random baseline. Our model constrains every non-terminal to have at least one argument child, and our Gibbs sampler initializes argument/modifier choices randomly subject to this constraint. We

| Model    | Precision | Recall | #Guessed | #Correct |
|----------|-----------|--------|----------|----------|
| Random   | 0.27      | 0.19   | 298394   | 82702    |
| Modifier | 0.62      | 0.15   | 108382   | 67516    |

Table 2: This table shows precision and recall in identifying modifier nodes in the corpus.

therefore calculated the probability that a node that was randomly initialized as a modifier was in fact a modifier, i.e. the precision of random initialization. Next, we looked at the precision of our model following training. Table 2 shows that among nodes that were labeled as modifiers, 0.27 were labeled correctly before training and 0.62 were labeled correctly after. This table also shows the recall performance for our model decreased by 0.04. Some of this decrease is due to limitations of the gold standard; for example, our model learns to classify infinitives and auxiliary verbs as arguments — consistent with standard linguistic analyses — whereas the gold standard classifies these as modifiers. Future work will investigate how the metric used for evaluation can be improved.

## 5 Summary

We have investigated the role of the argument/modifier distinction in learning. We first looked at whether introducing this distinction helps in generalizing from an input corpus. Our model, which represents modification using sister-adjunction, learns a richer lexicon than a model without modification, and its lexicon provides a more compact representation of the input corpus. We next looked at whether the traditional linguistic classification of arguments and modifiers can be induced from distributional information. Without supervision from the correct labelings of modifiers, our model learned to identify modifiers more accurately than chance. This suggests that although the argument/modifier distinction is traditionally drawn without reference to distributional properties, the distributional correlates of this distinction are sufficient to partially reconstruct it from a corpus. Taken together, these results suggest that representing the difference between arguments and modifiers may make it easier to acquire a language’s argument structure.

## Acknowledgments

We thank Vera Demberg for providing the gold standard, and Tom Wasow for helpful comments.

## References

- David Chiang and Daniel Bikel. 2002. Recovering latent information in treebanks. In *Proceedings of COLING 2002*.
- David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Trevor Cohn, Phil Blunsom, and Sharon Goldwater. 2010. Inducing tree-substitution grammars. *Journal of Machine Learning Research*, 11:3053–3096.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, Cambridge, Ma. MIT Press.
- Liliane Haegeman. 1994. *Government & Binding Theory*. Blackwell.
- Mirian Kaeshammer and Vera Demberg. 2012. German and English treebanks and lexica for tree-adjoining grammars. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2012)*.
- Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3. Technical report, Linguistic Data Consortium, Philadelphia.
- Timothy J. O’Donnell, Jesse Snedeker, Joshua B. Tenenbaum, and Noah D. Goodman. 2011. Productivity and reuse in language. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.
- Timothy J. O’Donnell. 2011. *Productivity and Reuse in Language*. Ph.D. thesis, Harvard University.
- Martha Palmer, P. Kingsbury, and Daniel Gildea. 2005. The proposition bank. *Computational Linguistics*, 31(1):71–106.
- Matt Post and Daniel Gildea. 2009. Bayesian learning of a tree substitution grammar. 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*.
- Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-tree grammars. In *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI, Stanford, CA, 2 edition.
- Carson T Schütze and Edward Gibson. 1999. Argumenthood and english prepositional phrase attachment. *Journal of Memory and Language*, 40(3):409–431.
- Carson T. Schütze. 1995. PP attachment and argumenthood. Technical report, Papers on language processing and acquisition, MIT working papers in linguistics, Cambridge, Ma.
- Mark Steedman. 2001. *The syntactic process*. The MIT press.
- Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, National University of Singapore, School of Computing.
- David Vadas and James Curran. 2007. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

# Benefactive/Malefactive Event and Writer Attitude Annotation

Lingjia Deng <sup>†</sup>, Yoonjung Choi <sup>\*</sup>, Janyce Wiebe <sup>†\*</sup>

<sup>†</sup> Intelligent System Program, University of Pittsburgh

<sup>\*</sup> Department of Computer Science, University of Pittsburgh

<sup>†</sup>lid29@pitt.edu, <sup>\*</sup>{yjchoi, wiebe}@cs.pitt.edu

## Abstract

This paper presents an annotation scheme for events that negatively or positively affect entities (*benefactive/malefactive events*) and for the attitude of the writer toward their agents and objects. Work on opinion and sentiment tends to focus on explicit expressions of opinions. However, many attitudes are conveyed implicitly, and benefactive/malefactive events are important for inferring implicit attitudes. We describe an annotation scheme and give the results of an inter-annotator agreement study. The annotated corpus is available online.

## 1 Introduction

Work in NLP on opinion mining and sentiment analysis tends to focus on explicit expressions of opinions. Consider, however, the following sentence from the MPQA corpus (Wiebe et al., 2005) discussed by (Wilson and Wiebe, 2005):

- (1) I think people are happy because Chavez has fallen.

The explicit sentiment expression, *happy*, is positive. Yet (according to the writer), the people are *negative* toward Chavez. As noted by (Wilson and Wiebe, 2005), the attitude toward Chavez is inferred from the explicit sentiment toward the event. An opinion-mining system that recognizes only explicit sentiments would not be able to perceive the negative attitude toward Chavez conveyed in (1). Such inferences must be addressed for NLP systems to be able to recognize the full range of opinions conveyed in language.

The inferences arise from interactions between sentiment expressions and events such as *fallen*, which negatively affect entities (*malefactive events*), and events such as *help*, which positively affect entities (*benefactive events*). While some corpora have been annotated for explicit opinion expressions (for example, (Kessler et al., 2010; Wiebe et al., 2005)), there isn't a previously published corpus annotated for benefactive/malefactive events. While (Anand and Reschke, 2010) conducted a related annotation study, their data are artificially constructed sentences incorporating event predicates from a fixed list, and their annotations are of the writer's attitude toward the events. The scheme presented here is the first scheme for annotating, in naturally-occurring text, benefactive/malefactive events themselves as well as the writer's attitude toward the agents and objects of those events.

## 2 Overview

For ease of communication, we use the terms *goodFor* and *badFor* for benefactive and malefactive events, respectively, and use the abbreviation *gfbf* for an event that is one or the other. There are many varieties of gfbf events, including destruction (as in *kill Bill*, which is bad for Bill), creation (as in *bake a cake*, which is good for the cake), gain or loss (as in *increasing costs*, which is good for the costs), and benefit or injury (as in *comforted the child*, which is good for the child) (Anand and Reschke, 2010).

The scheme targets clear cases of gfbf events. The event must be representable as a triple of contiguous text spans,  $\langle agent, gfbf, object \rangle$ . The agent must be a noun phrase, or it may be *implicit* (as in *the constituent will be destroyed*). The object must be a noun phrase.



Another component of the scheme is the **influencer**, a word whose effect is to either retain or reverse the polarity of a gfbf event. For example:

- (2) Luckily Bill *didn't* **kill** him.
- (3) The reform *prevented* companies from **hurting** patients.
- (4) John *helped* Mary to **save** Bill.

In (2) and (3), *didn't* and *prevented*, respectively, reverse the polarity from badFor to goodFor (not killing Bill is good for Bill; preventing companies from hurting patients is good for the patients). In (4), *helped* is an influencer which retains the polarity (i.e., helping Mary to save Bill is good for Bill). Examples (3) and (4) illustrate the case where an influencer introduces an additional agent (*reform* in (3) and *John* in (4)).

The agent of an influencer must be a noun phrase or *implicit*. The object must be another influencer or a gfbf event.

Note that, semantically, an influencer can be seen as good for or bad for its object. A reverser influencer makes its object irrealis (i.e., not happen). Thus, it is bad for it. In (3), for example, *prevent* is bad for the *hurting* event. A retainer influencer maintains its object, and thus is good for it. In (4), for example, *helped* maintains the *saving* event. For this reason, influencers and gfbf events are sometimes combined in the evaluations presented below (see Section 4.2).

Finally, the annotators are asked to mark the writer's attitude towards the agents of the influencers and gfbf events and the objects of the gfbf events. For example:

- (5) **Attack on Reform Is a Fight Against** Justice.
- (6) **Jettison** any reference to end-of-life counselling.

In (5), there are two badFor events:  $\langle \text{GOP, Attack on, Reform} \rangle$  and  $\langle \text{GOP Attack on Reform, Fight Against, Justice} \rangle$ . The writer's attitude toward both agents is negative, and his or her attitude toward both objects is positive. In (6), the writer conveys a negative attitude toward *end-of-life counselling*. The coding manual instructs the annotators to consider whether an attitude of the writer is communicated or revealed in the particular sentence which contains the gfbf event.

### 3 Annotation Scheme

There are four types of annotations: gfbf event, influencer, agent, and object. For gfbf events, the agent, object, and polarity (goodFor or badFor) are identified. For influencers, the agent, object and effect (reverse or retain) are identified. For agents and objects, the writer's attitude is marked (positive, negative, or none). The annotator links agents and objects to their gfbf and influencer annotations via explicit IDs. When an agent is not mentioned explicitly, the annotator should indicate that it is *implicit*. For any span the annotator is not certain about, he or she can set the *uncertain* option to be true.

The annotation manual includes guidelines to help clarify which events should be annotated.

Though it often is, the gfbf span need not be a verb or verb phrase. We saw an example above, namely (5). Even though *attack on* and *fight against* are not verbs, we still mark them because they represent events that are bad for the object. Note that, Goyal et al. (2012) present a method for automatically generating a lexicon of what they call *patient polarity verbs*. Such verbs correspond to gfbf events, except that gfbf events are, conceptually, events, not verbs, and gfbf spans are not limited to verbs (as just noted).

Recall from Section 2 that annotators should only mark gfbf events that may be represented as a triple,  $\langle \text{agent, gfbf, object} \rangle$ . The relationship should be perceptible by looking only at the spans in the triple. If, for example, another argument of the verb is needed to perceive the relationship, the annotators should not mark that event.

- (7) His uncle **left** him *a massive amount of debt*.
- (8) His uncle **left** him *a treasure*.

There is no way to break these sentences into triples that follow our rules.  $\langle \text{His uncle, left, him} \rangle$  doesn't work because we cannot perceive the polarity looking only at the triple; the polarity depends on *what* his uncle left him.  $\langle \text{His uncle, left him, a massive amount of debt} \rangle$  isn't correct: the event is not bad for the debt, it is bad for *him*. Finally,  $\langle \text{His uncle, left him a massive amount of debt, Null} \rangle$  isn't correct, since no object is identified.

Note that *him* in (7) and (8) are both considered benefactive semantic roles (Zúñiga and Kitilá, 2010). In general, gfbf objects are not equiva-

lent to benefactive/malefactive semantic roles. For example, in our scheme, (7) is a badFor event and (8) is a goodFor event, while *him* fills the benefactive semantic role in both. Further, according to (Zúñiga and Kittilä, 2010), *me* is the filler of the benefactive role in *She baked a cake for me*. Yet, in our scheme, *a cake* is the object of the goodFor event; *me* is not included in the annotations. The objects of gfbf events are what (Zúñiga and Kittilä, 2010) refer to as the primary targets of the events, whereas, they state, beneficiary semantic roles are typically optional arguments. The reason we annotate only the primary objects (and agents) is that the clear cases of attitude implicatures motivating this work (see Section 1) are inferences toward agents and primary objects of gfbf events.

Turning to influencers, there may be chains of them, where the ultimate polarity and agent must be determined compositionally. For example, the structure of *Jack stopped Mary from trying to kill Bill* is a reverser influencer (*stopped*) whose object is a retainer influencer (*trying*) whose object is, in turn, a badFor event (*kill*). The ultimate polarity of this event is goodFor and the “highest level” agent is Jack. In our scheme, all such chains of length  $N$  are treated as  $N - 1$  influencers followed by a single gfbf event. It will be up to an automatic system to calculate the ultimate polarity and agent using rules such as those presented in, e.g., (Moilanen and Pulman, 2007; Neviarouskaya et al., 2010).

To save some effort, the annotators are not asked to mark retainer influencers which do not introduce new agents. For example, for *Jack stopped trying to kill Bill*, there is no need to mark “trying.” Of course, all reverser influencers must be marked.

## 4 Agreement Study

To validate the reliability of the annotation scheme, we conducted an agreement study. In this section we introduce how we designed the agreement study, present the evaluation method and give the agreement results. Besides, we conduct a second-step consensus study to further analyze the disagreement.

### 4.1 Data and Agreement Study Design

For this study, we want to use data that is rich in opinions and implicatures. Thus we used the corpus from (Conrad et al., 2012), which consists of 134 documents from blogs and editorials about a controversial topic, “the Affordable Care Act”.

To measure agreement on various aspects of the annotation scheme, two annotators, who are co-authors, participated in the agreement study; one of the two wasn’t involved in developing the scheme. The new annotator first read the annotation manual and discussed it with the first annotator. Then, the annotators labelled 6 documents and discussed their disagreements to reconcile their differences. For the formal agreement study, we randomly selected 15 documents, which have a total of 725 sentences. These documents do not contain any examples in the manual, and they are different from the documents discussed during training. The annotators then independently annotated the 15 selected documents.

### 4.2 Agreement Study Evaluation

We annotate four types of items (gfbf event, influencer, agent, and object) and their corresponding attributes. As noted above in Section 2, influencers can also be viewed as gfbf events. Also, the two may be combined together in chains. Thus, we measure agreement for gfbf and influencer spans together, treating them as one type. Then we choose the subset of gfbf and influencer annotations that both annotators identified, and measure agreement on the corresponding agents and objects.

Sometimes the annotations differ even though the annotators recognize the same gfbf event. Consider the following sentence:

(9) Obama **helped** reform **curb** costs.

Suppose the annotations given by the annotators were:

Ann 1. ⟨Obama, helped, curb⟩  
           ⟨reform, curb, costs⟩  
 Ann 2. ⟨Obama, helped, reform⟩

The two annotators do agree on the ⟨Obama, helped, reform⟩ triple, the first one marking *helped* as a retainer and the other marking it as a goodFor event. To take such cases into consideration in our evaluation of agreement, if two spans overlap and one is marked as gfbf and the other as influencer, we use the following rules to match up their agents and objects:

- for a gfbf event, consider its agent and object as annotated;

- for an influencer, assign the agent of the influencer’s object to be the influencer’s object, and consider its agent as annotated and the newly-assigned object. In (9), Ann 2’s annotations remain the same and Ann 1’s become  $\langle \textit{Obama, helped, reform} \rangle$  and  $\langle \textit{reform, curb, costs} \rangle$ .

We use the same measurement for agreement for all types of spans. Suppose  $A$  is a set of annotations of a particular type and  $B$  is the set of annotations of the same type from the other annotator. For any text span  $a \in A$  and  $b \in B$ , the span coverage  $c$  measures the overlap between  $a$  and  $b$ . Two measures of  $c$  are adopted here.

**Binary:** As in (Wilson and Wiebe, 2003), if two spans  $a$  and  $b$  overlap, the pair is counted as 1, otherwise 0.

$$c_1(a, b) = 1 \quad \textit{if} \quad |a \cap b| > 0$$

**Numerical:** (Johansson and Moschitti, 2013) propose, for the pairs that are counted as 1 by  $c_1$ , a measure of the percentage of overlapping tokens,

$$c_2(a, b) = \frac{|a \cap b|}{|b|}$$

where  $|a|$  is the number of tokens in span  $a$ , and  $\cap$  gives the tokens that two spans have in common. As (Breck et al., 2007) point out,  $c_2$  avoids the problem of  $c_1$ , namely that  $c_1$  does not penalize a span covering the whole sentence, so it potentially inflates the results.

Following (Wilson and Wiebe, 2003), treating each set  $A$  and  $B$  in turn as the gold-standard, we calculate the average F-measure, denoted  $agr(A, B)$ .  $agr(A, B)$  is calculated twice, once with  $c = c_1$  and once with  $c = c_2$ .

$$\begin{aligned} match(A, B) &= \sum_{\substack{a \in A, b \in B, \\ |a \cap b| > 0}} c(a, b) \\ agr(A||B) &= \frac{match(A, B)}{|B|} \\ agr(A, B) &= \frac{agr(A||B) + agr(B||A)}{2} \end{aligned}$$

Now that we have the sets of annotations on which the annotators agree, we use  $\kappa$  (Artstein and Poesio, 2008) to measure agreement for the attributes. We report two  $\kappa$  values: one for the polarities of the gfbf events, together with the effects of the influencers, and one for the writer’s

|                 |       | gfbf & influencer | agent | object |
|-----------------|-------|-------------------|-------|--------|
| all annotations | $c_1$ | 0.70              | 0.92  | 1.00   |
|                 | $c_2$ | 0.69              | 0.87  | 0.97   |
| only certain    | $c_1$ | 0.75              | 0.92  | 1.00   |
|                 | $c_2$ | 0.72              | 0.87  | 0.98   |
| consensus study | $c_1$ | 0.85              | 0.93  | 0.99   |
|                 | $c_2$ | 0.81              | 0.88  | 0.98   |

Table 1: Span overlapping agreement  $agr(A, B)$  in agreement study and consensus study.

|         | polarity & effect | attitude |
|---------|-------------------|----------|
| all     | 0.97              | 0.89     |
| certain | 0.97              | 0.89     |

Table 2:  $\kappa$  for attribute agreement.

attitude toward the agents and objects. Note that, as in Example (9), sometimes one annotator marks a span as gfbf and the other marks it as an influencer; in such cases we regard *retain* and *goodfor* as the same attribute value and *reverse* and *badfor* as the same value. Table 1 gives the  $agr$  values and Table 2 gives the  $\kappa$  values.

### 4.3 Agreement Study Results

Recall that the annotator could choose whether (s)he is certain about the annotation. Thus, we evaluate two sets: all annotations and only those annotations that both annotators are certain about. The results are shown in the top four rows in Table 1.

The results for agents and objects in Table 1 are all quite good, indicating that, given a gfbf or influencer, the annotators are able to correctly identify the agent and object.

Table 1 also shows that results are not significantly worse when measured using  $c_2$  rather than  $c_1$ . This suggests that, in general, the annotators have good agreement concerning the boundaries of spans.

Table 2 shows that the  $\kappa$  values are high for both sets of attributes.

### 4.4 Consensus Analysis

Following (Medlock and Briscoe, 2007), we examined what percentage of disagreement is due to negligence on behalf of one or the other annotator (i.e., cases of clear gfbfs or influencers that were missed), though we conducted our consensus

study in a more independent manner than face-to-face discussion between the annotators. For annotator *Ann1*, we highlighted sentences for which only *Ann2* marked a gfbf event, and gave *Ann1*'s annotations back to him or her with the highlights added on top. For *Ann2* we did the same thing. The annotators reconsidered their highlighted sentences, making any changes they felt they should, without communicating with each other. There could be more than one annotation in a highlighted sentence; the annotators were not told the specific number.

After re-annotating the highlighted sentences, we calculate the agreement score for all the annotations. As shown in the last two rows in Table 1, the agreement for gfbf and influencer annotations increases quite a bit. Similar to the claim in (Medlock and Briscoe, 2007), it is reasonable to conclude that the actual agreement is approximately lower bounded by the initial values and upper bounded by the consensus values, though, compared to face-to-face consensus, we provide a tighter upper bound.

## 5 Corpus and Examples

Recall from in Section 4.1 that we use the corpus from (Conrad et al., 2012), which consists of 134 documents with a total of 8,069 sentences from blogs and editorials about “the Affordable Care Act”. There are 1,762 gfbf and influencer annotations. On average, more than 20 percent of the sentences contain a gfbf event or an influencer. Out of all gfbf and influencer annotations, 40 percent are annotated as goodFor or retain and 60 percent are annotated as badFor or reverse. For agents and objects, 52 percent are annotated as positive and 47 percent as negative. Only 1 percent are annotated as none, showing that almost all the sentences (in this corpus of editorials and blogs) which contain gfbf annotations are subjective. The annotated corpus is available online<sup>1</sup>.

To illustrate various aspects of the annotation scheme, in this section we give several examples from the corpus. In the examples below, words in square brackets are agents or objects, words in italics are influencers, and words in boldface are gfbf events.

1. And [it] will *enable* [Obama and the Democrats] - who run Washington - to get

<sup>1</sup><http://mpqa.cs.pitt.edu/>

back to **creating** [jobs].

(a) *Creating* is goodFor *jobs*; the agent is *Obama and the Democrats*.

(b) The phrase *to get back to* is a retainer influencer. But, the agent span is also *Obama and the Democrats*, as the same with the goodFor, so we don't have to give an annotation for it.

(c) The phrase *enable* is a retainer influencer. Since its agent span is different (namely, *it*), we do create an annotation for it.

2. [**Repealing** [the Affordable Care Act]] would **hurt** [families, businesses, and our economy].

(a) *Repealing* is a badFor event since it deprives the object, *the Affordable Care Act*, of its existence. In this case the agent is *implicit*.

(b) The agent of the badFor event *hurt* is the whole phrase *Repealing the Affordable Care Act*. Note that the agent span is in fact a noun phrase (even though it refers to an event). Thus, it doesn't break the rule that all agent gfbf spans should be noun phrases.

3. It is a moral obligation to *end* this indefensible **neglect of** [hard-working Americans].

(a) This example illustrates a gfbf that centers on a noun (*neglect*) rather than on a verb.

(b) It also illustrates the case when two words can be seen as gfbf events: both *end* and *neglect of* can be seen as badFor events. Following our specification, they are annotated as a chain ending in a single gfbf event: *end* is an influencer that reverses the polarity of the badFor event *neglect of*.

## 6 Conclusion

Attitude inferences arise from interactions between sentiment expressions and benefactive/malefactive events. Corpora have been annotated in the past for explicit sentiment expressions; this paper fills in a gap by presenting an annotation scheme for benefactive/malefactive events and the writer's attitude toward the agents and objects of those events. We conducted an agreement study, the results of which are positive.

**Acknowledgement** This work was supported in part by DARPA-BAA-12-47 DEFT grant #12475008 and National Science Foundation grant #IIS-0916046. We would like to thank the anonymous reviewers for their helpful feedback.

## References

- Pranav Anand and Kevin Reschke. 2010. Verb classes as evaluativity functor classes. In *Interdisciplinary Workshop on Verbs. The Identification and Representation of Verb Features*.
- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596, December.
- Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artificial intelligence, IJCAI'07*, pages 2683–2688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Alexander Conrad, Janyce Wiebe, Hwa, and Rebecca. 2012. Recognizing arguing subjectivity and argument tags. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics, ExProM '12*, pages 80–88, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Amit Goyal, Ellen Riloff, and Hal Daum III. 2012. A computational model for plot units. *Computational Intelligence*, pages no–no.
- Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3).
- Jason S. Kessler, Miriam Eckert, Lyndsay Clark, and Nicolas Nicolov. 2010. The 2010 icwsm jdpa sentiment corpus for the automotive domain. In *4th Int'l AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*.
- Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.
- Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.
- Alena Neviarouskaya, Helmut Prendinger, and Mitsuuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 806–814, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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.
- Theresa Wilson and Janyce Wiebe. 2003. Annotating opinions in the world press. In *Proceedings of the 4th ACL SIGdial Workshop on Discourse and Dialogue (SIGdial-03)*, pages 13–22.
- Theresa Wilson and Janyce Wiebe. 2005. Annotating attributions and private states. In *Proceedings of ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*.
- F. Zúñiga and S. Kittilä. 2010. Introduction. In F. Zúñiga and S. Kittilä, editors, *Benefactives and malefactives*, Typological studies in language. J. Benjamins Publishing Company.

# GuiTAR-based Pronominal Anaphora Resolution in Bengali

**Apurbalal Senapati**

Indian Statistical Institute  
203, B.T.Road, Kolkata-700108, India  
apurbalal.senapati@gmail.com

**Utpal Garain**

Indian Statistical Institute  
203, B.T.Road, Kolkata-700108, India  
utpal.garain@gmail.com

## Abstract

This paper attempts to use an off-the-shelf anaphora resolution (AR) system for Bengali. The language specific preprocessing modules of GuiTAR (v3.0.3) are identified and suitably designed for Bengali. Anaphora resolution module is also modified or replaced in order to realize different configurations of GuiTAR. Performance of each configuration is evaluated and experiment shows that the off-the-shelf AR system can be effectively used for Indic languages.

## 1 Introduction

Little computational linguistics research has been done for anaphora resolution (AR) in Indic languages. Notable research efforts in this area are conducted by Shobha et al. (2000), Prasad et al. (2000), Jain et al. (2004), Agrawal et al. (2007), Uppalapu et al. (2009). These works address AR problem in language like Hindi, some South Indian languages including Tamil. Dhar et al. (2008) reported a research on Bengali. Progress of the research through these works was difficult to quantify as most of the authors used their self-generated datasets and in some cases algorithms lack in required details to make them reproducible.

First rigorous effort was taken in ICON 2011 (ICON 2011) where a shared task was conducted on AR in three Indic languages (Hindi, Bengali, and Tamil). Training and test datasets were provided, both the machine learning (WEKA and SVM based classification) and rule-based approaches are used and the participating systems (4 teams for Bengali, and 2 teams each for Hindi and Tamil) were evaluated using five different metrics (MUC, B<sup>3</sup>, CEAFM, CEAFE, and BLANC). However, no team attempted to reuse any of the off-the-shelf AR systems. This paper aims to explore this issue to investigate how far useful such a system is for AR in Indic languages. Bengali has been taken as the reference

language and GuiTAR (Poesio, 2004) has been considered as the reference off-the-shelf system.

GuiTAR is primarily designed for English language and therefore, its direct application for Bengali is not possible for grammatical variations and resource limitations. Therefore, the central contribution of this paper is to develop required resources for Bengali and thereby providing them to GuiTAR for anaphora resolution. Our contribution also includes extension of the ICON2011 AR dataset for Bengali so that evaluation could be done on a bigger sized dataset. Finally, GuiTAR anaphora resolution module is replaced by a previously developed approach (which is primarily rule-based, Senapati, 2011; Senapati, 2012a) and performances of different configurations are compared.

## 2 Language specific issues in GuiTAR

GuiTAR has two major modules namely, preprocessing and anaphora resolution (Kabadjov, 2007). In both of these modules modifications are required to fit it to Bengali. Let's first identify the components in both of these two modules where replacement/modifications are needed.

**Pre-processing:** The purpose of this module is to make GuiTAR independent from input format specifications and variations. It takes as input in XML or text format. In case of text input, XML file generated by the LT-XML tool. The XML file contains the information like word boundaries (tokens), grammatical classes (part-of-speech), and chunking information. From the XML format MAS-XML (Minimum Anaphoric Syntax - XML) is produced to include minimal information namely, noun phrase boundaries, utterance boundaries, categories of pronoun, number information, gender information, etc. All these aspects are to be addressed for Bengali so that for a given input discourse in Bengali, MAS-XML file can be generated correctly. Next section explains how this issue.

**Anaphora resolution:** The GuiTAR system resolves four types of anaphoras. The *pronouns* (*personal and possessive*) are resolved by using

an implementation of MARS (Mitkov, 2002), whereas different algorithms are used for resolving *definite descriptions*, and *proper nouns*. In Mitkov’s algorithm whenever a *pronoun* is to be resolved, it finds a list of potential antecedents within a given ‘window’ and checks three types of syntactic agreements (i.e., person, number and gender) between an antecedent and the pronoun. In case of more than one potential antecedent exists in the list it would be recursively filtered applying sequentially five different antecedent indicators (aggregate score, immediate reference, collocational pattern, indicating verbs and referential distance) until there is only one element in the list, i.e., the selected antecedent. We introduce suitable modifications in this module so that the same implementation of MARS can work for Bengali. This is explained in Sec. 4.

### 3 Bengali NLP Resources

Pronouns in Bengali has been studied before (linguistically by Majumdar, 2000; Sengupta, 2000 and for computational linguistics: Senapati, 2012a). Table-1 categorizes all pronouns (522 in number) available in Bengali as observed in a corpus (Bengali corpus, undated) of 35 million words.

| Category                        | Permissible Pronouns              |
|---------------------------------|-----------------------------------|
| Honorific Singular              | তাঁর,তাঁকে, তিনি, তাঁরই, তিনিই,.. |
| Honorific Plural                | তাঁরা, তাঁরাই, যাঁরা, উনারা,..    |
| 1 <sup>st</sup> Person Singular | আমি, আমাকে, মোর,..                |
| 1 <sup>st</sup> Person Plural   | আমরা, আমাদেরকে, মোদের,..          |
| 2 <sup>nd</sup> Person Singular | তোর, তোমার, আপনার,..              |
| 2 <sup>nd</sup> Person Plural   | তোরা, তোমরা, আপনারা,..            |
| 3 <sup>rd</sup> Person Singular | এ, এর, ও, সে, তারও, তার,..        |
| 3 <sup>rd</sup> Person Plural   | এরা, ওরা, তারা, তাদের,..          |
| Reflexive Pronoun               | নিজে, নিজেই, নিজেকে, নিজের,..     |

Table 1: Language resource

#### 3.1 Number Acquisition for Nouns

In Bengali, a set of nominal suffixes (Bhattacharya, 1993) (inflections and classifier) are used to recognize the number (singular/plural) of noun. To identify the number of a noun, we check whether any of the nominal suffixes (indicating plurality) are attached with the noun. If found, the number of the noun is tagged

as plural. From the corpus, we identified 17 such suffixes (e.g. *দের* /*der*, *রা* /*ra*, *দিগের* /*diger*, *দিগকে* /*digke*, *গুলি* /*guli*, etc.) which are used for number acquisition for nouns.

#### 3.2 Honorificity of Nouns

The honorific agreement exists in Bengali. Honorificity of a noun is indicated by a word or expression with connotations conveying esteem or respect when used in addressing or referring to a person. In Bengali three degree of honorificity are observed for the second person and two for the third person (Majumdar, 2000; Sengupta, 2000). The second and third person pronouns have distinct forms for different degrees of honorificity. Honorificity information is applicable for proper nouns (person) and nouns indicating relations like father, mother, teacher, etc.

The honorificity information is identified by maintaining a list of terms which can be considered as honorific addressing terms (e.g. *ভদ্রলোক*/*bhadrolok*, *বাবু*/*babu*, *ডঃ*/*Dr.*, *মহাশয়*/*mohashoy*, *ডা.*/*Dr.*, etc.). About 20 such terms are there in the list and we get these terms from analysis of the Bengali corpus. When these terms are used to add honorificity of a noun they appear either before or after the noun. Another additional way for identifying the honorificity information is to look at the inflection of the main verb which is inflected with *ন*/*n* (i.e. *বলেন*/*bolen*, *করেন*/*koren* etc.).

Honorificity is extracted during the preprocessing phase and added with the attribute *hon* = *<value>*. The value is set ‘*sup*’ (superior i.e. highest degree of honor), ‘*neu*’ (neutral i.e. medium degree of honor) or ‘*inf*’ (inferior i.e. lowest degree of honor) based on their degree of honorificity. For pronouns, this information is available from the pronoun list (honorific singular and honorific plural) as shown in Table-1.

### 4 GuiTAR for Bengali

The following sections explain the modifications needed to configure GuiTAR for Bengali.

#### 4.1 GuiTAR Preprocessing for Bengali

For getting part-of-speech information, the Stanford POS tagger has been retrained for Bengali language. The tagger is trained with about tagged 10,000 sentences and is found to produce about 92% accuracy while tested on 2,000 sentences. A rule based Bengali chunker (De, 2011) is used to get chunking information. NEIs and their classes (person, location, and organization) are tagged

manually (we did not get any Bengali NEI tool). After adding all these information, the input text is formatted into GuiTAR specified input XML file and is converted into MAS-XML. This file contains other syntactic information: person, types of pronouns, number and honorificity. Information on person and types of pronouns comes from Table-1. Number and honorificity are identified as explained before. Gender information has little role in Bengali anaphora resolution and hence is not considered. Types of pronouns are taken from Table-1.

#### 4.2 GuiTAR-based Pronoun Resolution for Bengali

GuiTAR resolves pronouns using MARS approach (Mitkov, 2002) that makes use of several agreements (based on person, number and gender). Certain changes are required here as gender agreement has no role. This agreement has been replaced by the honorific agreement. Moreover, the way pronouns are divided in MARS implementation is not always relevant for Bengali pronouns. For example, we do not differentiate between personal and possessive pronouns but they are separately treated in MARS. In our case, we have only considered the personal and reflexive pronouns while applying MARS based implementation for anaphora resolution.

In case of more than one antecedent found, GuiTAR resolves it by using five antecedent indicators namely, aggregate score, immediate reference, collocational pattern, indicating verbs and referential distance. For Bengali, the indicating verb indicator has no role in filtering the antecedents and hence removed.

### 5 Data and data format

To evaluate the configured GuiTAR system the dataset provided by ICON 2011 (ICON 2011) has been used. They provided annotated data (POS tagged, chunked and name entity tagged) for three Indian languages including Bengali. The annotated data is represented by a column format. Figure 1 shows a sample of the annotated data and the details description of the data is given in Table - 2.

|            |   |   |       |     |       |   |      |
|------------|---|---|-------|-----|-------|---|------|
| story2.txt | 0 | 0 | সবশেষ | NN  | B-NP  | o | -    |
| story2.txt | 0 | 1 | তার   | PRP | B-NP  | o | (13) |
| story2.txt | 0 | 2 | মলে   | NN  | B-NP  | o | -    |
| story2.txt | 0 | 3 | হলো   | VM  | B-VGF | o | -    |
| story2.txt | 0 | 4 | এদিকে | NN  | B-NP  | o | -    |
| story2.txt | 0 | 5 | আর    | QF  | B-NP  | o | -    |

Figure 2. ICON 2011 data format.

We have changed this format into GuiTAR specified XML format and finally checked/corrected manually. GuiTAR Preprocessor converts this XML into MAS-XML which looks like something as shown in Figure 3.

```
<s id="s81">
<ne gId="nv380" id="ne237">
<W Lpos="NN">সবশেষ</W></ne>
<ne gId="nv381" id="ne238" hon="neu"
AAcat="pers-pro" AAPER="per3" AAnum="sing">
<nphead id="AAh54"><W Lpos="PRP">তার</W>
</nphead></ne>
<ne gId="nv382" id="ne239">
<W Lpos="NN">মলে</W></ne>
.....
```

Figure 3. Sample GuiTAR MAS-XML file for Bengali text.

| Column | Type         | Description                           |
|--------|--------------|---------------------------------------|
| 1      | Document Id  | Contains the file-name                |
| 2      | Part number  | File are divided into part numbered   |
| 3      | Word number  | Word index in the sentence            |
| 4      | Word         | Word itself                           |
| 4      | POS          | POS of the word                       |
| 5      | Chunking     | Chunking information using IOB format |
| 6      | NE tags      | Name Entity Information is given      |
| 7      | Description  | Description                           |
| 8      | Co-reference | Co-reference information              |

Table 2: Description of ICON 2011 data format

The ICON 2011 data contains nine texts from different domains (Tourism, Story, News article, Sports). We have extended this dataset by adding four more texts in the same format. Among these four pieces, three are short stories and one is taken from newspaper articles. Table 3 shows the distribution of pronouns in the whole test data set for Bengali.

| Data       | ICON2011 | Extended |
|------------|----------|----------|
| #text      | 9        | 4        |
| #words     | 22,531   | 4,923    |
| #pronouns  | 1,325    | 322      |
| #anaphoric | 1,019    | 253      |

Table 3: Coverage of ICON 2011 dataset



## 6 Evaluation

The modified GuiTAR system has been evaluated by the dataset as described above. The dataset contains 1647 pronouns out of them 706 are personal pronouns (including reflexive pronouns). As the MARS in GuiTAR resolves only personal pronouns, we have used only these personal pronouns for evaluation. Three different systems are configured as described below:

System-1 (Baseline): A baseline system is configured by considering the most recent noun phrase as the referent of a pronoun (the first noun phrase in the backward direction is the antecedent of a pronoun).

System-2 (GuiTAR with MARS): In this configuration, GuiTAR is used with the modifications (as described in Sec. 4.1) in its preprocessing module and the modified MARS (as described in Sec. 4.2) is used for pronominal anaphora resolution (PAR).

System-3 (GuiTAR with new a PAR module): Under this configuration, GuiTAR is used with the modifications (as described in Sec. 4.1) in its pre-processing module but MARS is replaced by a previously developed system (Senapati, 2011; Senapati, 2012a) for pronominal anaphora resolution in Bengali. This is basically a rule-based system. For every noun phrase (i.e. a possible antecedent) the method first maintains a list of possible pronouns which the antecedent could attach with (note that any noun phrase cannot be referred by any pronoun). On encountering a pronoun, the method searches for the antecedents for which the pronoun is in the respective pronoun-lists. If there is more than one such antecedent, a set of rules is applied to resolve. The approach for applying the rules is similar to the one proposed by Baldwin (1997).

The evaluation has used five metrics namely, MUC,  $B^3$ , CEAFM, CEAFE and BLANC. The experimental results are reported in Table 4. Results show that GuiTAR with MARS gives better result than the situation where the most recent antecedent is picked (i.e. the baseline system). This improvement is statistically significant ( $p < 0.03$  in a two-sided t-test). When MARS is replaced by system-3, further improvement is achieved which is also statistically significant ( $p < 0.01$ ).

### 6.1 Error analysis

Analysis of errors shows that errors in number acquisition and identification of the honorificity are two major errors during preprocessing phase.

These errors propagate and result in further errors during resolution. Resolution process itself introduces some new errors. For example, some Bengali personal pronouns are ambiguous (sometimes they are anaphoric whereas in other cases they may appear as non-anaphoric too).  $\text{ভার/tar}$ ,  $\text{সে/se}$  are two examples of such pronouns in Bengali (Senapati, 2012b) and the present resolution system is not able to resolve such cases.

| System | Metric | System-1<br>(Baseline) | GuiTAR             |          |
|--------|--------|------------------------|--------------------|----------|
|        |        |                        | System-2<br>(MARS) | System-3 |
| MUC    | P      | 0.453                  | 0.516              | 0.538    |
|        | R      | 0.550                  | 0.536              | 0.579    |
|        | F1     | 0.497                  | 0.526              | 0.558    |
| $B^3$  | P      | 0.766                  | 0.828              | 0.921    |
|        | R      | 0.771                  | 0.824              | 0.911    |
|        | F1     | 0.769                  | 0.826              | 0.916    |
| CEAFM  | P      | 0.785                  | 0.800              | 0.885    |
|        | R      | 0.632                  | 0.622              | 0.784    |
|        | F1     | 0.700                  | 0.700              | 0.832    |
| CEAFE  | P      | 0.797                  | 0.825              | 0.921    |
|        | R      | 0.552                  | 0.571              | 0.731    |
|        | F1     | 0.652                  | 0.675              | 0.815    |
| BLANC  | P      | 0.688                  | 0.700              | 0.732    |
|        | R      | 0.735                  | 0.736              | 0.741    |
|        | F1     | 0.711                  | 0.718              | 0.736    |
| Avg.   | F1     | 0.666                  | 0.689              | 0.771    |

Table 4: Experimental results

## 7 Conclusion

The present experiment shows that GuiTAR which is one of the off-the-shelf anaphora resolution systems can be effectively configured for Bengali. Basic NLP information required by GuiTAR pre-processing module has been supplied mostly through automatic tools. A suitable tool is needed for NEI in Bengali. This can be explored in future. It is also revealed that MARS based implementation in GuiTAR is not very suitable for Bengali because the antecedent indicators used by MARS are probably not very effective for Bengali. Suitably designed rule based system could produce better result as shown in the experiment. Addition of other resolution algorithms is definitely a future extension of this study. Resolution of non-personal pronouns (which were not considered here) would be addressed next. In future, the similar experiment can be easily extended to other Indic languages (especially for Hindi and Tamil for which annotated data is available).

## References

- Agarwal, S., Srivastava, M., Agarwal, P., Sanyal, R. 2007. Anaphora Resolution in Hindi Documents, in Proc. Natural Language Processing and Knowledge Engineering (IEEE NLP-KE), Beijing, China.
- Baldwin, B. 1997. *CogNIAC: high precision coreference with limited knowledge and linguistic resources*, In ACL/EACL workshop on Operational factors in practical, robust anaphora resolution, pages 38- 45, Madrid, Spain.
- Bengali Corpus. *TDIL Corpus in Unicode*, <http://www.isical.ac.in/~lru/downloadCorpus.html>.
- Bhattacharya, T. and Dasgupta, P. 1993. *Classifiers, word order and definiteness in Bengali*, In Proceedings of the Seminar on Word Order. Osmania University, Hyderabad, India.
- Dhar, A. and Garain, U. 2008. *A method for pronominal anaphora resolution in Bengali*, In Proc. of 6th Int. Conf. on Natural Language Processing (ICON), Student paper competition section, Pune, India.
- De, S., Dhar, A., Biswas, S. and Garain, U. 2011. *On Development and Evaluation of a Chunker for Bangla*, In Proc. 2nd Int. Conf. on Emerging Applications of Information Technology (EAIT), pp. 321-324, Kolkata, India.
- ICON. 2011. NLP Tools Contest: Anaphora Resolution in Indian Languages, In 9th Int. Conf. on Natural Language Processing (ICON), Chennai, India.
- Jain, P., M R. Mital, S. Kumar, A. Mukerjee, and A. M. Raina. 2004. *Anaphora Resolution in Multi-Person Dialogues*, in Proc. 5th SIGdial Workshop on Discourse and Dialogue, Cambridge, Massachusetts, USA.
- Kabadjov, M.A. 2007. *A Comprehensive Evaluation of Anaphora Resolution and Discourse-new Classification*, PhD thesis, Department of Computer Science, University of Essex.
- Majumdar, A. 2000. *Studies in the Anaphoric Relations in Bengali*, Publisher: Subarnarekha, India.
- Mitkov, R. 2002. *Anaphora Resolution*. Longman.
- Poesio, M. and Kabadjov, M.A. 2004. *A General-Purpose, off-the-shelf Anaphora Resolution Module: Implementation and Preliminary Evaluation*, in LREC 2004.
- Prasad, R. and Strube, M. 2000. *Discourse salience and pronoun resolution in Hindi*, in Penn Working Papers in Linguistics, pp. 189-208.
- Sengupta, G. 2000. *Lexical anaphors and pronouns in Bnagla*, Lexical Anaphors and Pronouns in Selected South Asian Languages: A Principled Typology (Eds. B.C. Lust, K. Wali, J.W. Gair, and K.V. Subbarao), pp. 277-280, Publisher: Mouton de Gruyter, Berlin, New York.
- Senapati, A. and Garain, U. 2011. *Anaphora Resolution System for Bengali by Pronoun Emitting Approach*, in Proc. NLP Tool Contest, 9th Int. Conf. on Natural Language Processing (ICON), Chennai, India.
- Senapati, A. and Garain, U. 2012a. *Anaphora Resolution in Bengali using global discourse knowledge*, In Int. Conf. of Asian Language Processing (IALP), Hanoi, Vietnam.
- Senapati, A. and Garain, U. 2012b. *Identification of Anaphoric tAr (তঁর) and se (সে) in Bengali*, In Proc. 34<sup>th</sup> All India Conference of Linguists (AICL), Shillong, India.
- Sobha, L. and Patnaik, B.N.Patnaik. 2000. *Vasisth: An Anaphora Resolution System For Indian Languages*, In Proc. Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications (ACIDCA), Monastir, Tunisia.
- Uppalapu, B. and Sharma, D.M. (2009). *Pronoun Resolution For Hindi*, in DAARC 2009.

# A Decade of Automatic Content Evaluation of News Summaries: Reassessing the State of the Art

**Peter A. Rankel**  
University of Maryland  
rankel@math.umd.edu

**John M. Conroy**  
IDA / Center for Computing Sciences  
conroy@super.org

**Hoa Trang Dang**  
National Institute of Standards and Technology  
hoa.dang@nist.gov

**Ani Nenkova**  
University of Pennsylvania  
nenkova@seas.upenn.edu

## Abstract

How good are automatic content metrics for news summary evaluation? Here we provide a detailed answer to this question, with a particular focus on assessing the ability of automatic evaluations to identify statistically significant differences present in manual evaluation of content. Using four years of data from the Text Analysis Conference, we analyze the performance of eight ROUGE variants in terms of accuracy, precision and recall in finding significantly different systems. Our experiments show that some of the neglected variants of ROUGE, based on higher order  $n$ -grams and syntactic dependencies, are most accurate across the years; the commonly used ROUGE-1 scores find too many significant differences between systems which manual evaluation would deem comparable. We also test combinations of ROUGE variants and find that they considerably improve the accuracy of automatic prediction.

## 1 Introduction

ROUGE (Lin, 2004) is a suite of automatic evaluations for summarization and was introduced a decade ago as a reasonable substitute for costly and slow human evaluation. The scores it produces are based on  $n$ -gram or syntactic overlap between an automatic summary and a set of human reference summaries. However, the field does not have a good grasp of which of the many evaluation scores is most accurate in replicating human judgements. This state of uncertainty has led to problems in comparing published work, as differ-

ent researchers choose to publish different variants of scores.

In this paper we reassess the strengths of ROUGE variants using the data from four years of Text Analysis Conference (TAC) evaluations, 2008 to 2011. To assess the performance of the automatic evaluations, we focus on determining statistical significance<sup>1</sup> between systems, where the gold-standard comes from comparing the systems using manual pyramid and responsiveness evaluations. In this setting, computing correlation coefficients between manual and automatic scores is not applicable as it does not take into account the statistical significance of the differences nor does it allow the use of more powerful statistical tests which use pairwise comparisons of performance on individual document sets. Instead, we report on the accuracy of decisions on pairs of systems, as well as the precision and recall of identifying pairs of systems which exhibit statistically significant differences in content selection performance.

## 2 Background

During 2008–2011, automatic summarization systems at TAC were required to create 100-word summaries. Each year there were two multi-document summarization sub-tasks, the initial summary and the update summary, usually referred to as task A and task B, respectively. The test inputs in each consisted of about 10 documents and the type of summary varied between query-focused and guided. There are between 44 and 48 test inputs on which systems are compared for each task.

In 2008 and 2009, task A was to produce a

<sup>1</sup>For the purpose of this study, we define a difference as significant when the test statistic attains a value corresponding to a  $p$ -value less than 0.05.

query-focused summary in response to a user information need stated both as a brief statement and a paragraph-long description of the information the user seeks to find. In 2010 and 2011 task A was “guided summarization”, where the test inputs came from a small set of predefined domains. These domains included accidents and natural disasters, attacks, health and safety, endangered resources, investigations and trials. Systems were provided with a list of important aspects of information for each domain and were asked to cover as many of these aspects as possible. The writers of the reference summaries for evaluation were given similar instructions. In all four years, task B was to produce an update summary for each of the inputs given in task A (query-focused or guided). In each case, a new, subsequent set of documents related to the topic of the respective test set for task A was provided to the system. The task was to generate an update summary aimed at a user who has already read all documents in the inputs for task A.

The two manual evaluation approaches used in TAC 2008–2011 are modified pyramid (Nenkova et al., 2007) and overall responsiveness. The pyramid method requires several reference summaries for each input. These are manually analyzed to discover content units based on meaning rather than specific wording. Each content unit is assigned a weight equal to the number of reference summaries that included that content unit. The modified pyramid score is defined as the sum of weights of the content units in the summary normalized by the weight of an ideally informative summary which expresses  $n$  content units, where  $n$  is equal to the average of content units in the reference summaries. Responsiveness, on the other hand, is based on direct human judgements, without the need for reference summaries. Assessors are presented with a statement of the user’s information need and the summary they need to evaluate. Then they rate how well they think the summary responds to the information need contained in the topic statement. Responsiveness was rated on a ten-point scale in 2009, and on a five-point scale in all other years.

For each sub-task during 2008–2011, we analyze the performance of only the top 30 systems, which roughly corresponds to the systems that performed better than or around the median according to each manual metric. Table 1 gives the number

of significant differences among the top 30 participating systems. We keep only the best performing systems for the analysis because we are interested in studying how well automatic evaluation metrics can correctly compare very good systems.

| Year | Pyr A | Pyr B | Resp A | Resp B |
|------|-------|-------|--------|--------|
| 2008 | 82    | 109   | 68     | 105    |
| 2009 | 146   | 190   | 106    | 92     |
| 2010 | 165   | 139   | 150    | 128    |
| 2011 | 39    | 83    | 5      | 11     |

Table 1: Number of pairs of significantly different systems among the top 30 across the years. There is a total of 435 pairs in each year.

### 3 Which ROUGE is best?

In this section, we study the performance of several ROUGE variants, including ROUGE- $n$ , for  $n = 1, 2, 3, 4$ , ROUGE-L, ROUGE-W-1.2, ROUGE-SU4, and ROUGE-BE-HM (Hovy et al., 2006). ROUGE- $n$  measures the  $n$ -gram recall of the evaluated summary compared to the available reference summaries. ROUGE-L is the ratio of the number of words in the longest common subsequence between the reference and the evaluated summary and the number of words in the reference. ROUGE-W-1.2 is a weighted version of ROUGE-L. ROUGE-SU4 is a combination of skip bigrams and unigrams, where the skip bigrams are formed for all words that appear in the text with no more than four intervening words in between. ROUGE-BE-HM computes recall of dependency syntactic relations between the summary and the reference.

To evaluate how well an automatic evaluation metric reproduces human judgments, we use prediction *accuracy* similar to Owczarzak et al. (2012). For each pair of systems in each subtask, we compare the results of two Wilcoxon signed-rank tests, one using the manual evaluation scores for each system and one using the automatic evaluation scores for each system (Rankel et al., 2011).<sup>2</sup> The accuracy then is simply the percent agreement between the results of these two tests.

<sup>2</sup>We use the Wilcoxon test as it was demonstrated by Rankel et al. (2011) to give more statistical power than unpaired tests. As reported by Yeh (2000), other tests such as randomized testing, may also be appropriate. There is considerable variation in system performance for different inputs (Nenkova and Louis, 2008) and paired tests remove the effect of the input.

| Metric  | Responsiveness |      |      |      | Pyramid     |      |      |      |
|---------|----------------|------|------|------|-------------|------|------|------|
|         | Acc            | P    | R    | BA   | Acc         | P    | R    | BA   |
| R1      | 0.58 (0.61)    | 0.24 | 0.64 | 0.57 | 0.62 (0.66) | 0.37 | 0.67 | 0.61 |
| R2      | 0.64 (0.63)    | 0.28 | 0.60 | 0.59 | 0.68 (0.69) | 0.43 | 0.63 | 0.64 |
| R3      | 0.70 (0.63)    | 0.31 | 0.48 | 0.60 | 0.73 (0.68) | 0.49 | 0.53 | 0.66 |
| R4      | 0.73 (0.64)    | 0.33 | 0.40 | 0.60 | 0.74 (0.65) | 0.50 | 0.45 | 0.65 |
| RL      | 0.50 (0.59)    | 0.20 | 0.56 | 0.54 | 0.54 (0.63) | 0.29 | 0.60 | 0.55 |
| R-SU4   | 0.61(0.62)     | 0.26 | 0.61 | 0.58 | 0.65 (0.68) | 0.40 | 0.65 | 0.63 |
| R-W-1.2 | 0.52(0.62)     | 0.21 | 0.54 | 0.55 | 0.57(0.64)  | 0.32 | 0.62 | 0.57 |
| R-BE-HM | 0.70 (0.63)    | 0.30 | 0.49 | 0.59 | 0.74(0.68)  | 0.49 | 0.56 | 0.66 |

Table 2: Accuracy, Precision, Recall, and Balanced Accuracy of each ROUGE variant, averaged across all eight tasks in 2008-2011, with and (without) significance.

As can be seen in Table 1, the manual evaluation metrics often did not show many significant differences between systems.<sup>3</sup> Thus, it is clear that the percent agreement will be high for an approach for automatic evaluation that always predicts zero significant differences. As traditionally done when dealing with such skewed distributions of classes, we also examine the *precision* and *recall* with respect to finding significant differences of several ROUGE variants, to better assess the quality of their prediction. To identify a measure that is strong at both predicting significant and non-significant differences we compute balanced accuracy, the mean of the accuracy of predicting significant differences and the accuracy of predicting no significant difference.<sup>4</sup>

Each of these four measures for judging the performance of ROUGE variants has direct intuitive interpretation, unlike other opaque measures such as correlation coefficients and F-measure which have formal definitions which do not readily yield to intuitive understanding.

<sup>3</sup>This is a somewhat surprising finding which may warrant further investigation. One possible explanation is that different systems generate similar summaries. Recent work has shown that this is unlikely to be the case because the collection of summaries from several systems indicates better what content is important than the single best summary (Louis and Nenkova, 2013). The short summary length for which the summarizers are compared may also contribute to the fact that there are few significant differences. In early NIST evaluations manual evaluations could not distinguish automatic and human summaries based on summaries of length 50 and 100 words and there were more significant differences between systems for 200-word summaries than for 100-word summaries (Nenkova, 2005).

<sup>4</sup>More generally, one could define a utility function which gives costs associated with errors and benefits to correct prediction. Balanced accuracy weighs all errors as equally bad and all correct prediction as equally good (von Neumann and Morgenstern, 1953).

Few prior studies have taken statistical significance into account during the assessment of automatic metrics for evaluation. For this reason we first briefly discuss ROUGE accuracy without taking significance into account. In this special case, agreement simply means that the automatic and manual evaluations agree on which of two systems is better, based on each system’s average score for all test inputs for a given task. It is very rare that the average scores of two systems are equal, so there is always a better system in each pair, and random prediction would have 50% accuracy.

Many papers do not report the significance of differences in ROUGE scores (for the ROUGE variant of their choice), but simply claim that their system  $X$  with higher average ROUGE score than system  $Y$  is better than system  $Y$ . Table 2 lists the average accuracy with significance taken into account and then in parentheses, accuracy without taking significance into account. The data demonstrate that the best accuracy of the eight ROUGE metrics is a meager 64% for responsiveness when significance is not taken into account. So the conclusion about the relative merit of systems would be different from that based on manual evaluation in one out of three comparisons. However, the best accuracy rises to 73% when significance is taken into account; an incorrect conclusion will be drawn in one out of four comparisons. The reduction in error is considerable.

Furthermore, ROUGE-3 and ROUGE-4, which are rarely reported, are among the most accurate. Note also, these results differ considerably from those reported by Owczarzak et al. (2012), where ROUGE-2 was shown to have accuracy of 81% for responsiveness and 89% for pyramid. The wide differences are due to the fact we are only consid-

ering systems which scored in the top 30. This illustrates that our automatic metrics are not as good at discriminating systems near the top. These findings give strong support for the idea of requiring authors to report the significance of the difference between their summarization system and the chosen baseline; the conclusions about relative merits of the system would be more similar to those one would draw from manual evaluation.

In addition to accuracy, Table 2 gives precision, recall and balanced accuracy for each of the eight ROUGE measures when significance is taken into account. ROUGE-1 is arguably the most widely used score in the literature and Table 2 reveals an interesting property: ROUGE-1 has high recall but low precision. This means that it reports many significant differences, most of which do not exist according to the manual evaluations.

Balanced accuracy helps us identify which ROUGE variants are most accurate in finding statistical significance and correctly predicting that two systems are not significantly different. For the pyramid evaluation, the variants with best balanced accuracy (66%) are ROUGE-3 and ROUGE-BE, with ROUGE-4 just a percent lower at 65%. For responsiveness the configuration is similar, with ROUGE-3 and ROUGE-4 tied for best (60%), and ROUGE-BE just a percent lower.

The good performance of higher-order  $n$ -grams is quite surprising because these are practically never used for reporting results in the literature. Based on our results however, they are much more likely to accurately reproduce conclusions that would have been drawn from manual evaluation of top-performing systems.

#### 4 Multiple hypothesis tests to combine ROUGE variants

We now consider a method to combine multiple evaluation scores in order to obtain a stronger ensemble metric. The idea of combining ROUGE variants has been explored in the prior literature. Conroy and Dang (2008), for example, proposed taking linear combinations of ROUGE metrics. This approach was extended by Rankel et al. (2012) by including measures of linguistic quality. Recently, Amigó et al. (2012) applied the “heterogeneity principle” and combined ROUGE scores to improve the *precision* relative to a human evaluation metric. Their results demonstrate that a consensus among ROUGE scores can predict more ac-

curately if an improvement in a human evaluation metric will be achieved.

Along the lines of these investigations, we examine the performance of a simple combination of variants: Call the difference between two systems significant only when *all* the variants in the combination indicate significance. As in the section above, a paired Wilcoxon signed-rank test is used to determine the level of significance.

| ROUGE Combination | Acc  | Prec | Rec  | BA   |
|-------------------|------|------|------|------|
| R1_R2_R4_RBE      | 0.76 | 0.77 | 0.36 | 0.76 |
| R1_R4_RBE         | 0.76 | 0.76 | 0.36 | 0.76 |
| R2_R4_RBE         | 0.76 | 0.74 | 0.40 | 0.75 |
| R4_RBE            | 0.76 | 0.73 | 0.41 | 0.75 |
| R1_R2_R4          | 0.76 | 0.71 | 0.40 | 0.74 |
| R1_R4             | 0.75 | 0.70 | 0.40 | 0.73 |
| R2_R4             | 0.75 | 0.68 | 0.44 | 0.73 |
| R1_R2_RBE         | 0.75 | 0.66 | 0.48 | 0.72 |
| R2_RBE            | 0.75 | 0.64 | 0.52 | 0.72 |
| R4                | 0.74 | 0.62 | 0.47 | 0.70 |
| R1_RBE            | 0.74 | 0.62 | 0.49 | 0.70 |
| R1_R2             | 0.73 | 0.57 | 0.62 | 0.70 |
| RBE               | 0.73 | 0.57 | 0.58 | 0.68 |
| R2                | 0.71 | 0.53 | 0.69 | 0.68 |
| R1                | 0.62 | 0.43 | 0.69 | 0.63 |

Table 3: Accuracy, Precision, Recall, and Balanced Accuracy of each ROUGE combination on TAC 2008-2010 pyramid.

We considered all possible combinations of four ROUGE metrics that exhibited good properties in the analyses presented so far: ROUGE-1 (because of its high recall), ROUGE-2 (because of high accuracy when significance is not taken into account) and ROUGE-4 and ROUGE-BE, which showed good balanced accuracy.

The performance of these combinations for reproducing the decisions in TAC 2008-2010 based on the pyramid<sup>5</sup> evaluation are given in Table 3. The best balanced accuracy (76%) is for the combination of all four variants. As more variants are combined, precision increases but recalls drops.

#### 5 Comparison with automatic evaluations from AESOP 2011

In 2009-2011, TAC ran the task of Automatically Evaluating Summaries of Peers (AESOP), to com-

<sup>5</sup>The ordering of the metric combinations relative to responsiveness was almost identical to the ordering relative to the pyramid evaluation, and precision and recall exhibited the same trend as more metrics were added to the combination.

| Evaluation Metric      | Pyramid A   |             |             |             | Pyramid B   |             |             |             | Responsiveness A |             |             |             | Responsiveness B |             |             |             |
|------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------------|-------------|-------------|-------------|------------------|-------------|-------------|-------------|
|                        | Acc         | P           | R           | BA          | Acc         | P           | R           | BA          | Acc              | P           | R           | BA          | Acc              | P           | R           | BA          |
| CLASSY1                | 0.60        | 0.02        | <b>0.60</b> | 0.50        | <b>0.84</b> | 0.03        | 0.18        | 0.50        | 0.61             | 0.14        | 0.64        | 0.54        | 0.70             | 0.21        | 0.22        | 0.52        |
| DemokritosGR1          | 0.59        | 0.01        | 0.20        | 0.50        | 0.79        | 0.07        | 0.55        | 0.53        | 0.66             | 0.18        | <b>0.79</b> | 0.58        | 0.64             | 0.17        | 0.24        | 0.49        |
| uOttawa3               | 0.44        | 0.01        | <b>0.60</b> | 0.50        | 0.48        | 0.02        | 0.36        | 0.50        | 0.52             | 0.13        | 0.77        | 0.55        | 0.43             | 0.13        | 0.36        | 0.46        |
| DemokritosGR2          | 0.78        | 0.01        | 0.20        | 0.50        | 0.76        | 0.06        | 0.55        | 0.52        | 0.76             | 0.23        | 0.69        | 0.60        | 0.67             | 0.22        | 0.29        | 0.52        |
| C-S-IITH4              | 0.69        | 0.01        | 0.20        | 0.50        | 0.77        | 0.07        | 0.64        | 0.53        | 0.82             | <b>0.29</b> | 0.74        | <b>0.63</b> | 0.60             | 0.15        | 0.24        | 0.47        |
| C-S-IITH1              | 0.60        | 0.01        | 0.40        | 0.50        | 0.70        | 0.06        | <b>0.82</b> | 0.53        | 0.69             | 0.20        | <b>0.79</b> | 0.59        | 0.60             | 0.22        | <b>0.42</b> | 0.52        |
| BEwT-E                 | 0.73        | 0.01        | 0.20        | 0.50        | 0.80        | 0.01        | 0.09        | 0.49        | 0.79             | 0.25        | 0.72        | <b>0.61</b> | 0.72             | <b>0.31</b> | 0.39        | <b>0.58</b> |
| R1-R2-R4-RBE           | <b>0.89</b> | <b>0.40</b> | 0.44        | <b>0.67</b> | 0.76        | 0.27        | 0.17        | 0.55        | <b>0.88</b>      | 0.00        | 0.00        | 0.49        | <b>0.91</b>      | 0.03        | 0.09        | 0.50        |
| R1-R4-RBE              | <b>0.89</b> | <b>0.40</b> | 0.44        | <b>0.67</b> | 0.77        | <b>0.35</b> | 0.24        | <b>0.59</b> | <b>0.88</b>      | 0.00        | 0.00        | 0.49        | 0.90             | 0.03        | 0.09        | 0.50        |
| All ROUGE <sub>s</sub> | <b>0.89</b> | <b>0.40</b> | 0.44        | <b>0.67</b> | 0.75        | 0.26        | 0.16        | 0.54        | <b>0.88</b>      | 0.00        | 0.00        | 0.49        | <b>0.91</b>      | 0.04        | 0.09        | 0.51        |

Table 4: Best performing AESOP systems from TAC 2011; Scores within the 95% confidence interval of the best are in bold face.

pare automatic evaluation methods for automatic summarization. Here we show how the submitted AESOP metrics compare to the best ROUGE variants that we have established so far. We report the results on 2011 only, because even when the same team participated in more than one year, the metrics submitted were different and the 2011 results represent the best effort of these teams. However, as we saw in Table 1, in 2011 there were very few significant differences between the top summarization systems. In this sense the tasks that year represent a challenging dataset for testing automatic evaluations.

The results for the best AESOP systems (according to one or more measures), and the corresponding results for the ROUGE combinations are shown in Table 4. These AESOP systems are: CLASSY1 (Conroy et al., 2011; Rankel et al., 2012), DemokritosGR1 and 2 (Giannakopoulos et al., 2008; Giannakopoulos et al., 2010), uOttawa3 (Kennedy et al., 2011), C-S-IITH1 and 4 (Kumar et al., 2011; Kumar et al., 2012), and BEwT-E (Tratz and Hovy, 2008).<sup>6</sup> The combination metrics achieve the highest accuracy by generally predicting correctly when there are no significant differences between the systems. In addition, for 2008-2010, where far more differences between systems occur, the results of Table 3 show the combination metrics outperformed use of a single metric and are competitive with the best metrics of AESOP 2011. Thus, the combination metrics have the ability to discriminate under both conditions giving good prediction of human evaluation.

<sup>6</sup>To perform the comparison in the table the scores for each system and document set were needed. Some systems have changed after TAC 2011, but the data needed for these comparisons were not available. BEwT-E did not participate in AESOP 2011 and these data were provided by Stephen Tratz. Special thanks to Stephen for providing these data.

## 6 Conclusion

We have tested the best-known automatic evaluation metrics (ROUGE) on several years of TAC data and compared their performance with recently developed AESOP metrics. We discovered that some of the rarely used variants of ROUGE perform surprisingly well, and that by combining different ROUGE<sub>s</sub> together, one can create an evaluation metric that is extremely competitive with metrics submitted to the latest AESOP task. Our results were reported in terms of several different measures, and in each case, compared how well the automatic metric predicted significant differences found in manual evaluation. We believe strongly that developers should include statistical significance when reporting differences in ROUGE scores of theirs and other systems, as this improves the accuracy and credibility of their results. Significant improvement in multiple ROUGE scores is a significantly stronger indicator that the developers have made a noteworthy improvement in text summarization. Systems that report significant improvement using a combination of ROUGE-BE (or its improved version BEwT-E) in conjunction with ROUGE-1, 2, and 4, are more likely to give rise to summaries that humans would judge as significantly better.

## Acknowledgments

The authors would like to thank Ed Hovy who raised the question “How well do automatic metrics perform when comparing top systems?” Ed’s comments helped motivate this work. In addition, we would like to thank our anonymous referees for their insightful comments, which contributed *significantly* to this paper.

## References

- Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. 2012. The heterogeneity principle in evaluation measures for automatic summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 36–43, Montréal, Canada, June. Association for Computational Linguistics.
- John M. Conroy and Hoa Trang Dang. 2008. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 145–152, Manchester, UK, August. Coling 2008 Organizing Committee.
- John M. Conroy, Judith D. Schlesinger, and Dianne P. O’Leary. 2011. Nouveau-ROUGE: A Novelty Metric for Update Summarization. *Computational Linguistics*, 37(1):1–8.
- George Giannakopoulos, Vangelis Karkaletsis, George A. Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *TSLP*, 5(3).
- George Giannakopoulos, George A. Vouros, and Vangelis Karkaletsis. 2010. Mudos-ng: Multi-document summaries using n-gram graphs (tech report). *CoRR*, abs/1012.2042.
- Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated summarization evaluation with basic elements. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, pages 899–902.
- Alistair Kennedy, Anna Kazantseva Saif Mohammad, Terry Copeck, Diana Inkpen, and Stan Szpakowicz. 2011. Getting emotional about news. In *Fourth Text Analysis Conference (TAC 2011)*.
- Niraj Kumar, Kannan Srinathan, and Vasudeva Varma. 2011. Using unsupervised system with least linguistic features for tac-aesop task. In *Fourth Text Analysis Conference (TAC 2011)*.
- N. Kumar, K. Srinathan, and V. Varma. 2012. Using graph based mapping of co-occurring words and closeness centrality score for summarization evaluation. *Computational Linguistics and Intelligent Text Processing*, pages 353–365.
- 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.
- Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39:267–300.
- Ani Nenkova and Annie Louis. 2008. Can you summarize this? identifying correlates of input difficulty for multi-document summarization. In *ACL*, pages 825–833.
- Ani Nenkova, Rebecca J. Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *TSLP*, 4(2).
- Ani Nenkova. 2005. Discourse factors in multi-document summarization. In *AAAI*, pages 1654–1655.
- Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9, Montréal, Canada, June. Association for Computational Linguistics.
- Peter Rankel, John Conroy, Eric Slud, and Dianne O’Leary. 2011. Ranking human and machine summarization systems. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 467–473, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Peter A. Rankel, John M. Conroy, and Judith D. Schlesinger. 2012. Better metrics to automatically predict the quality of a text summary. *Algorithms*, 5(4):398–420.
- Stephen Tratz and Eduard Hovy. 2008. Summarisation evaluation using transformed basic elements. In *Proceedings TAC 2008*. NIST.
- John von Neumann and Oskar Morgenstern. 1953. *Theory of games and economic behavior*. Princeton Univ. Press, Princeton, NJ, 3. ed. edition.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics - Volume 2, COLING ’00*, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.



# On the Predictability of Human Assessment: when Matrix Completion Meets NLP Evaluation

Guillaume Wisniewski

Université Paris Sud

LIMSI-CNRS

Orsay, France

guillaume.wisniewski@limsi.fr

## Abstract

This paper tackles the problem of collecting reliable human assessments. We show that knowing multiple scores for each example instead of a single score results in a more reliable estimation of a system quality. To reduce the cost of collecting these multiple ratings, we propose to use matrix completion techniques to predict some scores knowing only scores of other judges and some common ratings. Even if prediction performance is pretty low, decisions made using the predicted score proved to be more reliable than decision based on a single rating of each example.

## 1 Introduction

Human assessment is often considered as the best, if not the only, way to evaluate ‘subjective’ NLP tasks like MT or speech generation. However, human evaluations are doomed to be noisy and, sometimes, even contradictory as they depend on individual perception and understanding of the score scale that annotators generally use in remarkably different ways (Koehn and Monz, 2006). Moreover, annotation is known to be a long and frustrating process and annotator fatigue has been identified as another source of noise (Pighin et al., 2012).

In addition to defining and enforcing stricter guidelines, several solutions have been proposed to reduce the annotation effort and produce more reliable ratings. For instance, to limit the impact of the score scale interpretation, in the WMT evaluation campaign (Callison-Burch et al., 2012), annotators are asked to rank translation hypotheses

from best to worst instead of providing absolute scores (e.g. in terms of adequacy or fluency). Generalizing this approach, several works (Pighin et al., 2012; Lopez, 2012) have defined novel annotation protocols to reduce the number of judgments that need to be collected. However, all these methods suffer from several limitations: first, they provide no interpretable information about the quality of the system (only a relative comparison between two systems is possible); second, (Koehn, 2012) has recently shown that the ranking they induce is not reliable.

In this work, we study an alternative approach to the problem of collecting reliable human assessments. Our basic assumption, motivated by the success of ensemble methods, is that having several judgments for each example, even if they are noisy, will result in a more reliable decision than having a single judgment. An evaluation campaign should therefore aim at gathering a *score matrix*, in which each example is rated by all judges instead of having each judge rate only a small subset of examples, thereby minimizing redundancy. Obviously, the former approach requires a large annotation effort and is, in practice, not feasible. That is why, to reduce the number of judgments that must be collected, we propose to investigate the possibility of using matrix completion techniques to recover the entire score matrix from a sample of its entries. The question we try to answer is whether the missing scores of one judge can be predicted knowing only scores of other judges and some shared ratings.

The contributions of this paper are twofold: i) we show how knowing the full score matrix instead of a single score for each example provides a more reliable estimation of a system quality (Section 3); ii) we present preliminary experiments

showing that missing data techniques can be used to recover the score matrix from a sample of its entries despite the low inter-rater agreement (Section 4).

## 2 Matrix Completion

The recovering of a matrix from a sampling of its entries is a task of considerable interest (Candès and Recht, 2012). It can be used, for instance, in recommender systems: rows of the matrix represent users that are rating movies (columns of the matrix); the resulting matrix is mostly unknown (each user only rates a few movies) and the task consists in completing the matrix so that movies that any user is likely to like can be predicted.

Matrix completion generally relies on the *low rank hypothesis*: because of hidden factors between the observations (the columns of the matrix), the matrix has a low rank. For instance, in recommender systems it is commonly believed that only a few factors contribute to an individual’s tastes. Formally, recovering a matrix  $M$  amounts at solving:

$$\begin{aligned} & \text{minimize} \quad \text{rank } \mathbf{X} \\ & \text{subject to} \quad X_{ij} = M_{ij} \quad (i, j) \in \Omega \end{aligned} \quad (1)$$

where  $\mathbf{X}$  is the decision variable and  $\Omega$  is the set of known entries. This optimization problem seeks the simplest explanation fitting the observed data.

Solving the rank minimization problem has been proved to be NP-hard (Chistov and Grigor’ev, 1984). However several convex relaxations of this program have been proposed. In this work, we will consider the relaxation of the rank by the nuclear norm<sup>1</sup> that can be efficiently solved by semidefinite programming (Becker et al., 2011). This relaxation enjoys many theoretical guarantees with respect to the optimality of its solution (under mild assumptions its solution is also the solution of the original problem), the conditions under which the matrix can be recovered and the number of entries that must be sampled to recover the original matrix. In our experiments we used TFOCS,<sup>2</sup> a free implementation of this method.

<sup>1</sup>The nuclear norm of a matrix is the sum of its singular values; the relation between rank and nuclear norm is similar to the one between  $\ell_0$  and  $\ell_1$  norms.

<sup>2</sup><http://cvxr.com/tfocs/>

## 3 Corpora

For our experiments we considered two publicly available corpora in which multiple human ratings (i.e. scores on an ordinal scale) were available.

**The CE Corpus** The first corpus of human judgments we have considered has been collected for the WMT12 shared task on quality estimation (Callison-Burch et al., 2012).<sup>3</sup> The data set is made of 2,254 English sentences and their automatic translations in Spanish predicted by a standard Moses system. Each sentence pair is accompanied by three estimates in the range 1 to 5 of its translation quality expressed in terms of post-editing effort. These human grades are in the range 1 to 5, the latter standing for a very good translation that hardly requires post-editing, while the former identifies very poor automatic translations that are not deemed to be worth the post-editing effort.

As pointed out by the task organizers, despite the special care that was taken to ensure the quality of the data, the inter-raters agreement was much lower than what is typically observed in NLP tasks (Artstein and Poesio, 2008): the weighted  $\kappa$  ranged from 0.39 to 0.50 depending on the pair of annotators considered<sup>4</sup>; the Fleiss coefficient (a generalization of  $\kappa$  to multi-raters) was 0.25 and the Kendall  $\tau_b$  correlation coefficient<sup>5</sup> between 0.64 and 0.68, meaning that, on average, two raters do not agree on the relative order of two translations almost two out of five times. In fact, as often observed for the sentence level human evaluation of MT outputs, the different judges have used the score scale differently: the second judge had a clear tendency to give more ‘medium’ scores than the others, and the variance of her scores was low. Because their distributions are different, standardizing the scores has only a very limited impact on the agreement.

If, as in many manual evaluations, each example had been rated by a single judge chosen randomly, the resulting scores would have been only moderately correlated with the average of the three scores which is, intuitively, a better estimate of the ‘true’ quality: the 95% confidence interval of the

<sup>3</sup>The corpus is available from <http://www.statmt.org/wmt12/quality-estimation-task.html>

<sup>4</sup>The weighted  $\kappa$  is a generalization of the  $\kappa$  to ordinal data; a linear weighting schema was used.

<sup>5</sup>Note that, in statistics, agreement is a stronger notion than correlation, as the former compare the actual values.

$\tau_b$  between the averaged scores and the ‘sampled’ score is 0.754–0.755.

**TIDES** The second corpus considered was collected for the DARPA TIDES program: a team of human judges provided multiple assessments of adequacy and fluency for Arabic to English and Chinese to English automatic translations.<sup>6</sup> For space reasons, only results on the Chinese to English fluency corpus will be presented; similar results were achieved on the other corpora.

In the considered corpus, 31 sets of automatic translations, generated by three systems, have been rated by two judges on a scale of 1 to 5. The inter-rater agreement is very low: depending on the pair of judges, the weighted  $\kappa$  is between -0.05 and 0.2, meaning that agreement occurs less often than predicted by chance alone. More importantly, if the ratings of a pair of judges were used to decide which is the best system among two, the two judges will disagree 36% of the time. This ‘agreement’ score is computed as follows: if  $m_{A,i}$  is the mean of the scores given to system  $A$  by the  $i$ -th annotator, we say that there is no agreement in a pairwise comparison if  $m_{A,i} > m_{B,i}$  and  $m_{A,j} < m_{B,j}$ , i.e. if two judges rank two systems in a different order; the score is then the percentage of agreement when considering all pairs of systems and judges.

Considering the full scoring matrix instead of single scores has a large impact: if each example is rated by a single judge (chosen randomly), the resulting comparison between the two systems will be different from the decision made by averaging the two scores of the full score matrix in almost 20% of the comparisons.

## 4 Experimental Results

### 4.1 Testing the Low-Rank Hypothesis

Matrix completion relies on the hypothesis that the matrix has a low rank. We first propose to test this hypothesis on simulated data, using a method similar to the one proposed in (Mathet et al., 2012), to evaluate the impact of noise in human judgments on the score matrix rank. Artificial ratings are generated as follows: a MT system is producing  $n$  translations the quality of which,  $q_i$ , is estimated by a continuous value, that represents, for instance, a hTER score. This

<sup>6</sup>These corpora are available from LDC under the references ldc2003t17 and ldc2003t18

value is drawn from  $\mathcal{N}(\mu, \sigma^2)$ . Based on this ‘intrinsic’ quality, two ratings,  $a_i$  and  $b_i$ , are generated according to three strategies: in the first,  $a_i$  and  $b_i$  are sampled from  $\mathcal{N}(q_i, \theta)$ ; in the second,  $a_i \sim \mathcal{N}(q_i + \frac{\theta}{2}, \sigma'^2)$  and  $b_i \sim \mathcal{N}(q_i - \frac{\theta}{2}, \sigma'^2)$  and in the third,  $a_i \sim \mathcal{N}(q_i, \sigma'^2)$  and the  $b_i$  is drawn from a bimodal distribution  $\frac{1}{2}(\mathcal{N}(q_i - \frac{\theta}{2}, \sigma'^2) + \mathcal{N}(q_i + \frac{\theta}{2}, \sigma'^2))$  (with  $\sigma'^2 < \frac{\theta}{2}$ ).  $\theta$  describes the noise level.

Each of these strategies models a different kind of noise that has been observed in different evaluation campaigns (Koehn and Monz, 2006): the first one describes random noise in the ratings; the second a systematic difference in the annotators’ interpretation of the score scale and the third, the situation in which one annotator gives medium score while the other one tend to commit more strongly to whether she considered the translation good or bad. Stacking all these judgments results in a  $n \times 2$  score matrix. To test whether this matrix has a low rank or not, we assess how close it is to its approximation by a rank 1 matrix. A well-known result (Lawson and Hanson, 1974) states that the Frobenius norm of the difference of these matrices is equal to the 2nd singular value of the original matrix; the quality of the approximation can thus be estimated by  $\rho$ , defined as the 2nd eigenvalue of the matrix normalized by its norm (Leon, 1994). Intuitively, the smaller  $\rho$ , the better the approximation.

Figure 1 represents the impact of the noise level on the condition number. As a baseline, we have also represented  $\rho$  for a random matrix. All values are averaged over 100 simulations. As it could be expected,  $\rho$  is close to 0 for small noise level; but even for moderate noise level, the second eigenvalue continue to be small, suggesting that the matrix can still be approximated by a matrix of rank 1 without much loss of information. As a comparison, on average,  $\rho = 0.08$  for the CE score matrix, in spite of the low inter-rater agreement.

### 4.2 Prediction Performance

We conducted several experiments to evaluate the possibility to use matrix completion to recover a score matrix. Experiments consist in choosing randomly  $k\%$  of the entries of a matrix; these entries are considered unknown and predicted using the method introduced in Section 2 denoted `pred` in the following. In our experiments  $k$  varies from 10% to 40%. Note that, when, as in our exper-

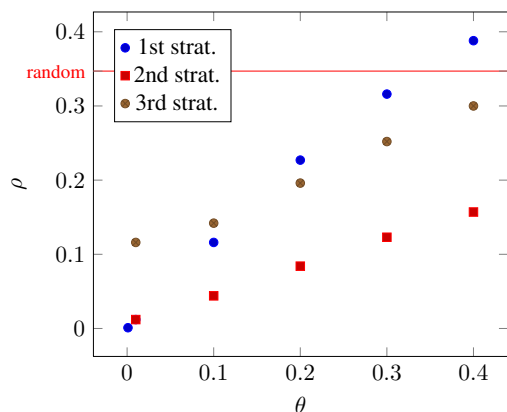


Figure 1: Evolution of the condition number  $\rho$  with the noise level  $\theta$  for the different strategies (see text for details)

iments, only two judges are involved,  $k = 50\%$  would mean that each example is rated by a single judge. Two simple methods for handling missing data are used as baselines: in the first one, denoted `rand`, missing scores are chosen randomly; the second one, denoted `mean`, predicts for all the missing scores of a judge the mean of her known scores.

We propose to evaluate the quality of the recovery, first by comparing the predicted score to their true value and then by evaluating the decision that will be made when considering the recovered matrix instead of the full matrix.

**Prediction Performance** Comparing the completed matrix to the original score matrix can be done in terms of Mean Absolute Error (MAE) defined as  $\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$  where  $\hat{y}_i$  is the predicted value and  $y_i$  the corresponding ‘true’ value; the sum runs over all unknown values of the matrix.

Table 1 presents the results achieved by the different methods. All reported results are averaged over 10 runs (i.e.: sampling of the score matrix and prediction of the missing scores) and over all pairs of judges. All tables also report the 95% confidence interval. The MAE of the `rand` method is almost constant, whatever the number of samples is. Performance of the matrix completion technique is not so good: predicted scores are quite different than true scores. In particular, performance falls quickly when the number of missing data increases. This observation is not surprising: when 40% of the scores are missing, only a few examples have more than a single score and many have no score at all. In these conditions recovering

| missing data | pred                           | mean                           |
|--------------|--------------------------------|--------------------------------|
| 40%          | $0.78 \pm 6.21 \times 10^{-3}$ | $0.72 \pm 8.86 \times 10^{-3}$ |
| 30%          | $0.83 \pm 3.19 \times 10^{-3}$ | $0.80 \pm 5.42 \times 10^{-3}$ |
| 20%          | $0.88 \pm 2.49 \times 10^{-3}$ | $0.87 \pm 3.54 \times 10^{-3}$ |
| 10%          | $0.93 \pm 1.76 \times 10^{-3}$ | $0.92 \pm 1.51 \times 10^{-3}$ |

Table 2: Correlation between the rankings induced by the recovered matrix and the original score matrix for the CE corpus

the matrix is almost impossible. The performance of the simple `mean` technique is, comparatively, pretty good, especially when only a few entries are known. However, the `pred` method always outperform the `rand` method showing that there are dependencies between the two ratings even if statistical measures of agreement are low.

**Impact on the Decision** The negative results of the previous paragraph only provide indirect measure of the recovery quality as it is not the value of the score that is important but the decision that it will support. That is why, we also evaluated matrix recovery in a more task-oriented way by comparing the decision made when considering the recovered score matrix instead of the ‘true’ score matrix.

For the CE corpus, a task-oriented evaluation can be done by comparing the rankings induced by the recovered matrix and by the original matrix when examples are ordered according to their averaged score. Such a ranking can be used by a MT user to set a quality threshold granting her control over translation quality (Soricut and Echihabi, 2010). Table 2 shows the correlation between the two rankings as evaluated by  $\tau_b$ . The two rankings appear to be highly correlated, the matrix completion technique outperforming slightly the `mean` baseline. More importantly, even when 40% of the data are missing, the ranking induced by the true scores is better correlated to the ranking induced by the predicted scores than to the ranking induced when each example is only rated once: as reported in Section 3, the  $\tau_b$  is, in this case, 0.75.

For the TIDES corpus, we computed the number of pairs of judges for which the results of a pairwise comparison between two systems is different when the systems are evaluated using the predicted scores and the true scores. Results presented in Table 3 show that considering the predicted matrix is far better than having judges rate

| k   | QE                           |                              |      | TIDES                        |                              |      |
|-----|------------------------------|------------------------------|------|------------------------------|------------------------------|------|
|     | pred                         | mean                         | rand | pred                         | mean                         | rand |
| 40% | 1.14 $\pm 2.9 \cdot 10^{-2}$ | 0.78 $\pm 6.6 \cdot 10^{-3}$ | 1.45 | —                            | —                            | —    |
| 30% | 0.94 $\pm 2.9 \cdot 10^{-2}$ | 0.78 $\pm 7.4 \cdot 10^{-3}$ | 1.44 | 0.95 $\pm 2.7 \cdot 10^{-2}$ | 0.43 $\pm 2.6 \cdot 10^{-2}$ | 1.37 |
| 20% | 0.77 $\pm 3.4 \cdot 10^{-2}$ | 0.78 $\pm 1.0 \cdot 10^{-2}$ | 1.45 | 0.76 $\pm 2.6 \cdot 10^{-2}$ | 0.41 $\pm 2.5 \cdot 10^{-2}$ | 1.38 |
| 10% | 0.65 $\pm 2.1 \cdot 10^{-2}$ | 0.79 $\pm 1.9 \cdot 10^{-2}$ | 1.47 | 0.48 $\pm 3.0 \cdot 10^{-2}$ | 0.41 $\pm 2.5 \cdot 10^{-2}$ | 1.36 |

Table 1: Completion performance as evaluated by the MAE for the three prediction methods and the three corpora considered.

random samples of the examples: the number of disagreement falls from 20% (Sect. 3) to less than 4%. While the `mean` method outperforms the `pred` method, this result shows that, even in case of low inter-rater agreement, there is still enough information to predict the score of one annotator knowing only the score of the others.

For the tasks considered, decisions based on a recovered matrix are therefore more similar to decisions made considering the full score matrix than decisions based on a single rating of each example.

## 5 Conclusion

This paper proposed a new way of collecting reliable human assessment. We showed, on two corpora, that knowing multiple scores for each example instead of a single score results in a more reliable estimation of the quality of a NLP system. We proposed to use matrix completion techniques to reduce the annotation effort required to collect these multiple ratings. Our experiments showed that while scores predicted using these techniques are pretty different from the true scores, decisions considering them are more reliable than decisions based on a single score.

Even if it can not predict scores accurately, we believe that the connection between NLP evaluation and matrix completion has many potential applications. For instance, it can be applied to identify errors made when collecting scores by comparing the predicted and actual scores.

## 6 Acknowledgments

This work was partly supported by ANR project Trace (ANR-09-CORD-023). The author would like to thank François Yvon and Nicolas Pécheux for their helpful questions and comments on the various drafts of this work.

| % missing data | pred  | mean   |
|----------------|-------|--------|
| 30%            | 9.24% | 3.53 % |
| 20%            | 6.45% | 2.10 % |
| 10%            | 3.66% | 1.20 % |

Table 3: Disagreements in a pairwise comparison of two systems of the TIDES corpus, when the systems are evaluated using the predicted scores and the true scores

## References

- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596, December.
- Stephen R. Becker, Emmanuel J. Candès, and Michael C. Grant. 2011. Templates for convex cone problems with applications to sparse signal recovery. *Math. Prog. Comput.*, 3(3):165–218.
- 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 *Proc. of WMT*, pages 10–51, Montréal, Canada, June. ACL.
- Emmanuel Candès and Benjamin Recht. 2012. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6):111–119, June.
- A. Chistov and D. Grigor’ev. 1984. Complexity of quantifier elimination in the theory of algebraically closed fields. In M. Chytil and V. Koubek, editors, *Math. Found. of Comp. Science*, volume 176, pages 17–31. Springer Berlin / Heidelberg.
- Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proc. WMT*, pages 102–121, New York City, June. ACL.
- Philipp Koehn. 2012. Simulating human judgment in machine translation evaluation campaigns. In *Proc. of IWSLT*.
- Charles L. Lawson and Richard J. Hanson. 1974. *Solving Least Squares Problems*. Prentice Hall.

- Stephen J: Leon. 1994. *Linear Algebra with Applications*. Macmillan,.
- Adam Lopez. 2012. Putting human assessments of machine translation systems in order. In *Proc. of WMT*, pages 1–9, Montréal, Canada, June. ACL.
- Yann Mathet, Antoine Widlcher, Karën Fort, Claire François, Olivier Galibert, Cyril Grouin, Juliette Kahn, Sophie Rosset, and Pierre Zweigenbaum. 2012. Manual corpus annotation: Giving meaning to the evaluation metrics. In *Proceedings of COLING 2012: Posters*, pages 809–818, Mumbai, India, December.
- Daniele Pighin, Lluís Formiga, and Lluís Màrquez. 2012. A graph-based strategy to streamline translation quality assessments. In *Proc. of AMTA*.
- Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proc. of WMT*, pages 259–268, Athens, Greece, March. ACL.
- Radu Soricut and Abdessamad Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proc. of ACL*, pages 612–621, Uppsala, Sweden, July. ACL.

# Automated Pyramid Scoring of Summaries using Distributional Semantics

Rebecca J. Passonneau\* and Emily Chen† and Weiwei Guo† and Dolores Perin‡

\*Center for Computational Learning Systems, Columbia University

†Department of Computer Science, Columbia University

‡Teachers College, Columbia University

(becky@ccls. | ec2805@ | weiwei@cs.) columbia.edu, perin@tc.edu

## Abstract

The pyramid method for content evaluation of automated summarizers produces scores that are shown to correlate well with manual scores used in educational assessment of students' summaries. This motivates the development of a more accurate automated method to compute pyramid scores. Of three methods tested here, the one that performs best relies on latent semantics.

## 1 Introduction

The pyramid method is an annotation and scoring procedure to assess semantic content of summaries in which the content units emerge from the annotation. Each content unit is weighted by its frequency in human reference summaries. It has been shown to produce reliable rankings of automated summarization systems, based on performance across multiple summarization tasks (Nenkova and Passonneau, 2004; Passonneau, 2010). It has also been applied to assessment of oral narrative skills of children (Passonneau et al., 2007). Here we show its potential for assessment of the reading comprehension of community college students. We then present a method to automate pyramid scores based on latent semantics.

The pyramid method depends on two phases of manual annotation, one to identify weighted content units in model summaries written by proficient humans, and one to score target summaries against the models. The first annotation phase yields Summary Content Units (SCUs), sets of text fragments that express the same basic content. Each SCU is weighted by the number of model summaries it occurs in.

Figure 1 illustrates a Summary Content Unit taken from pyramid annotation of five model summaries of an elementary physics text. The elements of an SCU are its index; a label, created by the annotator; contributors (Ctr.), or text fragments from the model summaries; and the weight (Wt.), corresponding to the number of contributors from distinct model summaries. Four of the five model

|        |   |
|--------|---|
| Index  | 105   |
| Label  | <i>Matter is what makes up all objects or substances</i>                        |
| Ctr. 1 | Matter is what makes up all objects or substances                               |
| Ctr. 2 | matter as the stuff that all objects and substances in the universe are made of |
| Ctr. 3 | Matter is identified as being present everywhere and in all substances          |
| Ctr. 4 | Matter is all the objects and substances around us                              |
| Wt.    | 4   |

Figure 1: A Summary Content Unit (SCU)

summaries contribute to SCU 105 shown here. The four contributors have lexical items in common (*matter, objects, substances*), and many differences (*makes up, being present*). SCU weights, which range from 1 to the number of model summaries  $M$ , induce a partition on the set of SCUs in all summaries into subsets  $T_w, w \in 1, \dots, M$ . The resulting partition is referred to as a pyramid because, starting with the subset for SCUs with weight 1, each next subset has fewer SCUs.

To score new target summaries, they are first annotated to identify which SCUs they express. Application of the pyramid method to assessment of student reading comprehension is impractical without an automated method to annotate target summaries. Previous work on automated pyramid scores of automated summarizers performs well at ranking systems on many document sets, but is not precise enough to score human summaries of a single text. We test three automated pyramid scoring procedures, and find that one based on distributional semantics correlates best with manual pyramid scores, and has higher precision and recall for content units in students' summaries than methods that depend on string matching.

## 2 Related Work

The most prominent NLP technique applied to reading comprehension is LSA (Landauer and Dumais, 1997), an early approach to latent semantic analysis claimed to correlate with reading comprehension (Foltz et al., 2000). More recently, LSA

has been incorporated with a suite of NLP metrics to assess students' strategies for reading comprehension using think-aloud protocols (Boonthum-Denecke et al., 2011). The resulting tool, and similar assessment tools such as Coh-Metrix, assess aspects of readability of texts, such as coherence, but do not assess students' comprehension through their writing (Graesser et al., 2004; Graesser et al., 2011). E-rater is an automated essay scorer for standardized tests such as GMAT that also relies on a suite of NLP techniques (Burststein et al., 1998; Burststein, 2003). The pyramid method (Nenkova and Passonneau, 2004), was inspired in part by work in reading comprehension that scores content using human annotation (Beck et al., 1991).

An alternate line of research attempts to replicate human reading comprehension. An automated tool to read and answer questions relies on abductive reasoning over logical forms extracted from text (Wellner et al., 2006). One of the performance issues is resolving meanings of words: removal of WordNet features degraded performance.

The most widely used automated content evaluation is ROUGE (Lin, 2004; Lin and Hovy, 2003). It relies on model summaries, and depends on ngram overlap measures of different types. Because of its dependence on strings, it performs better with larger sets of model summaries. In contrast to ROUGE, pyramid scoring is robust with as few as four or five model summaries (Nenkova and Passonneau, 2004). A fully automated approach to evaluation for ranking systems that requires no model summaries incorporates latent semantic distributional similarities across words (Louis and Nenkova, 2009). The authors note, however, it does not perform well on individual summaries.

### 3 Criteria for Automated Scoring

Pyramid scores of students' summaries correlate well with a manual *main ideas* score developed for an intervention study with community college freshmen who attended remedial classes (Perin et al., In press). Twenty student summaries by students who attended the same college and took the same remedial course were selected from a larger set of 322 that summarized an elementary physics text. All were native speakers of English, and scored within 5 points of the mean reading score for the larger sample. For the intervention study, student summaries had been assigned a score to represent how many main ideas from the source text were covered (Perin et al., In press). Inter-

rater reliability of the main ideas score, as given by the Pearson correlation coefficient, was 0.92.

One of the co-authors created a model pyramid from summaries written by proficient Masters of Education students, annotated 20 target summaries against this pyramid, and scored the result. The raw score of a target summary is the sum of its SCU weights. Pyramid scores have been normalized by the number of SCUs in the summary (analogous to precision), or the average number of SCUs in model summaries (analogous to recall). We normalized raw scores as the average of the two previous normalizations (analogous to F-measure). The resulting scores have a high Pearson's correlation of 0.85 with the main idea score (Perin et al., In press) that was manually assigned to the students' summaries.

To be pedagogically useful, an automated method to assign pyramid scores to students' summaries should meet the following criteria: 1) reliably rank students' summaries of a source text, 2) assign correct pyramid scores, and 3) identify the correct SCUs. A method could do well on criterion 1 but not 2, through scores that have uniform differences from corresponding manual pyramid scores. Also, since each weight partition will have more than one SCU, it is possible to produce the correct numeric score by matching incorrect SCUs that have the correct weights. Our method meets the first two criteria, and has superior performance on the third to other methods.

### 4 Approach: Dynamic Programming

Previous work observed that assignment of SCUs to a target summary can be cast as a dynamic programming problem (Harnly et al., 2005). The method presented there relied on unigram overlap to score the closeness of the match of each eligible substring in a summary against each SCU in the pyramid. It returned the set of matches that yielded the highest score for the summary. It produced good rankings across summarization tasks, but assigned scores much lower than those assigned by humans. Here we extend the DP approach in two ways. We test two new semantic text similarities, a string comparison method and a distributional semantic method, and we present a general mechanism to set a threshold value for an arbitrary computation of text similarity.

Unigram overlap ignores word order, and cannot consider the latent semantic content of a string, only the observed unigram tokens. To



take order into account, we use Ratcliff/Obershelp (R/O), which measures overlap of common subsequences (Ratcliff and Metzner, 1988). To take the underlying semantics into account, we use cosine similarity of 100-dimensional latent vectors of the candidate substrings and of the textual components of the SCU (label and contributors). Because the algorithm optimizes for the total sum of all SCUs, when there is no threshold similarity to count as a match, it favors matching shorter substrings to SCUs with higher weights. Therefore, we add a threshold to the algorithm, below which matches are not considered. Because each similarity metric has different properties and distributions, a single absolute value threshold is not comparable across metrics. We present a method to set comparable thresholds across metrics.

#### 4.1 Latent Vector Representations

To represent the semantics of SCUs and candidate substrings of target summaries, we applied the latent vector model of Guo and Diab (2012).<sup>1</sup> Guo and Diab find that it is very hard to learn a 100-dimension latent vector based only on the limited observed words in a short text. Hence they include unobserved words that provide thousands more features for a short text. This produces more accurate results for short texts, which makes the method suitable for our problem. Weighted matrix factorization (WMF) assigns a small weight for missing words so that latent semantics depends largely on observed words.

A 100-dimension latent vector representation was learned for every span of contiguous words within sentence bounds in a target summary, for the 20 summaries. The training data was selected to be domain independent, so that our model could be used for summaries across domains. Thus we prepared a corpus that is balanced across topics and genres. It is drawn from WordNet sense definitions, Wiktionary sense definitions, and the Brown corpus. It yields a co-occurrence matrix  $M$  of unique words by sentences of size  $46,619 \times 393,666$ .  $M_{ij}$  holds the TF-IDF value of word  $w_i$  in sentence  $s_j$ . Similarly, the contributors to and the label for an SCU were given a 100-dimensional latent vector representation. These representations were then used to compare candidates from a summary to SCUs in the pyramid.

<sup>1</sup><http://www.cs.columbia.edu/~weiwei/code.html#wtmf>.

#### 4.2 Three Comparison Methods

An SCU consists of at least two text strings: the SCU label and one contributor. As in Harnly et al. (2005), we use three similarity comparisons  $scusim(X, SCU)$ , where  $X$  is the target summary string. When the comparison parameter is set to  $\min(\max, \text{or mean})$ , the similarity of  $X$  to each SCU contributor and the label is computed in turn, and the minimum ( $\max, \text{or mean}$ ) is returned.

#### 4.3 Similarity Thresholds

We define a threshold parameter for a target SCU to match a pyramid SCU based on the distributions of scores each similarity method gives to the target SCUs identified by the human annotator. Annotation of the target summaries yielded 204 SCUs. The similarity score being a continuous random variable, the empirical sample of 204 scores is very sparse. Hence, we use a Gaussian kernel density estimator to provide a non-parametric estimation of the probability densities of scores assigned by each of the similarity methods to the manually identified SCUs. We then select five threshold values corresponding to those for which the inverse cumulative density function (icdf) is equal to 0.05, 0.10, 0.15, 0.20 and 0.25. Each threshold represents the probability that a manually identified SCU will be missed.

### 5 Experiment

The three similarity computations, three methods to compare against SCUs, and five icdf thresholds yield 45 variants, as shown in Figure 2. Each variant was evaluated by comparing the unnormalized automated variant, e.g., Lvc, max, 0.64 (its 0.15 icdf) to the human gold scores, using each of the evaluation metrics described in the next subsection. To compute confidence intervals for the evaluation metrics for each variant, we use bootstrapping with 1000 samples (Efron and Tibshirani, 1986).

To assess the 45 variants, we compared their scores to the manual scores. We also compared the sets of SCUs retrieved. By our criterion 1), an automated score that correlates well with manual scores for summaries of a given text could be used

$$(3 \text{ Similarities}) \times (3 \text{ Comparisons}) \times (5 \text{ Thresholds}) = 45 \\ (\text{Uni, R/O, Lvc}) \times (\min, \text{mean}, \max) \times (0.05, \dots, 0.25)$$

Figure 2: Notation used for the 45 variants

| Variant (with icdf)    | P (95% conf.), rank  | S (95% conf.), rank  | K (95% conf.), rank   | $\mu$ | Diff. | T test |
|------------------------|----------------------|----------------------|-----------------------|-------|-------|--------|
| LVC, max, 0.64 (0.15)  | 0.93 (0.94, 0.92), 1 | 0.94 (0.93, 0.97), 1 | 0.88 (0.85, 0.91), 1  | 49.9  | 15.65 | 0.0011 |
| R/O, mean, 0.23 (0.15) | 0.92 (0.91, 0.93), 3 | 0.93 (0.91, 0.95), 2 | 0.83 (0.80, 0.86), 3  | 49.8  | 15.60 | 0.0012 |
| R/O, mean, 0.26 (0.20) | 0.92 (0.90, 0.93), 4 | 0.92 (0.90, 0.94) 4  | 0.80 (0.78, 0.83), 5  | 47.7  | 13.45 | 0.0046 |
| LVC, max, 0.59 (0.10)  | 0.91 (0.89, 0.92), 8 | 0.93 (0.91, 0.95) 3  | 0.83 (0.80, 0.87), 2  | 52.7  | 18.50 | 0.0002 |
| LVC, min, 0.40 (0.20)  | 0.92 (0.90, 0.93), 2 | 0.87 (0.84, 0.91) 11 | 0.74 (0.69, 0.79), 11 | 37.5  | 3.30  | 0.4572 |

Table 1: Five variants from the top twelve of all correlations, with confidence interval and rank (P=Pearson’s, S=Spearman, K=Kendall’s tau), mean summed SCU weight, difference of mean from mean gold score, T test p-value.

to indicate how well students rank against other students. We report several types of correlation tests. Pearson’s tests the strength of a linear correlation between the two sets of scores; it will be high if the same order is produced, with the same distance between pairs of scores. The Spearman rank correlation is said to be preferable for ordinal comparisons, meaning where the unit interval is less relevant. Kendall’s tau, an alternative rank correlation, is less sensitive to outliers and more intuitive. It is the proportion of concordant pairs (pairs in the same order) less the proportion of discordant pairs. Since correlations can be high when differences are uniform, we use Student’s T to test whether differences score means statistically significant. Criterion 2) is met if the correlations are high and the means are not significantly different.

## 6 Results

The correlation tests indicate that several variants achieve sufficiently high correlations to rank students’ summaries (criterion 2). On all correlation tests, the highest ranking automated method is LVC, max, 0.64; this similarity threshold corresponds to the 0.15 icdf. As shown in Table 1, the Pearson correlation is 0.93. Note, however, that it is not significantly higher than many of its competitors. LVC, min, 0.40 did not rank as highly for Spearman and Kendall’s tau correlations, but the Student’s T result in column 3 of Table 1 shows that this is the only variant in the table that yields absolute scores that are not significantly different from the human annotated scores. Thus this variant best balances criteria 1 and 2.

The differences in the unnormalized score computed by the automated systems from the score assigned by human annotation are consistently positive. Inspection of the SCUs retrieved by each automated variant reveals that the automated systems lean toward the tendency to identify false positives. This may result from the DP implementation decision to maximize the score. To get a measure of the degree of overlap between the SCUs that were selected automatically versus manually (cri-

terion 4), we computed recall and precision for the various methods. Table 2 shows the mean recall and precision (with standard deviations) across all five thresholds for each combination of similarity method and method of comparison to the SCU. The low standard deviations show that the recall and precision are relatively similar across thresholds for each variant. The LVC methods outperform R/O and unigram overlap methods, particularly for the precision of SCUs retrieved, indicating the use of distributional semantics is a superior approach for pyramid summary scoring than methods based on string matching.

The unigram overlap and R/O methods show the least variation across comparison methods (min, mean, max). LVC methods outperform them, on precision (Table 2). Meeting all three criteria is difficult, and the LVC method is clearly superior.

## 7 Conclusion

We extended a dynamic programming framework (Harnly et al., 2005) to automate pyramid scores more accurately. Improvements resulted from principled thresholds for similarity, and from a vector representation (LVC) to capture the latent semantics of short spans of text (Guo and Diab, 2012). The LVC methods perform best at all three criteria for a pedagogically useful automatic metric. Future work will address how to improve precision and recall of the gold SCUs.

## Acknowledgements

We thank the reviewers for very valuable insights.

| Variant   | $\mu$ Recall (std) | $\mu$ Precision (std) | F score |
|-----------|--------------------|-----------------------|---------|
| Uni, min  | 0.69 (0.08)        | 0.35 (0.02)           | 0.52    |
| Uni, max  | 0.70 (0.03)        | 0.35 (0.04)           | 0.53    |
| Uni, mean | 0.69 (0.02)        | 0.39 (0.04)           | 0.54    |
| R/O, min  | 0.69 (0.08)        | 0.34 (0.01)           | 0.51    |
| R/O, max  | 0.72 (0.03)        | 0.33 (0.04)           | 0.52    |
| R/O, mean | 0.71 (0.06)        | 0.38 (0.02)           | 0.54    |
| LVC, min  | 0.61 (0.03)        | 0.38 (0.04)           | 0.49    |
| LVC, max  | 0.74 (0.06)        | 0.48 (0.01)           | 0.61    |
| LVC, mean | 0.75 (0.06)        | 0.50 (0.02)           | 0.62    |

Table 2: Recall and precision for SCU selection

## References

- Isabel L. Beck, Margaret G. McKeown, Gale M. Sinatra, and Jane A. Loxterman. 1991. Revising social studies text from a text-processing perspective: Evidence of improved comprehensibility. *Reading Research Quarterly*, pages 251–276.
- Chutima Boonthum-Denecke, Philip M. McCarthy, Travis A. Lamkin, G. Tanner Jackson, Joseph P. Maglianoc, and Danielle S. McNamara. 2011. Automatic natural language processing and the detection of reading skills and reading comprehension. In *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, pages 234–239.
- Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. 1998. Automated scoring using a hybrid feature identification technique. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 206–210, Montreal, Quebec, Canada, August. Association for Computational Linguistics.
- Jill Burstein. 2003. The e-rater® scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis and J. Burstein, editors, *Automated Essay Scoring: A Cross-disciplinary Perspective*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.
- Bradley Efron and Robert Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1:54–77.
- Peter W. Foltz, Sara Gilliam, and Scott Kendall. 2000. Supporting content-based feedback in on-line writing evaluation with LSA. *Interactive Learning Environments*, 8:111–127.
- Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36:193202.
- Arthur C. Graesser, Danielle S. McNamara, and Jonna M. Kulikowich. 2011. Coh-Metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40:223–234.
- Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 864–872.
- Aaron Harnly, Ani Nenkova, Rebecca J. Passonneau, and Owen Rambow. 2005. Automation of summary evaluation by the Pyramid Method. In *Recent Advances in Natural Language Processing (RANLP)*, pages 226–232.
- Thomas K Landauer and Susan T. Dumais. 1997. A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 71–78.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 463–470.
- Annie Louis and Ani Nenkova. 2009. Evaluating content selection in summarization without human models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 306–314, Singapore, August. Association for Computational Linguistics.
- Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–152.
- Rebecca J. Passonneau, Adam Goodkind, and Elena Levy. 2007. Annotation of children’s oral narrations: Modeling emergent narrative skills for computational applications. In *Proceedings of the Twentieth Annual Meeting of the Florida Artificial Intelligence Research Society (FLAIRS-20)*, pages 253–258. AAAI Press.
- Rebecca Passonneau. 2010. Formal and functional assessment of the Pyramid Method for summary content evaluation. *Natural Language Engineering*, 16.
- D. Perin, R. H. Bork, S. T. Peverly, and L. H. Mason. In press. A contextualized curricular supplement for developmental reading and writing. *Journal of College Reading and Learning*.
- J. W. Ratcliff and D. Metzener. 1988. Pattern matching: the Gestalt approach.
- Ben Wellner, Lisa Ferro, Warren R. Greiff, and Lynette Hirschman. 2006. Reading comprehension tests for computer-based understanding evaluation. *Natural Language Engineering*, 12(4):305–334.

# Are Semantically Coherent Topic Models Useful for Ad Hoc Information Retrieval?

**Romain Deveaud**     **Eric SanJuan**

University of Avignon - LIA  
Avignon, France

romain.deveaud@univ-avignon.fr

eric.sanjuan@univ-avignon.fr

**Patrice Bellot**

Aix-Marseille University - LSIS  
Marseille, France

patrice.bellot@lsis.org

## Abstract

The current topic modeling approaches for Information Retrieval do not allow to explicitly model query-oriented latent topics. More, the semantic coherence of the topics has never been considered in this field. We propose a model-based feedback approach that learns Latent Dirichlet Allocation topic models on the top-ranked pseudo-relevant feedback, and we measure the semantic coherence of those topics. We perform a first experimental evaluation using two major TREC test collections. Results show that retrieval performances tend to be better when using topics with higher semantic coherence.

## 1 Introduction

Representing documents as mixtures of “topics” has always been a challenge and an objective for researchers working in text-related fields. Based on the words used within a document, topic models learn topic level relations by assuming that the document covers a small set of concepts. Learning the topics from a document collection can help to extract high level semantic information, and help humans to understand the meaning of documents. Latent Semantic Indexing (Deerwester et al., 1990) (LSI), probabilistic Latent Semantic Analysis (Hofmann, 2001) (pLSA) and Latent Dirichlet Allocation (Blei et al., 2003) (LDA) are the most famous approaches that tried to tackle this problem throughout the years. Topics produced by these methods are generally fancy and appealing, and often correlate well with human concepts. This is one of the reasons of the intensive use of topic models (and especially LDA) in current research in Natural Language Processing (NLP) related areas.

One main problem in *ad hoc* Information Retrieval (IR) is the difficulty for users to translate a

complex information need into a keyword query. The most popular and effective approach to overcome this problem is to improve the representation of the query by adding query-related “concepts”. This approach mostly relies on pseudo-relevance feedback, where these so-called “concepts” are the most frequent words occurring in the top-ranked documents retrieved by the retrieval system (Lavrenko and Croft, 2001). From that perspective, topic models seem attractive in the sense that they can provide a descriptive and intuitive representation of concepts. But how can we quantify the usefulness of these topics with respect to an IR system? Recently, researchers developed measures which evaluate the semantic coherence of topic models (Newman et al., 2010; Mimno et al., 2011; Stevens et al., 2012). We adopt their view of semantic coherence and apply one of these measures to query-oriented topics.

Several studies concentrated on improving the quality of document ranking using topic models, especially probabilistic ones. The approach by Wei and Croft (2006) was the first to leverage LDA topics to improve the estimate of document language models and achieved good empirical results. Following this pioneering work, several studies explored the use of pLSA and LDA under different experimental settings (Park and Ramamohanarao, 2009; Yi and Allan, 2009; Andrzejewski and Buttler, 2011; Lu et al., 2011). The reported results suggest that the words and the probability distributions learned by probabilistic topic models are effective for query expansion. The main drawback of these approaches is that topics are learned on the whole target document collection prior to retrieval, thus leading to a static topical representation of the collection. Depending on the query and on its specificity, topics may either be too coarse or too fine to accurately represent the latent concepts of the query. Recently, Ye et al. (2011) proposed a method which uses

LDA and learns topics directly on a limited set of documents. While this approach is a first step towards modeling query-oriented topics, it lacks some theoretic principles and only aims to heuristically construct a “best” topic (from all learned topics) before expanding the query with its most probable words. More, none of the aforementioned works studied the semantic coherence of those generated topics. We tackle these issues by making the following contributions:

- we introduce Topic-Driven Relevance Models, a model-based feedback approach (Zhai and Lafferty, 2001) for integrating topic models into relevance models by learning topics *on* pseudo-relevant feedback documents (as opposed to the entire document collection),
- we explore the coherence of those generated topics using the queries of two major and well-established TREC test collections,
- we evaluate the effects coherent topics have on *ad hoc* IR using the same test collections.

## 2 Topic-Driven Relevance Models

### 2.1 Relevance Models

The goal of relevance models is to improve the representation of a query  $Q$  by selecting terms from a set of initially retrieved documents (Lavrenko and Croft, 2001). As the concentration of relevant documents is usually higher in the top ranks of the ranking list, this is constituted by a number  $N$  of top-ranked documents. Relevance models usually perform better when combined with the original query model (or maximum likelihood estimate). Let  $\tilde{\theta}_Q$  be this maximum likelihood query estimate and  $\hat{\theta}_Q$  a relevance model, the updated new query model is given by:

$$P(w|\theta_Q) = \lambda P(w|\tilde{\theta}_Q) + (1 - \lambda)P(w|\hat{\theta}_Q) \quad (1)$$

where  $\lambda \in [0, 1]$  is a parameter that controls the tradeoff between the original query model and the relevance model. One of the most robust variants of the relevance models is computed as follows:

$$P(w|\hat{\theta}_Q) \propto \sum_{\theta_D \in \Theta} P(\theta_D) P(w|\theta_D) \prod_{t \in Q} P(t|\theta_D) \quad (2)$$

where  $\Theta$  is a set of pseudo-relevant feedback documents and  $\theta_D$  is the language model of document  $D$ . This notion of estimating a query model is

often referred to as model-based feedback (Zhai and Lafferty, 2001). We assume  $P(\theta_D)$  to be uniform, resulting in an estimated relevance model based on a sum of document models weighted by the query likelihood score. The final, interpolated, estimate expressed in equation (1) is often referred in the literature as RM3. We tackle the null probabilities problem by smoothing the document language model using the well-known Dirichlet smoothing (Zhai and Lafferty, 2004).

### 2.2 LDA-based Feedback Model

The estimation of the feedback model  $\hat{\theta}_Q$  constitutes the first contribution of this work. We propose to explicitly model the latent topics (or concepts) that exist behind an information need, and to use them to improve the query representation. We consider  $\Theta$  as the set of pseudo-relevant feedback documents from which the latent concepts would be extracted. The retrieval algorithm used to obtain these documents can be of any kind, the important point is that  $\Theta$  is a reduced collection that contains the top documents ranked by an automatic and state-of-the-art retrieval process.

Instead of viewing  $\Theta$  as a set of document language models that are likely to contain topical information about the query, we take a probabilistic topic modeling approach. We specifically focus on Latent Dirichlet Allocation (LDA), since it is currently one of the most representative. In LDA, each topic multinomial distribution  $\phi_k$  is generated by a conjugate Dirichlet prior with parameter  $\beta$ , while each document multinomial distribution  $\theta_d$  is generated by a conjugate Dirichlet prior with parameter  $\alpha$ . In other words,  $\theta_{d,k}$  is the probability of topic  $k$  occurring in document  $D$  (i.e.  $P(k|D)$ ). Respectively,  $\phi_{k,w}$  is the probability of word  $w$  belonging to topic  $k$  (i.e.  $P(w|k)$ ). We use variational inference implemented in the LDA-C software<sup>1</sup> to overcome intractability issues (Blei et al., 2003; Griffiths and Steyvers, 2004). Under this setting, we compute the topic-driven estimation of the query model using the following equation:

$$P(w|\hat{\theta}_Q) \propto \sum_{\theta_D \in \Theta} \left( P(\theta_D) P(w|\theta_D) P_{TM}(w|D) \prod_{t \in Q} P(t|\theta_D) \right) \quad (3)$$

where  $P_{TM}(w|D)$  is the probability of word  $w$  occurring in document  $D$  using the previously

<sup>1</sup>[www.cs.princeton.edu/~blei/lda-c](http://www.cs.princeton.edu/~blei/lda-c)

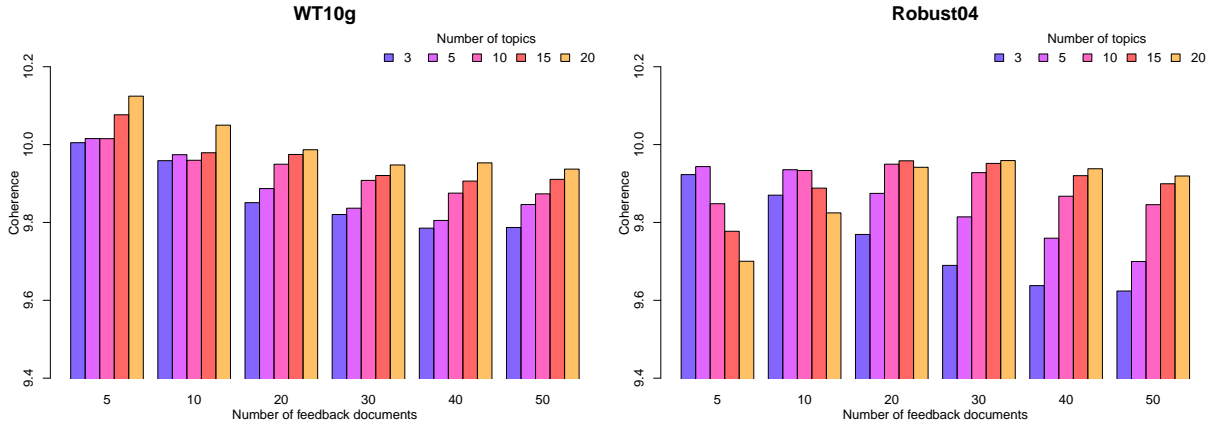


Figure 1: Semantic coherence of the topic models for different values of  $K$ , in function of the number  $N$  of feedback documents.

learned multinomial distributions. Let  $\mathcal{T}_\Theta$  be a topic model learned on the  $\Theta$  set of feedback documents, this probability is given by:

$$P_{TM}(w|D) = \sum_{k \in \mathcal{T}_\Theta} \phi_{k,w} \cdot \theta_{D,k} \quad (4)$$

High probabilities are thus given to words that are important in topic  $k$ , when  $k$  is an important topic in document  $D$ . In the remainder of this paper, we refer to this general approach as TDRM for Topic-Driven Relevance Models.

### 2.3 Measuring the coherence of query-oriented topics

TDRM relies on two important parameters: the number of topics  $K$  that we want to learn, and the number of feedback documents  $N$  from which LDA learns the topics. Varying these two parameters can help to capture more information and to model finer topics, but how about their global semantic coherence?

Term similarities measured in restricted domains was the first step for evaluating semantic coherence (Gliozzo et al., 2007), and was a first basis for the development of several topic coherence evaluation measures (Newman et al., 2010). Computing the Pointwise Mutual Information (PMI) of all word pairs over Wikipedia was found to be an effective metric using news and books corpora. Recently, Stevens et al. (2012) used (among others) an aggregate version of this metric to evaluate large amounts of topic models. We use this method to evaluate the coherence of query-oriented topics. Specifically, the coherence

of a topic model  $\mathcal{T}_\Theta^K$  composed of  $K$  topics is:

$$c(\mathcal{T}_\Theta^K) = \frac{1}{K} \sum_{i=1}^K \sum_{(w,w') \in k_i} \log \frac{P(w,w') + \epsilon}{P(w)P(w')} \quad (5)$$

where probabilities of word occurrences and co-occurrences are estimated using an external reference corpus. Following Newman et al. (2010), we use Wikipedia to compute PMI and set  $\epsilon = 1$  as in (Stevens et al., 2012).

## 3 Evaluation

### 3.1 Experimental setup

We performed our evaluation using two main TREC<sup>2</sup> collections: Robust04 and WT10g. Robust04 is composed 528,155 of news articles coming from three newspapers and the FBIS. It supported the TREC 2004 Robust track, from which we used the 250 query topics (numbers: 301-450, 601-700). The WT10g collection is composed of 1,692,096 web pages, and supported the TREC Web track for four years (2001-2004). We focus on the 2000 and 2001 ad-hoc query topics (numbers: 451-550). We used the open-source indexing and retrieval system Indri<sup>3</sup> to run our experiments. We indexed the two collections with the exact same parameters: tokens were stemmed with the well-known light Krovetz stemmer and stopwords were removed using the standard English stoplist embedded with Indri (417 words).

### 3.2 Semantic coherence evaluation

Most coherent topics are composed of rare words that do not often occur in the reference corpus, but

<sup>2</sup>trec.nist.gov

<sup>3</sup>lemurproject.org/indri.php

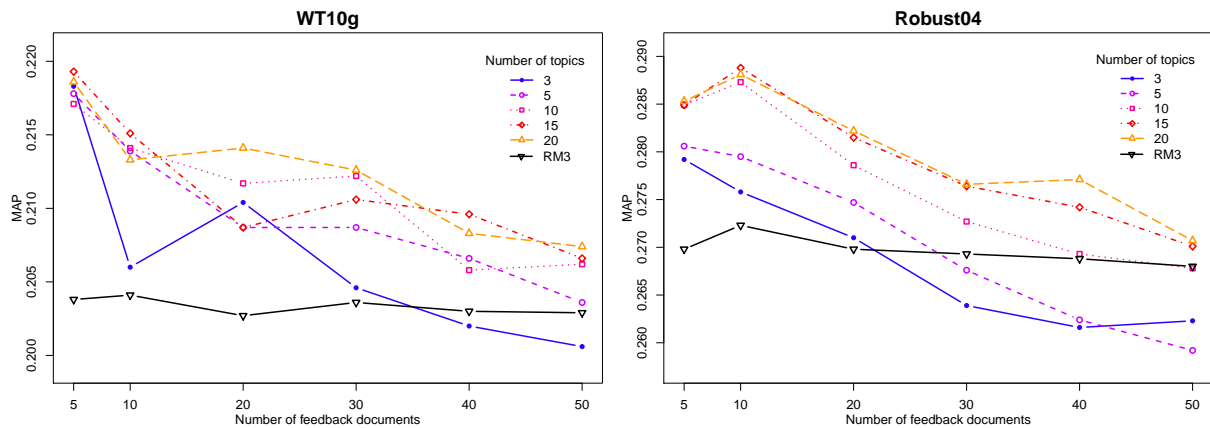


Figure 2: Retrieval performance in terms of Mean Average Precision (MAP) of the TDRM approach. Each line represent a different number of topics  $K$ , and the performance are reported in function the number  $N$  of feedback documents. The black, plain line represents the RM3 baseline.

co-occur at lot together. We see on Figure 1 that very coherent topics are identified in the top 5 and 10 feedback documents for the WT10g collection, suggesting that closely related documents are retrieved in the top ranks. Results are quite different on the Robust04 collection, where topic models with 20 topics on 5 documents are the least coherent. However, when looking at the Robust04 documents, we see that they are on average almost twice smaller than the WT10g web pages. We hypothesize that the heterogeneous nature of the web allows to model very different topics covering several aspects of the query, while news articles are contributions focused on a single subject.

Overall, the more coherent topic models contain a reasonable amount of topics (10-15), thus allowing to fit with variable amounts of documents. The attentive reader will notice that the topic coherence scores are very high compared to those previously reported in the literature (Stevens et al., 2012). The TDRM approach captures topics that are centered around a specific information need, often with a limited vocabulary, which favors word co-occurrence. On the other hand, topics learned on entire collections are coarser than ours, which leads to lower coherence scores.

### 3.3 Document retrieval results

Since TDRM is based on Relevance Models (Lavrenko and Croft, 2001), we take the RM3 approach presented in Section 2.1 as baseline. The  $\lambda$  parameter is common between RM3 and TDRM and is determined for each query using leave-one-query-out cross-validation (that is: learn the

best parameter setting for all queries but one, and evaluate the held-out query using the previously learned parameter).

We report *ad hoc* document retrieval performances in Figure 2. We noticed in the previous section that the most coherent topic models were modeled using 5 feedback documents and 20 topics for the WT10g collection, and this parameter combination also achieves the best retrieval results. Overall, using 10, 15 or 20 topics allow it to achieve high and similar performance from 5 to 20 documents. We observe than using 20 topics for the Robust04 collection consistently achieves the highest results, with the topic model coherence growing as the number of feedback documents increases. Although topics coming from news articles may be limited, they benefit from the rich vocabulary of professional writers who are trained to avoid repetition. Their use of synonyms allows TDRM to model deep topics, with a comprehensive description of query aspects. Since synonyms are less likely to co-occur in encyclopedic articles like Wikipedia, we think that, in our case, the semantic coherence measure could be more accurate using other textual resources. This measure seems however to be effective when dealing with heterogeneously structured documents.

## 4 Conclusions & Future Work

Overall, modeling query-oriented topic models and estimating the feedback query model using these topics greatly improves *ad hoc* Information Retrieval, compared to state-of-the-art relevance models. While semantically coherent topic mod-

els do not seem to be effective in the context of a news articles search task, they are a good indicator of effectiveness in the context of web search. Measuring the semantic coherence of query topics could help predict query effectiveness or even choose the best query-representative topic model.

## Acknowledgments

This work was supported by the French Agency for Scientific Research (Agence Nationale de la Recherche) under CAAS project (ANR 2010 CORD 001 02).

## References

- David Andrzejewski and David Buttler. 2011. Latent Topic Feedback for Information Retrieval. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 600–608.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Alfio Massimiliano Gliozzo, Marco Pennacchiotti, and Patrick Pantel. 2007. The Domain Restriction Hypothesis: Relating Term Similarity and Semantic Consistency. In *Human Language Technologies: The 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 131–138.
- Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl.
- Thomas Hofmann. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196.
- Victor Lavrenko and W. Bruce Croft. 2001. Relevance-Based Language Models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127.
- Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14:178–203.
- David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272.
- David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108.
- Laurence A. Park and Kotagiri Ramamohanarao. 2009. The Sensitivity of Latent Dirichlet Allocation for Information Retrieval. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD '09, pages 176–188.
- Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 952–961.
- Xing Wei and W. Bruce Croft. 2006. LDA-based Document Models for Ad-hoc Retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185.
- Zheng Ye, Jimmy Xiangji Huang, and Hongfei Lin. 2011. Finding a Good Query-Related Topic for Boosting Pseudo-Relevance Feedback. *JASIST*, 62(4):748–760.
- Xing Yi and James Allan. 2009. A Comparative Study of Utilizing Topic Models for Information Retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 29–41. Springer-Verlag.
- 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.
- Chengxiang Zhai and John Lafferty. 2004. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214.



# Post-Retrieval Clustering Using Third-Order Similarity Measures

**José G. Moreno**  
Normandie University  
UNICAEN, GREYC CNRS  
F-14032 Caen, France  
jose.moreno@unicaen.fr

**Gaël Dias**  
Normandie University  
UNICAEN, GREYC CNRS  
F-14032 Caen, France  
gael.dias@unicaen.fr

**Guillaume Cleuziou**  
University of Orléans  
LIFO  
F-45067 Orléans, France  
cleuziou@univ-orleans.fr

## Abstract

Post-retrieval clustering is the task of clustering Web search results. Within this context, we propose a new methodology that adapts the classical  $K$ -means algorithm to a third-order similarity measure initially developed for NLP tasks. Results obtained with the definition of a new stopping criterion over the ODP-239 and the MORESQUE gold standard datasets evidence that our proposal outperforms all reported text-based approaches.

## 1 Introduction

Post-retrieval clustering (PRC), also known as search results clustering or ephemeral clustering, is the task of clustering Web search results. For a given query, the retrieved Web snippets are automatically clustered and presented to the user with meaningful labels in order to minimize the information search process. This technique can be particularly useful for polysemous queries but it is hard to implement efficiently and effectively (Carpineto et al., 2009). Indeed, as opposed to classical text clustering, PRC must deal with small collections of short text fragments (Web snippets) and be processed in run-time.

As a consequence, most of the successful methodologies follow a monothetic approach (Zamir and Etzioni, 1998; Ferragina and Gulli, 2008; Carpineto and Romano, 2010; Navigli and Crisafulli, 2010; Scaiella et al., 2012). The underlying idea is to discover the most discriminant topical words in the collection and group together Web snippets containing these relevant terms. On the other hand, the polythetic approach which main idea is to represent Web snippets as word feature vectors has received less attention, the only relevant work being (Osinski and Weiss, 2005). The main reasons for this situation are that (1) word

feature vectors are hard to define in small collections of short text fragments (Timonen, 2013), (2) existing second-order similarity measures such as the cosine are unadapted to capture the semantic similarity between small texts, (3) Latent Semantic Analysis has evidenced inconclusive results (Osinski and Weiss, 2005) and (4) the labeling process is a surprisingly hard extra task (Carpineto et al., 2009).

This paper is motivated by the fact that the polythetic approach should lead to improved results if correctly applied to small collections of short text fragments. For that purpose, we propose a new methodology that adapts the classical  $K$ -means algorithm to a third-order similarity measure initially developed for Topic Segmentation (Dias et al., 2007). Moreover, the adapted  $K$ -means algorithm allows to label each cluster directly from its centroids thus avoiding the abovementioned extra task. Finally, the evolution of the objective function of the adapted  $K$ -means is modeled to automatically define the “best” number of clusters.

Finally, we propose different experiments over the ODP-239 (Carpineto and Romano, 2010) and MORESQUE (Navigli and Crisafulli, 2010) datasets against the most competitive text-based PRC algorithms: STC (Zamir and Etzioni, 1998), LINGO (Osinski and Weiss, 2005), OPTIMSRC (Carpineto and Romano, 2010) and the classical bisecting incremental  $K$ -means (which may be seen as a baseline for the polythetic paradigm)<sup>1</sup>. A new evaluation measure called the b-cubed  $F$ -measure ( $F_{b^3}$ ) and defined in (Amigó et al., 2009) is then calculated to evaluate both cluster homogeneity and completeness. Results evidence that our proposal outperforms all state-of-the-art approaches with a maximum  $F_{b^3} = 0.452$  for ODP-239 and  $F_{b^3} = 0.490$  for MORESQUE.

<sup>1</sup>The TOPICAL algorithm proposed by (Scaiella et al., 2012) is a knowledge-driven methodology based on Wikipedia.

## 2 Polythetic Post-Retrieval Clustering

The  $K$ -means is a geometric clustering algorithm (Lloyd, 1982). Given a set of  $n$  data points, the algorithm uses a local search approach to partition the points into  $K$  clusters. A set of  $K$  initial cluster centers is chosen. Each point is then assigned to the center closest to it and the centers are recomputed as centers of mass of their assigned points. The process is repeated until convergence. To assure convergence, an objective function  $Q$  is defined which decreases at each processing step. The classical objective function is defined in Equation (1) where  $\pi_k$  is a cluster labeled  $k$ ,  $x_i \in \pi_k$  is an object in the cluster,  $m_{\pi_k}$  is the centroid of the cluster  $\pi_k$  and  $E(\cdot, \cdot)$  is the Euclidean distance.

$$Q = \sum_{k=1}^K \sum_{x_i \in \pi_k} E(x_i, m_{\pi_k})^2. \quad (1)$$

Within the context of PRC, the  $K$ -means algorithm needs to be adapted to integrate third-order similarity measures (Mihalcea et al., 2006; Dias et al., 2007). Third-order similarity measures, also called weighted second-order similarity measures, do not rely on exact matches of word features as classical second-order similarity measures (e.g. the cosine metric), but rather evaluate similarity based on related matches. In this paper, we propose to use the third-order similarity measure called InfoSimba introduced in (Dias et al., 2007) for Topic Segmentation and implement its simplified version  $S_s^3$  in Equation 2.

$$S_s^3(X_i, X_j) = \frac{1}{p^2} \sum_{k=1}^p \sum_{l=1}^p X_{ik} * X_{jl} * S(W_{ik}, W_{jl}). \quad (2)$$

Given two Web snippets  $X_i$  and  $X_j$ , their similarity is evaluated by the similarity of its constituents based on any symmetric similarity measure  $S(\cdot, \cdot)$  where  $W_{ik}$  (resp.  $W_{jl}$ ) corresponds to the word at the  $k^{th}$  (resp.  $l^{th}$ ) position in the vector  $X_i$  (resp.  $X_j$ ) and  $X_{ik}$  (resp.  $X_{jl}$ ) is the weight of word  $W_{ik}$  (resp.  $W_{jl}$ ) in the set of retrieved Web snippets. A direct consequence of the change in similarity measure is the definition of a new objective function  $Q_{S_s^3}$  to ensure convergence. This function is defined in Equation (3) and must be maximized<sup>2</sup>.

<sup>2</sup>A maximization process can easily be transformed into a minimization one

$$Q_{S_s^3} = \sum_{k=1}^K \sum_{x_i \in \pi_k} S_s^3(x_i, m_{\pi_k}). \quad (3)$$

A cluster centroid  $m_{\pi_k}$  is defined by a vector of  $p$  words  $(w_1^{\pi_k}, \dots, w_p^{\pi_k})$ . As a consequence, each cluster centroid must be instantiated in such a way that  $Q_{S_s^3}$  increases at each step of the clustering process. The choice of the best  $p$  words representing each cluster is a way of assuring convergence. For that purpose, we define a procedure which consists in selecting the best  $p$  words from the global vocabulary  $V$  in such a way that  $Q_{S_s^3}$  increases. The global vocabulary is the set of all words which appear in any context vector.

So, for each word  $w \in V$  and any symmetric similarity measure  $S(\cdot, \cdot)$ , its interestingness  $\lambda^k(w)$  is computed as regards to cluster  $\pi_k$ . This operation is defined in Equation (4) where  $s_i \in \pi_k$  is any Web snippet from cluster  $\pi_k$ . Finally, the  $p$  words with higher  $\lambda^k(w)$  are selected to construct the cluster centroid. In such a way, we can easily prove that  $Q_{S_s^3}$  is maximized. Note that a word which is not part of cluster  $\pi_k$  may be part of the centroid  $m_{\pi_k}$ .

$$\lambda^k(w) = \frac{1}{p} \sum_{s_i \in \pi_k} \sum_{w_q^i \in s_i} S(w_q^i, w). \quad (4)$$

Finally, we propose to rely on a modified version of the  $K$ -means algorithm called Global  $K$ -means (Likasa et al., 2003), which has proved to lead to improved results. To solve a clustering problem with  $M$  clusters, all intermediate problems with 1, 2, ...,  $M - 1$  clusters are sequentially solved. The underlying idea is that an optimal solution for a clustering problem with  $M$  clusters can be obtained using a series of local searches using the  $K$ -means algorithm. At each local search, the  $M - 1$  cluster centers are always initially placed at their optimal positions corresponding to the clustering problem with  $M - 1$  clusters. The remaining  $M^{th}$  cluster center is initially placed at several positions within the data space. In addition to effectiveness, the method is deterministic and does not depend on any initial conditions or empirically adjustable parameters. Moreover, its adaptation to PRC is straightforward.

## 3 Stopping Criterion

Once clustering has been processed, selecting the best number of clusters still remains to be decided.

For that purpose, numerous procedures have been proposed (Milligan and Cooper, 1985). However, none of the listed methods were effective or adaptable to our specific problem. So, we proposed a procedure based on the definition of a rational function which models the quality criterion  $Q_{S_s^3}$ . To better understand the behaviour of  $Q_{S_s^3}$  at each step of the adapted  $GK$ -means algorithm, we present its values for  $K = 10$  in Figure 1.

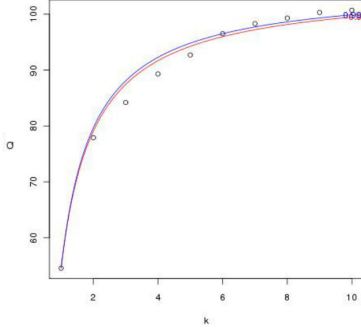


Figure 1:  $Q_{S_s^3}$  and its modelisation.

$Q_{S_s^3}$  can be modelled as in Equation (5) which converges to a limit  $\alpha$  when  $K$  increases and starts from  $Q_{S_s^3}^1$  (i.e.  $Q_{S_s^3}$  at  $K = 1$ ). The underlying idea is that the best number of clusters is given by the  $\beta$  value which maximizes the difference with the average  $\beta^{mean}$ . So,  $\alpha$ ,  $\beta$  and  $\gamma$  need to be expressed independently of unknown variables.

$$\forall K, f(K) = \alpha - \frac{\gamma}{K^\beta}. \quad (5)$$

As  $\alpha$  can theoretically or operationally be defined and it can easily be proved that  $\gamma = \alpha - Q_{S_s^3}^1$ ,  $\beta$  needs to be defined based on  $\gamma$  or  $\alpha$ . This can also be easily proved and the given result is expressed in Equation (6).

$$\beta = \frac{\log(\alpha - Q_{S_s^3}^1) - \log(\alpha - Q_{S_s^3}^K)}{\log(K)}. \quad (6)$$

Now, the value of  $\alpha$  which best approximates the limit of the rational function must be defined. For that purpose, we computed its maximum theoretical and experimental values as well as its approximated maximum experimental value based on the  $\delta^2$ -Aitken (Aitken, 1926) procedure to accelerate convergence as explained in (Kuroda et al., 2008). Best results were obtained with the maximum experimental value which is defined as building the cluster centroid  $m_{\pi_k}$  for each Web

snippet individually. Finally, the best number of clusters is defined as in Algorithm (1) and each one receives its label based on the  $p$  words with greater interestingness of its centroid  $m_{\pi_k}$ .

---

**Algorithm 1** The best  $K$  selection procedure.

---

1. Calculate  $\beta^K$  for each  $K$
  2. Evaluate the mean of all  $\beta^K$  i.e.  $\beta^{mean}$
  3. Select  $\beta^K$  which maximizes  $\beta^K - \beta^{mean}$
  4. Return  $K$  as the best number of partitions
- 

This situation is illustrated in Figure (1) where the red line corresponds to the rational functional for  $\beta^{mean}$  and the blue line models the best  $\beta$  value (i.e. the one which maximizes the difference with  $\beta^{mean}$ ). In this case, the best number would correspond to  $\beta^6$  and as a consequence, the best number of clusters would be 6. In order to illustrate the soundness of the procedure, we present the different values for  $\beta$  at each  $K$  iteration and the differences between consecutive values of  $\beta$  at each iteration in Figure 2. We clearly see that the highest inclination of the curve is between cluster 5 and 6 which also corresponds to the highest difference between two consecutive values of  $\beta$ .

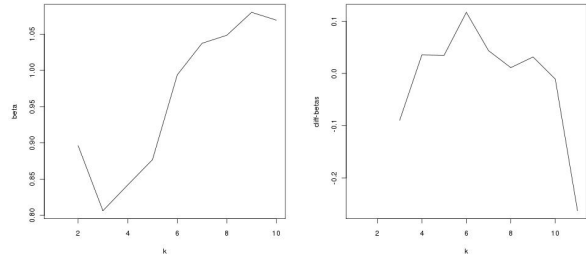


Figure 2: Values of  $\beta$  (on the left) and differences between consecutive values of  $\beta$  (on the right).

## 4 Evaluation

Evaluating PRC systems is a difficult task as stated in (Carpineto et al., 2009). Indeed, a successful PRC system must evidence high quality level clustering. Ideally, each query subtopic should be represented by a unique cluster containing all the relevant Web pages inside. However, this task is far from being achievable. As such, this constraint is reformulated as follows: the task of PRC systems is to provide complete topical cluster coverage of a given query, while avoiding excessive

| $F_{b3}$   |     |   | $K$   |       |              |              |              |       |       |       |       |              | Stop Criterion |  |
|------------|-----|---|-------|-------|--------------|--------------|--------------|-------|-------|-------|-------|--------------|----------------|--|
|            |     |   | 2     | 3     | 4            | 5            | 6            | 7     | 8     | 9     | 10    | $F_{b3}$     | Avg. $K$       |  |
| <i>SCP</i> | $p$ | 2 | 0.387 | 0.396 | 0.398        | 0.396        | 0.391        | 0.386 | 0.382 | 0.378 | 0.374 | 0.395        | 4.799          |  |
|            |     | 3 | 0.400 | 0.411 | 0.412        | 0.409        | 0.406        | 0.400 | 0.397 | 0.391 | 0.388 | 0.411        | 4.690          |  |
|            |     | 4 | 0.405 | 0.416 | 0.423        | 0.425        | 0.423        | 0.420 | 0.416 | 0.414 | 0.411 | 0.441        | 4.766          |  |
|            |     | 5 | 0.408 | 0.422 | <b>0.431</b> | <b>0.431</b> | 0.429        | 0.429 | 0.423 | 0.422 | 0.421 | <b>0.452</b> | 4.778          |  |
| <i>PMI</i> | $p$ | 2 | 0.391 | 0.399 | 0.397        | 0.393        | 0.388        | 0.383 | 0.377 | 0.373 | 0.366 | 0.393        | 4.778          |  |
|            |     | 3 | 0.408 | 0.418 | 0.422        | 0.418        | 0.414        | 0.410 | 0.405 | 0.398 | 0.392 | 0.416        | 4.879          |  |
|            |     | 4 | 0.420 | 0.434 | 0.439        | 0.439        | 0.435        | 0.430 | 0.425 | 0.420 | 0.412 | 0.436        | 4.874          |  |
|            |     | 5 | 0.423 | 0.444 | <b>0.451</b> | <b>0.451</b> | <b>0.451</b> | 0.445 | 0.441 | 0.434 | 0.429 | <b>0.450</b> | 4.778          |  |

Table 1:  $F_{b3}$  for *SCP* and *PMI* for the global search and the stopping criterion for the ODP-239 dataset.

|          |          | Adapted <i>GK</i> -means |       |              |              |              |            |       |              |       |       | STC   | LINGO | BIK | OPTIMSRC |
|----------|----------|--------------------------|-------|--------------|--------------|--------------|------------|-------|--------------|-------|-------|-------|-------|-----|----------|
|          |          | <i>SCP</i>               |       |              |              |              | <i>PMI</i> |       |              |       |       |       |       |     |          |
|          |          | $p$                      |       |              |              |              | $p$        |       |              |       |       |       |       |     |          |
| ODP-239  | $F_1$    | 0.312                    | 0.341 | 0.352        | 0.366        | 0.332        | 0.358      | 0.378 | <b>0.390</b> | 0.324 | 0.273 | 0.200 | 0.313 |     |          |
|          | $F_2$    | 0.363                    | 0.393 | 0.404        | 0.416        | 0.363        | 0.395      | 0.421 | <b>0.435</b> | 0.319 | 0.167 | 0.173 | 0.341 |     |          |
|          | $F_5$    | 0.411                    | 0.441 | 0.453        | 0.462        | 0.390        | 0.430      | 0.459 | <b>0.476</b> | 0.322 | 0.153 | 0.165 | 0.380 |     |          |
|          | $F_{b3}$ | 0.395                    | 0.411 | 0.441        | <b>0.452</b> | 0.393        | 0.416      | 0.436 | 0.450        | 0.403 | 0.346 | 0.307 | N/A   |     |          |
| MORESQUE | $F_1$    | 0.627                    | 0.649 | <b>0.665</b> | 0.664        | 0.615        | 0.551      | 0.543 | 0.571        | 0.455 | 0.326 | 0.317 | N/A   |     |          |
|          | $F_2$    | 0.685                    | 0.733 | 0.767        | <b>0.770</b> | 0.644        | 0.548      | 0.521 | 0.551        | 0.392 | 0.260 | 0.269 | N/A   |     |          |
|          | $F_5$    | 0.747                    | 0.817 | 0.865        | <b>0.872</b> | 0.679        | 0.563      | 0.519 | 0.553        | 0.370 | 0.237 | 0.255 | N/A   |     |          |
|          | $F_{b3}$ | 0.482                    | 0.482 | 0.473        | 0.464        | <b>0.490</b> | 0.465      | 0.462 | 0.485        | 0.460 | 0.399 | 0.315 | N/A   |     |          |

Table 2: PRC comparative results for  $F_\beta$  and  $F_{b3}$  over the ODP-239 and MORESQUE datasets.

redundancy of the subtopics in the result list of clusters. So, in order to evaluate our methodology, we propose two different evaluations. First, we want to evidence the quality of the stopping criterion when compared to an exhaustive search over all tunable parameters. Second, we propose a comparative evaluation with existing state-of-the-art algorithms over gold standard datasets and recent clustering evaluation metrics.

#### 4.1 Text Processing

Before the clustering process takes place, Web snippets are represented as word feature vectors. In order to define the set of word features, the Web service proposed in (Machado et al., 2009) is used<sup>3</sup>. In particular, it assigns a relevance score to any token present in the set of retrieved Web snippets based on the analysis of left and right token contexts. A specific threshold is then applied to withdraw irrelevant tokens and the remaining ones form the vocabulary  $V$ . Then, each Web snippet is represented by the set of its  $p$  most relevant tokens in the sense of the  $W(\cdot)$  value proposed in (Machado et al., 2009). Note that within the proposed Web service, multiword units are also identified. They are exclusively composed of relevant individual tokens and their weight is given by the arithmetic mean of their constituents scores.

<sup>3</sup>Access to this Web service is available upon request.

#### 4.2 Intrinsic Evaluation

The first set of experiments focuses on understanding the behaviour of our methodology within a greedy search strategy for different tunable parameters defined as a tuple  $\langle p, K, S(W_{ik}, W_{jl}) \rangle$ . In particular,  $p$  is the size of the word feature vectors representing both Web snippets and centroids ( $p = 2..5$ ),  $K$  is the number of clusters to be found ( $K = 2..10$ ) and  $S(W_{ik}, W_{jl})$  is the collocation measure integrated in the InfoSimba similarity measure. In these experiments, two association measures which are known to have different behaviours (Pecina and Schlesinger, 2006) are tested. We implement the Symmetric Conditional Probability (Silva et al., 1999) in Equation (7) which tends to give more credits to frequent associations and the Pointwise Mutual Information (Church and Hanks, 1990) in Equation (8) which over-estimates infrequent associations. Then, best  $\langle p, K, S(W_{ik}, W_{jl}) \rangle$  configurations are compared to our stopping criterion.

$$SCP(W_{ik}, W_{jl}) = \frac{P(W_{ik}, W_{jl})^2}{P(W_{ik}) \times P(W_{jl})}. \quad (7)$$

$$PMI(W_{ik}, W_{jl}) = \log_2 \frac{P(W_{ik}, W_{jl})}{P(W_{ik}) \times P(W_{jl})}. \quad (8)$$

In order to perform this task, we evaluate performance based on the  $F_{b3}$  measure defined in (Amigó et al., 2009) over the ODP-239 gold standard dataset proposed in (Carpineto and Romano,

2010). In particular, (Amigó et al., 2009) indicate that common metrics such as the  $F_\beta$ -measure are good to assign higher scores to clusters with high homogeneity, but fail to evaluate cluster completeness. First results are provided in Table 1 and evidence that the best configurations for different  $\langle p, K, S(W_{ik}, W_{jl}) \rangle$  tuples are obtained for high values of  $p$ ,  $K$  ranging from 4 to 6 clusters and  $PMI$  steadily improving over  $SCP$ . However, such a fuzzy configuration is not satisfactory. As such, we proposed a new stopping criterion which evidences coherent results as it (1) does not depend on the used association measure ( $F_{b3}^{SCP} = 0.452$  and  $F_{b3}^{PMI} = 0.450$ ), (2) discovers similar numbers of clusters independently of the length of the  $p$ -context vector and (3) increases performance with high values of  $p$ .

### 4.3 Comparative Evaluation

The second evaluation aims to compare our methodology to current state-of-the-art text-based PRC algorithms. We propose comparative experiments over two gold standard datasets (ODP-239 (Carpineto and Romano, 2010) and MORESQUE (Di Marco and Navigli, 2013)) for STC (Zamir and Etzioni, 1998), LINGO (Osinski and Weiss, 2005), OPTIMSRC (Carpineto and Romano, 2010) and the Bisecting Incremental  $K$ -means (BIK) which may be seen as a baseline for the polythetic paradigm. A brief description of each PRC algorithm is given as follows.

**STC:** (Zamir and Etzioni, 1998) defined the Suffix Tree Clustering algorithm which is still a difficult standard to beat in the field. In particular, they propose a monothetic clustering technique which merges base clusters with high string overlap. Indeed, instead of using the classical Vector Space Model (VSM) representation, they propose to represent Web snippets as compact tries.

**LINGO:** (Osinski and Weiss, 2005) proposed a polythetic solution called LINGO which takes into account the string representation proposed by (Zamir and Etzioni, 1998). They first extract frequent phrases based on suffix-arrays. Then, they reduce the term-document matrix (defined as a VSM) using Single Value Decomposition to discover latent structures. Finally, they match group descriptions with the extracted topics and assign relevant documents to them.

**OPTIMSRC:** (Carpineto and Romano, 2010) showed that the characteristics of the outputs returned by PRC algorithms suggest the adoption of a meta clustering approach. As such, they introduce a novel criterion to measure the concordance of two partitions of objects into different clusters based on the information content associated to the series of decisions made by the partitions on single pairs of objects. Then, the meta clustering phase is casted to an optimization problem of the concordance between the clustering combination and the given set of clusterings.

With respect to implementation, we used the Carrot2 APIs<sup>4</sup> which are freely available for STC, LINGO and the classical BIK. It is worth noticing that all implementations in Carrot2 are tuned to extract exactly 10 clusters. For OPTIMSRC, we reproduced the results presented in the paper of (Carpineto and Romano, 2010) as no implementation is freely available. The results are illustrated in Table 2 including both  $F_\beta$ -measure and  $F_{b3}$ . They evidence clear improvements of our methodology when compared to state-of-the-art text-based PRC algorithms, over both datasets and all evaluation metrics. But more important, even when the  $p$ -context vector is small ( $p = 3$ ), the adapted  $GK$ -means outperforms all other existing text-based PRC which is particularly important as they need to perform in real-time.

## 5 Conclusions

In this paper, we proposed a new PRC approach which (1) is based on the adaptation of the  $K$ -means algorithm to third-order similarity measures and (2) proposes a coherent stopping criterion. Results evidenced clear improvements over the evaluated state-of-the-art text-based approaches for two gold standard datasets. Moreover, our best  $F_1$ -measure over ODP-239 (0.390) approximates the highest ever-reached  $F_1$ -measure (0.413) by the TOPICAL knowledge-driven algorithm proposed in (Scaiella et al., 2012)<sup>5</sup>. These results are promising and in future works, we propose to define new knowledge-based third-order similarity measures based on studies in entity-linking (Ferragina and Scaiella, 2010).

<sup>4</sup><http://search.carrot2.org/stable/search> [Last access: 15/05/2013].

<sup>5</sup>Notice that the authors only propose the  $F_1$ -measure although different results can be obtained for different  $F_\beta$ -measures and  $F_{b3}$  as evidenced in Table 2.

## References

- A.C. Aitken. 1926. On bernoulli's numerical solution of algebraic equations. *Research Society Edinburgh*, 46:289–305.
- E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.
- C. Carpineto and G. Romano. 2010. Optimal meta search results clustering. In *33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 170–177.
- C. Carpineto, S. Osinski, G. Romano, and D. Weiss. 2009. A survey of web clustering engines. *ACM Computer Survey*, 41(3):1–38.
- K. Church and P. Hanks. 1990. Word association norms mutual information and lexicography. *Computational Linguistics*, 16(1):23–29.
- A. Di Marco and R. Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(4):1–43.
- G. Dias, E. Alves, and J.G.P. Lopes. 2007. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *Proceedings of 22nd Conference on Artificial Intelligence (AAAI)*, pages 1334–1339.
- P. Ferragina and A. Gulli. 2008. A personalized search engine based on web-snippet hierarchical clustering. *Software: Practice and Experience*, 38(2):189–225.
- P. Ferragina and U. Scaiella. 2010. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1625–1628.
- M. Kuroda, M. Sakakihara, and Z. Geng. 2008. Acceleration of the em and ecm algorithms using the aitken  $\delta^2$  method for log-linear models with partially classified data. *Statistics & Probability Letters*, 78(15):2332–2338.
- A. Likasa, Vlassis. N., and J. Verbeek. 2003. The global k-means clustering algorithm. *Pattern Recognition*, 36:451–461.
- S.P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.
- D. Machado, T. Barbosa, S. Pais, B. Martins, and G. Dias. 2009. Universal mobile information retrieval. In *Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction (HCI)*, pages 345–354.
- R. Mihalcea, C. Corley, and C. Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 775–780.
- G.W. Milligan and M.C. Cooper. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.
- R. Navigli and G. Crisafulli. 2010. Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 116–126.
- S. Osinski and D. Weiss. 2005. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54.
- P. Pecina and P. Schlesinger. 2006. Combining association measures for collocation extraction. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL)*, pages 651–658.
- U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. 2012. Topical clustering of search results. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 223–232.
- J. Silva, G. Dias, S. Guilloré, and J.G.P. Lopes. 1999. Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. In *Proceedings of 9th Portuguese Conference in Artificial Intelligence (EPIA)*, pages 113–132.
- M. Timonen. 2013. *Term Weighting in Short Documents for Document Categorization, Keyword Extraction and Query Expansion*. Ph.D. thesis, University of Helsinki, Finland.
- O. Zamir and O. Etzioni. 1998. Web document clustering: A feasibility demonstration. In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 46–54.

# Automatic Coupling of Answer Extraction and Information Retrieval

**Xuchen Yao** and **Benjamin Van Durme**  
Johns Hopkins University  
Baltimore, MD, USA

**Peter Clark**  
Vulcan Inc.  
Seattle, WA, USA

## Abstract

Information Retrieval (IR) and Answer Extraction are often designed as isolated or loosely connected components in Question Answering (QA), with repeated over-engineering on IR, and not necessarily performance gain for QA. We propose to tightly integrate them by coupling automatically learned features for answer extraction to a shallow-structured IR model. Our method is very quick to implement, and significantly improves IR for QA (measured in Mean Average Precision and Mean Reciprocal Rank) by 10%-20% against an uncoupled retrieval baseline in both document and passage retrieval, which further leads to a downstream 20% improvement in QA  $F_1$ .

## 1 Introduction

The overall performance of a Question Answering system is bounded by its Information Retrieval (IR) front end, resulting in research specifically on Information Retrieval for Question Answering (IR4QA) (Greenwood, 2008; Sakai et al., 2010). Common approaches such as query expansion, structured retrieval, and translation models show patterns of complicated engineering on the IR side, or isolate the upstream passage retrieval from downstream answer extraction. We argue that: 1. an IR front end should deliver exactly what a QA<sup>1</sup> back end needs; 2. many intuitions employed by QA should be and can be *re-used* in IR, rather than *re-invented*. We propose a coupled retrieval method with prior knowledge of its downstream QA component, that feeds QA with exactly the information needed.

<sup>1</sup>After this point in the paper we use the term QA in a narrow sense: QA without the IR component, i.e., answer extraction.

As a motivating example, using the question *When was Alaska purchased from the TREC 2002 QA track as the query to the Indri search engine, the top sentence retrieved from the accompanying AQUAINT corpus is:*

Eventually Alaska Airlines will allow all travelers who have purchased electronic tickets through any means.

While this relates *Alaska* and *purchased*, it is not a useful passage for the given question.<sup>2</sup> It is apparent that the question asks for a date. Prior work proposed predictive annotation (Prager et al., 2000; Prager et al., 2006): text is first annotated in a predictive manner (of what types of questions it might answer) with 20 answer types and then indexed. A question analysis component (consisting of 400 question templates) maps the desired answer type to one of the 20 existing answer types. Retrieval is then performed with both the question and predicated answer types in the query.

However, predictive annotation has the limitation of being labor intensive and assuming the underlying NLP pipeline to be accurate. We avoid these limitations by directly asking the downstream QA system for the information about *which entities answer which questions*, via two steps: 1. reusing the question analysis components from QA; 2. forming a query based on the most relevant answer features given a question from the learned QA model. There is no query-time overhead and no manual template creation. Moreover, this approach is more robust against, e.g., entity recognition errors, because answer typing knowledge is learned from how the data was *actually* labeled, not from how the data was *assumed* to be labeled (e.g., manual templates usually assume perfect labeling of named entities, but often it is not the case

<sup>2</sup>Based on a non-optimized IR configuration, none of the top 1000 returned passages contained the correct answer: 1867.

in practice).

We use our statistically-trained QA system (Yao et al., 2013) that recognizes the association between question type and expected answer types through various features. The QA system employs a linear chain Conditional Random Field (CRF) (Lafferty et al., 2001) and tags each token as either an answer (ANS) or not (O). This will be our off-the-shelf QA system, which recognizes the association between question type and expected answer types through various features based on e.g., part-of-speech tagging (POS) and named entity recognition (NER).

With weights optimized by CRF training (Table 1), we can learn how answer features are correlated with question features. These features, whose weights are optimized by the CRF training, directly reflect what the most important answer types associated with each question type are. For instance, line 2 in Table 1 says that if there is a *when* question, and the current token’s NER label is DATE, then it is likely that this token is tagged as ANS. IR can easily make use of this knowledge: for a *when* question, IR retrieves sentences with tokens labeled as DATE by NER, or POS tagged as CD. The only extra processing is to pre-tag and index the text with POS and NER labels. The analyzing power of discriminative answer features for IR comes *for free* from a trained QA system. Unlike predictive annotation, statistical evidence determines the best answer features given the question, with no manual pattern or templates needed.

To compare again predictive annotation with our approach: predictive annotation works in a *forward* mode, downstream QA is tailored for upstream IR, i.e., QA works on whatever IR retrieves. Our method works in reverse (*backward*): downstream QA dictates upstream IR, i.e., IR retrieves what QA wants. Moreover, our approach extends easily beyond fixed *answer types* such as named entities: we are already using POS tags as a demonstration. We can potentially use any helpful *answer features* in retrieval. For instance, if the QA system learns that *in order to* is highly correlated with *why* question through lexicalized features, or some certain dependency relations are helpful in answering questions with specific structures, then it is natural and easy for the IR component to incorporate them.

There is also a distinction between our method and the technique of *learning to rank* applied in

| feature                           | label | weight |
|-----------------------------------|-------|--------|
| qword=when POS <sub>0</sub> =CD   | ANS   | 0.86   |
| qword=when NER <sub>0</sub> =DATE | ANS   | 0.79   |
| qword=when POS <sub>0</sub> =CD   | O     | -0.74  |

Table 1: Learned weights for sampled features with respect to the label of *current* token (indexed by [0]) in a CRF. The larger the weight, the more “important” is this feature to help tag the current token with the corresponding label. For instance, line 1 says when answering a *when* question, and the POS of current token is CD (cardinal number), it is likely (large weight) that the token is tagged as ANS.

QA (Bilotti et al., 2010; Agarwal et al., 2012). Our method is a *QA-driven* approach that provides supervision for IR from a learned QA model, while learning to rank is essentially an *IR-driven* approach: the supervision for IR comes from a labeled ranking list of retrieval results.

Overall, we make the following contributions:

- Our proposed method tightly integrates QA with IR and the reuse of analysis from QA does not put extra overhead on the IR queries. This QA-driven approach provides a holistic solution to the task of IR4QA.
- We learn statistical evidence about what the form of answers to different questions look like, rather than using manually authored templates. This provides great flexibility in using answer features in IR queries.

We give a full spectrum evaluation of all three stages of IR+QA: document retrieval, passage retrieval and answer extraction, to examine thoroughly the effectiveness of the method.<sup>3</sup> All of our code and datasets are publicly available.<sup>4</sup>

## 2 Background

Besides Predictive Annotation, our work is closest to structured retrieval, which covers techniques of dependency path mapping (Lin and Pantel, 2001; Cui et al., 2005; Kaisser, 2012), graph matching with Semantic Role Labeling (Shen and Lapata, 2007) and answer type checking (Pinchak et al., 2009), etc. Specifically, Bilotti et al. (2007) proposed indexing text with their semantic roles and named entities. Queries then include constraints of semantic roles and named entities for the predicate and its arguments in the question. Improvements in recall of answer-bearing sentences were shown over the bag-of-words baseline. Zhao and

<sup>3</sup>Rarely are all three aspects presented in concert (see §2).

<sup>4</sup><http://code.google.com/p/jacana/>



Callan (2008) extended this work with approximate matching and smoothing. Most research uses parsing to assign deep structures. Compared to shallow (POS, NER) structured retrieval, deep structures need more processing power and smoothing, but might also be more precise.<sup>5</sup>

Most of the above (except Kaisser (2012)) only reported on IR or QA, but not both, assuming that improvement in one naturally improves the other. Bilotti and Nyberg (2008) challenged this assumption and called for tighter coupling between IR and QA. This paper is aimed at that challenge.

### 3 Method

Table 1 already shows some examples of features associating question types with answer types. We store the features and their learned weights from the trained model for IR usage.

We let the trained QA system guide the query formulation when performing coupled retrieval with Indri (Strohman et al., 2005), given a corpus already annotated with POS tags and NER labels. Then retrieval runs in four steps (Figure 1):

1. Question Analysis. The question analysis component from QA is reused here. In this implementation, the only information we have chosen to use from the question is the question word (e.g., *how*, *who*) and the lexical answer types (LAT) in case of *what/which* questions.
2. Answer Feature Selection. Given the question word, we select the 5 highest weighted features (e.g.,  $\text{POS}[0]=\text{CD}$  for a *when* question).
3. Query Formulation. The original question is combined with the top features as the query.
4. Coupled Retrieval. Indri retrieves a ranked list of documents or passages.

As motivated in the introduction, this framework is aimed at providing the following benefits:

**Reuse of QA components on the IR side.** IR reuses both code for question analysis and top weighted features from QA.

**Statistical selection of answer features.** For instance, the NER tagger we used divides location into two categories: GPE (geo locations) and LOC

(non-GPE). Both of them are learned to be important to *where* questions.

**Error tolerance along the NLP pipeline.** IR and QA share the same processing pipeline. Systematic errors made by the processing tools are tolerated, in the sense that if the same pre-processing error is made on both the question and sentence, an answer may still be found. Take the previous *where* question, besides  $\text{NER}[0]=\text{GPE}$  and  $\text{NER}[0]=\text{LOC}$ , we also found oddly  $\text{NER}[0]=\text{PERSON}$  an important feature, due to that the NER tool sometimes mistakes PERSON for LOC. For instance, the volcano name Mauna Loa is labeled as a PERSON instead of a LOC. But since the importance of this feature is recognized by downstream QA, the upstream IR is still motivated to retrieve it.

Queries were lightly optimized using the following strategies:

**Query Weighting** In practice query words are weighted:

```
#weight(1.0 When 1.0 was 1.0 Alaska 1.0 purchased
 $\alpha$  #max(#any:CD #any:DATE))
```

with a weight  $\alpha$  for the answer types tuned via cross-validation.

Since NER and POS tags are not lexicalized they accumulate many more counts (i.e. term frequency) than individual words, thus we in general downweight by setting  $\alpha < 1.0$ , giving the expected answer types “enough say” but not “too much say”:

**NER Types First** We found NER labels better indicators of expected answer types than POS tags. The reasons are two-fold: 1. In general POS tags are too coarse-grained in answer types than NER labels. E.g., NNP can answer *who* and *where* questions, but is not as precise as PERSON and GPE. 2. POS tags accumulate even more counts than NER labels, thus they need separate downweighting. *Learning* the interplay of these weights in a joint IR/QA model, is an interesting path for future work. If the top-weighted features are based on NER, then we do not include POS tags for that question. Otherwise POS tags are useful, for instance, in answering *how* questions.

**Unigram QA Model** The QA system uses *up to* trigram features (Table 1 shows examples of unigram and bigram features). Thus it is able to learn, for instance, that a POS sequence of IN CD NNS is likely an answer to a *when* question (such as: *in 5 years*). This requires that the IR queries

<sup>5</sup>Ogilvie (2010) showed in chapter 4.3 that keyword and named entities based retrieval actually outperformed SRL-based structured retrieval in MAP for the answer-bearing sentence retrieval task in their setting. In this paper we do not intend to re-invent another parse-based structure matching algorithm, but only use shallow structures to show the idea of coupling QA with IR; in the future this might be extended to incorporate “deeper” structure.

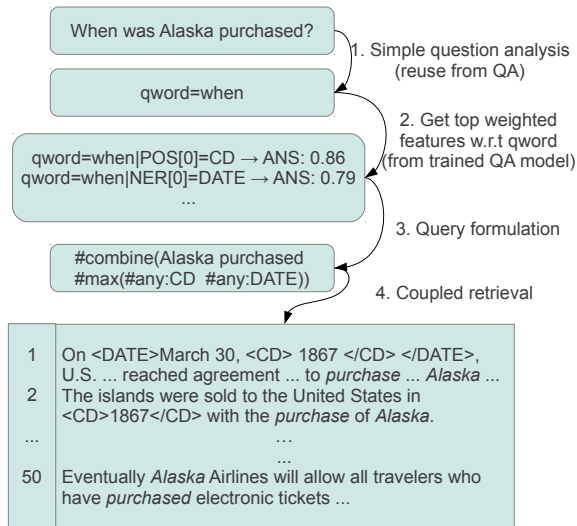


Figure 1: Coupled retrieval with queries directly constructed from highest weighted features of downstream QA. The retrieved and ranked list of sentences is POS and NER tagged, but only query-relevant tags are shown due to space limit. A bag-of-words retrieval approach would have the sentence shown above at rank 50 at its top position instead.

look for a consecutive IN CD NNS sequence. We drop this strict constraint (which may need further smoothing) and only use unigram features, not by simply extracting “good” unigram features from the trained model, but by re-training the model with only unigram features. In answer extraction, we still use up to trigram features.<sup>6</sup>

## 4 Experiments

We want to measure and compare the performance of the following retrieval techniques:

1. *uncoupled retrieval* with an off-the-shelf IR engine by using the question as query (baseline),
2. *QA-driven coupled retrieval* (proposed), and
3. *answer-bearing retrieval* by using both the question and known answer as query, only evaluated for answer extraction (upper bound),

at the three stages of question answering:

1. Document retrieval (for relevant docs from corpus), measured by Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).
2. Passage retrieval (finding relevant sentences from the document), also by MAP and MRR.
3. Answer extraction, measured by  $F_1$ .

<sup>6</sup>This is because the weights of unigram to trigram features in a loglinear CRF model is a balanced consequence for maximization. A unigram feature might end up with lower weight because another trigram containing this unigram gets a higher weight. Then we would have missed this feature if we only used top unigram features. Thus we re-train the model with only unigram features to make sure weights are “assigned properly” among only unigram features.

| set                  | questions |            | sentences |            |
|----------------------|-----------|------------|-----------|------------|
|                      | #all      | #pos.      | #all      | #pos.      |
| TRAIN                | 2205      | 1756 (80%) | 22043     | 7637 (35%) |
| TEST <sub>gold</sub> | 99        | 88 (89%)   | 990       | 368 (37%)  |

Table 2: Statistics for AMT-collected data (total cost was around \$800 for paying three Turkers per sentence). Positive questions are those with an answer found. Positive sentences are those bearing an answer.

All coupled and uncoupled queries are performed with Indri v5.3 (Strohman et al., 2005).

### 4.1 Data

**Test Set for IR and QA** The MIT109 test collection by Lin and Katz (2006) contains 109 questions from TREC 2002 and provides a near-exhaustive judgment of relevant documents for each question. We removed 10 questions that do not have an answer by matching the TREC answer patterns. Then we call this test set **MIT99**.

**Training Set for QA** We used Amazon Mechanical Turk to collect training data for the QA system by issuing answer-bearing queries for TREC1999-2003 questions. For the top 10 retrieved sentences for each question, three Turkers judged whether each sentence contained the answer. The inter-coder agreement rate was 0.81 (Krippendorff, 2004; Artstein and Poesio, 2008).

The 99 questions of MIT99 were extracted from the Turk collection as our TEST<sub>gold</sub> with the remaining as TRAIN, with statistics shown in Table 2. Note that only 88 questions out of MIT99 have an answer from the top 10 query results.

Finally both the training and test data were sentence-segmented and word-tokenized by NLTK (Bird and Loper, 2004), dependency-parsed by the Stanford Parser (Klein and Manning, 2003), and NER-tagged by the Illinois Named Entity Tagger (Ratinov and Roth, 2009) with an 18-label type set.

**Corpus Preprocessing for IR** The AQUAINT (LDC2002T31) corpus, on which the MIT99 questions are based, was processed in exactly the same manner as was the QA training set. But only sentence boundaries, POS tags and NER labels were kept as the annotation of the corpus.

### 4.2 Document and Passage Retrieval

We issued uncoupled queries consisting of question words, and QA-driven coupled queries consisting of both the question and expected answer types, then retrieved the top 1000 documents, and

| type     | coupled       |               | uncoupled |        |
|----------|---------------|---------------|-----------|--------|
|          | MAP           | MRR           | MAP       | MRR    |
| document | <b>0.2524</b> | <b>0.4835</b> | 0.2110    | 0.4298 |
| sentence | <b>0.1375</b> | <b>0.2987</b> | 0.1200    | 0.2544 |

Table 3: Coupled vs. uncoupled document/sentence retrieval in MAP and MRR on MIT99. Significance level (Smucker et al., 2007) for both MAP:  $p < 0.001$  and for both MRR:  $p < 0.05$ .

finally computed MAP and MRR against the gold-standard MIT99 per-document judgment.

To find the best weighting  $\alpha$  for coupled retrieval, we used 5-fold cross-validation and finalized at  $\alpha = 0.1$ . Table 3 shows the results. Coupled retrieval outperforms (20% by MAP with  $p < 0.001$  and 12% by MRR with  $p < 0.01$ ) uncoupled retrieval significantly according to paired randomization test (Smucker et al., 2007).

For passage retrieval, we extracted relevant single sentences. Recall that MIT99 only contains document-level judgment. To generate a test set for sentence retrieval, we matched each sentence from relevant documents provided by MIT99 for each question against the TREC answer patterns.

We found no significant difference between retrieving sentences from the documents returned by document retrieval or directly from the corpus. Numbers of the latter are shown in Table 3. Still, coupled retrieval is significantly better by about 10% in MAP and 17% in MRR.

### 4.3 Answer Extraction

Lastly we sent the sentences to the downstream QA engine (trained on TRAIN) and computed  $F_1$  per  $K$  for the top  $K$  retrieved sentences,<sup>7</sup> shown in Figure 2. The best  $F_1$  with coupled sentence retrieval is 0.231, 20% better than  $F_1$  of 0.192 with uncoupled retrieval, both at  $K = 1$ .

The two descending lines at the bottom reflect the fact that the majority-voting mechanism from the QA system was too simple:  $F_1$  drops as  $K$  increases. Thus we also computed  $F_1$ 's assuming perfect voting: a voting oracle that always selects the correct answer as long as the QA system produces one, thus the two ascending lines in the center of Figure 2. Still,  $F_1$  with coupled retrieval is always better: reiterating the fact that coupled retrieval covers more answer-bearing sentences.

<sup>7</sup>Lin (2007), Zhang et al. (2007), and Kaisser (2012) also evaluated on MIT109. However their QA engines used web-based search engines, thus leading to results that are neither reproducible nor directly comparable with ours.

Finally, to find the upper bound for QA, we drew the two upper lines, testing on TEST<sub>gold</sub> described in Table 2. The test sentences were obtained with answer-bearing queries. This is assuming almost perfect IR. The gap between the top two and other lines signals more room for improvements for IR in terms of better coverage and better rank for answer-bearing sentences.

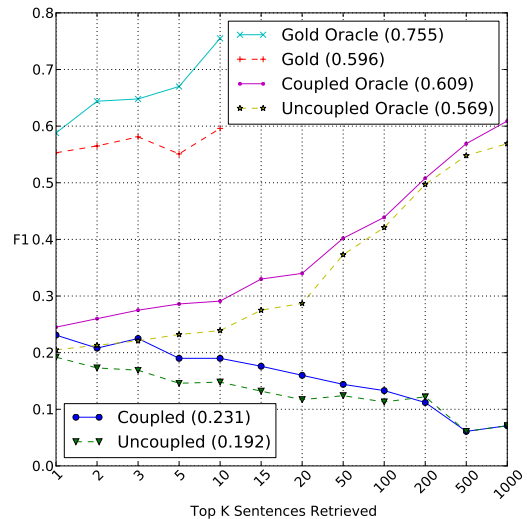


Figure 2:  $F_1$  values for answer extraction on MIT99. Best  $F_1$ 's for each method are parenthesized in the legend. “Oracle” methods assumed perfect voting of answer candidates (a question is answered correctly if the system ever produced one correct answer for it). “Gold” was tested on TEST<sub>gold</sub>.

## 5 Conclusion

We described a method to perform coupled information retrieval with a prior knowledge of the downstream QA system. Specifically, we coupled IR queries with automatically learned answer features from QA and observed significant improvements in document/passage retrieval and boosted  $F_1$  in answer extraction. This method has the merits of not requiring hand-built question and answer templates and being flexible in incorporating various answer features automatically learned and optimized from the downstream QA system.

## Acknowledgement

We thank Vulcan Inc. for funding this work. We also thank Paul Ogilvie, James Mayfield, Paul McNamee, Jason Eisner and the three anonymous reviewers for insightful comments.

## References

- Arvind Agarwal, Hema Raghavan, Karthik Subbian, Prem Melville, Richard D. Lawrence, David C. Gondek, and James Fan. 2012. Learning to rank for robust question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12*, pages 833–842, New York, NY, USA. ACM.
- Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.
- M.W. Bilotti and E. Nyberg. 2008. Improving text retrieval precision and answer accuracy in question answering systems. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 1–8.
- M.W. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg. 2007. Structured retrieval for question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 351–358. ACM.
- M.W. Bilotti, J. Elsas, J. Carbonell, and E. Nyberg. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 459–468. ACM.
- Steven Bird and Edward Loper. 2004. Nltk: The natural language toolkit. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 214–217, Barcelona, Spain, July.
- Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '05*, pages 400–407, New York, NY, USA. ACM.
- Mark A. Greenwood, editor. 2008. *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*. Coling 2008 Organizing Committee, Manchester, UK, August.
- Michael Kaisser. 2012. Answer Sentence Retrieval by Matching Dependency Paths acquired from Question/Answer Sentence Pairs. In *EACL*, pages 88–98.
- Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *In Proc. the 41st Annual Meeting of the Association for Computational Linguistics*.
- Klaus H. Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Inc, 2nd edition.
- 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.
- J. Lin and B. Katz. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861.
- D. Lin and P. Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.
- Jimmy Lin. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2), April.
- P. Ogilvie. 2010. *Retrieval using Document Structure and Annotations*. Ph.D. thesis, Carnegie Mellon University.
- Christopher Pinchak, Davood Rafiei, and Dekang Lin. 2009. Answer typing for information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management, CIKM '09*, pages 1955–1958, New York, NY, USA. ACM.
- John Prager, Eric Brown, Anni Coden, and Dragomir Radev. 2000. Question-answering by predictive annotation. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '00*, pages 184–191, New York, NY, USA. ACM.
- J. Prager, J. Chu-Carroll, E. Brown, and K. Czuba. 2006. Question answering by predictive annotation. *Advances in Open Domain Question Answering*, pages 307–347.
- L. Ratnov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6.
- Tetsuya Sakai, Hideki Shima, Noriko Kando, Ruihua Song, Chuan-Jie Lin, Teruko Mitamura, Miho Sugimoto, and Cheng-Wei Lee. 2010. Overview of the ntcir-7 aclia ir4qa task. In *Proceedings of NTCIR-8 Workshop Meeting*, Tokyo, Japan.
- D. Shen and M. Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of EMNLP-CoNLL*, pages 12–21.
- M.D. Smucker, J. Allan, and B. Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM.

- T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, pages 2–6. Citeseer.
- Xuchen Yao, Benjamin Van Durme, Peter Clark, and Chris Callison-Burch. 2013. Answer Extraction as Sequence Tagging with Tree Edit Distance. In *Proceedings of NAACL 2013*.
- Xian Zhang, Yu Hao, Xiaoyan Zhu, Ming Li, and David R. Cheriton. 2007. Information distance from a question to an answer. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07*, pages 874–883, New York, NY, USA. ACM.
- L. Zhao and J. Callan. 2008. A generative retrieval model for structured documents. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1163–1172. ACM.

# An Improved MDL-Based Compression Algorithm for Unsupervised Word Segmentation

Ruey-Cheng Chen

National Taiwan University

1 Roosevelt Rd. Sec. 4

Taipei 106, Taiwan

rueycheng@turing.csie.ntu.edu.tw

## Abstract

We study the mathematical properties of a recently proposed MDL-based unsupervised word segmentation algorithm, called regularized compression. Our analysis shows that its objective function can be efficiently approximated using the negative empirical pointwise mutual information. The proposed extension improves the baseline performance in both efficiency and accuracy on a standard benchmark.

## 1 Introduction

Hierarchical Bayes methods have been mainstream in unsupervised word segmentation since the dawn of hierarchical Dirichlet process (Goldwater et al., 2009) and adaptors grammar (Johnson and Goldwater, 2009). Despite this wide recognition, they are also notoriously computational prohibitive and have limited adoption on larger corpora. While much effort has been directed to mitigating this issue within the Bayes framework (Borschinger and Johnson, 2011), many have found minimum description length (MDL) based methods more promising in addressing the scalability problem.

MDL-based methods (Rissanen, 1978) rely on underlying search algorithms to segment the text in as many possible ways and use description length to decide which to output. As different algorithms explore different trajectories in the search space, segmentation accuracy depends largely on the search coverage. Early work in this line focused more on existing segmentation algorithm, such as branching entropy (Tanaka-Ishii, 2005; Zhikov et al., 2010) and bootstrap voting experts (Hewlett and Cohen, 2009; Hewlett and Cohen, 2011). A recent study (Chen et al., 2012) on a compression-based algorithm, *regularized compression*, has achieved comparable performance result to hierarchical Bayes methods.

Along this line, in this paper we present a novel extension to the regularized compressor algorithm. We propose a lower-bound approximate to the original objective and show that, through analysis and experimentation, this amendment improves segmentation performance and runtime efficiency.

## 2 Regularized Compression

The dynamics behind regularized compression is similar to digram coding (Witten et al., 1999). One first breaks the text down to a sequence of characters ( $W_0$ ) and then works from that representation up in an agglomerative fashion, iteratively removing word boundaries between the two selected word types. Hence, a new sequence  $W_i$  is created in the  $i$ -th iteration by merging all the occurrences of some selected bigram  $(x, y)$  in the original sequence  $W_{i-1}$ . Unlike in digram coding, where the most frequent pair of word types is always selected, in regularized compression a specialized decision criterion is used to balance compression rate and vocabulary complexity:

$$\begin{aligned} \min. \quad & -\alpha f(x, y) + |W_{i-1}| \Delta \tilde{H}(W_{i-1}, W_i) \\ \text{s.t.} \quad & \text{either } x \text{ or } y \text{ is a character} \\ & f(x, y) > n_{ms}. \end{aligned}$$

Here, the criterion is written slightly differently. Note that  $f(x, y)$  is the bigram frequency,  $|W_{i-1}|$  the sequence length of  $W_{i-1}$ , and  $\Delta \tilde{H}(W_{i-1}, W_i) = \tilde{H}(W_i) - \tilde{H}(W_{i-1})$  is the difference between the empirical Shannon entropy measured on  $W_i$  and  $W_{i-1}$ , using maximum likelihood estimates. Specifically, this empirical estimate  $\tilde{H}(W)$  for a sequence  $W$  corresponds to:

$$\log |W| - \frac{1}{|W|} \sum_{x:\text{types}} f(x) \log f(x).$$

For this equation to work, one needs to estimate other model parameters. See Chen et al. (2012) for a comprehensive treatment.

|           |        |        |        |       |
|-----------|--------|--------|--------|-------|
|           | $f(x)$ | $f(y)$ | $f(z)$ | $ W $ |
| $W_{i-1}$ | $k$    | $l$    | $0$    | $N$   |
| $W_i$     | $k-m$  | $l-m$  | $m$    | $N-m$ |

Table 1: The change between iterations in word frequency and sequence length in regularized compression. In the new sequence  $W_i$ , each occurrence of the  $x$ - $y$  bigram is replaced with a new (conceptually unseen) word  $z$ . This has an effect of reducing the number of words in the sequence.

### 3 Change in Description Length

The second term of the aforementioned objective is in fact an approximate to the change in description length. This is made obvious by coding up a sequence  $W$  using the Shannon code, with which the description length of  $W$  is equal to  $|W|\tilde{H}(W)$ . Here, the change in description length between sequences  $W_{i-1}$  and  $W_i$  is written as:

$$\Delta L = |W_i|\tilde{H}(W) - |W_{i-1}|\tilde{H}(W_{i-1}). \quad (1)$$

Let us focus on this equation. Suppose that the original sequence  $W_{i-1}$  is  $N$ -word long, the selected word type pair  $x$  and  $y$  each occurs  $k$  and  $l$  times, respectively, and altogether  $x$ - $y$  bigram occurs  $m$  times in  $W_{i-1}$ . In the new sequence  $W_i$ , each of the  $m$  bigrams is replaced with an unseen word  $z = xy$ . These altogether have reduced the sequence length by  $m$ . The end result is that compression moves probability masses from one place to the other, causing a change in description length. See Table 1 for a summary to this exchange.

Now, as we expand Equation (1) and reorganize the remaining, we find that:

$$\begin{aligned} \Delta L = & (N-m)\log(N-m) - N\log N \\ & + k\log k - (k-m)\log(k-m) \\ & + l\log l - (l-m)\log(l-m) \\ & + 0\log 0 - m\log m \end{aligned} \quad (2)$$

Note that each line in Equation (2) is of the form  $x_1\log x_1 - x_2\log x_2$  for some  $x_1, x_2 \geq 0$ . We exploit this pattern and derive a bound for  $\Delta L$  through analysis. Consider  $g(x) = x\log x$ . Since  $g''(x) > 0$  for  $x \geq 0$ , by the Taylor series we have the following relations for any  $x_1, x_2 \geq 0$ :

$$\begin{aligned} g(x_1) - g(x_2) & \leq (x_1 - x_2)g'(x_1), \\ g(x_1) - g(x_2) & \geq (x_1 - x_2)g'(x_2). \end{aligned}$$

Plugging these into Equation (2), we have:

$$m\log\frac{(k-m)(l-m)}{Nm} \leq \Delta L \leq \infty. \quad (3)$$

The lower bound<sup>1</sup> at the left-hand side is a best-case estimate. As our aim is to minimize  $\Delta L$ , we use this quantity to serve as an approximate.

### 4 Proposed Method

Based on this finding, we propose the following two variations (see Figure 1) for the regularized compression framework:

- $G_1$ : Replacing the second term in the original objective with the lower bound in Equation (3). The new objective function is written out as Equation (4).
- $G_2$ : Same as  $G_1$  except that the lower bound is divided by  $f(x, y)$  beforehand. The normalized lower bound approximates the per-word change in description length, as shown in Equation (5). With this variation, the function remains in a scalarized form as the original does.

We use the following procedure to compute description length. Given a word sequence  $W$ , we write out all the induced word types (say,  $M$  types in total) entry by entry as a character sequence, denoted as  $C$ . Then the overall description length is:

$$|W|\tilde{H}(W) + |C|\tilde{H}(C) + \frac{M-1}{2}\log|W|. \quad (6)$$

Three free parameters,  $\alpha$ ,  $\rho$ , and  $n_{ms}$  remain to be estimated. A detailed treatment on parameter estimation is given in the following paragraphs.

**Trade-off  $\alpha$**  This parameter controls the balance between compression rate and vocabulary complexity. Throughout this experiment, we estimated this parameter using MDL-based grid search. Multiple search runs at different granularity levels were employed as necessary.

**Compression rate  $\rho$**  This is the minimum threshold value for compression rate. The compressor algorithm would go on as many iteration as possible until the overall compression rate (i.e.,

<sup>1</sup>Sharp-eyed readers may have noticed the similarity between the lower bound and the negative (empirical) pointwise mutual information. In fact, when  $f(z) > 0$  in  $W_{i-1}$ , it can be shown that  $\lim_{m \rightarrow 0} \Delta L/m$  converges to the empirical pointwise mutual information (proof omitted here).

$$G_1 \equiv f(x, y) \left( \log \frac{(f(x) - f(x, y))(f(y) - f(x, y))}{|W_{i-1}|f(x, y)} - \alpha \right) \quad (4)$$

$$G_2 \equiv -\alpha f(x, y) + \log \frac{(f(x) - f(x, y))(f(y) - f(x, y))}{|W_{i-1}|f(x, y)} \quad (5)$$

Figure 1: The two newly-proposed objective functions.

word/character ratio) is lower than  $\rho$ . Setting this value to 0 forces the compressor to go on until no more can be done. In this paper, we experimented with predetermined  $\rho$  values as well as those learned from MDL-based grid search.

**Minimum support  $n_{ms}$**  We simply followed the suggested setting  $n_{ms} = 3$  (Chen et al., 2012).

## 5 Evaluation

### 5.1 Setup

In the experiment, we tested our methods on Brent’s derivation of the Bernstein-Ratner corpus (Brent and Cartwright, 1996; Bernstein-Ratner, 1987). This dataset is distributed via the CHILDES project (MacWhinney and Snow, 1990) and has been commonly used as a standard benchmark for phonetic segmentation. Our baseline method is the original regularized compressor algorithm (Chen et al., 2012). In our experiment, we considered the following three search settings for finding the model parameters:

- (a) Fix  $\rho$  to 0 and vary  $\alpha$  to find the best value (in the sense of description length);
- (b) Fix  $\alpha$  to the best value found in setting (a) and vary  $\rho$ ;
- (c) Set  $\rho$  to a heuristic value 0.37 (Chen et al., 2012) and vary  $\alpha$ .

Settings (a) and (b) can be seen as running a stochastic grid searcher one round for each parameter<sup>2</sup>. Note that we tested (c) here only to compare with the best baseline setting.

### 5.2 Result

Table 2 summarizes the result for each objective and each search setting. The best  $(\alpha, \rho)$  pair for

<sup>2</sup>A more formal way to estimate both  $\alpha$  and  $\rho$  is to run a stochastic searcher that varies between settings (a) and (b), fixing the best value found in the previous run. Here, for simplicity, we leave this to future work.

| Run       |                  | P           | R           | F           |
|-----------|------------------|-------------|-------------|-------------|
| Baseline  |                  | 76.9        | 81.6        | 79.2        |
| $G_1$ (a) | $\alpha : 0.030$ | 76.4        | 79.9        | 78.1        |
| $G_1$ (b) | $\rho : 0.38$    | 73.4        | 80.2        | 76.8        |
| $G_1$ (c) | $\alpha : 0.010$ | 75.7        | 80.4        | 78.0        |
| $G_2$ (a) | $\alpha : 0.002$ | <b>82.1</b> | 80.0        | 81.0        |
| $G_2$ (b) | $\rho : 0.36$    | 79.1        | 81.7        | 80.4        |
| $G_2$ (c) | $\alpha : 0.004$ | 79.3        | <b>84.2</b> | <b>81.7</b> |

Table 2: The performance result on the Bernstein-Ratner corpus. Segmentation performance is measured using word-level precision (P), recall (R), and F-measure (F).

$G_1$  is (0.03, 0.38) and the best for  $G_2$  is (0.002, 0.36). On one hand, the performance of  $G_1$  is consistently inferior to the baseline across all settings. Although approximation error was one possible cause, we noticed that the compression process was no longer properly regularized, since  $f(x, y)$  and the  $\Delta L$  estimate in the objective are intermingled. In this case, adjusting  $\alpha$  has little effect in balancing compression rate and complexity.

The second objective  $G_2$ , on the other hand, did not suffer as much from the aforementioned lack of regularization. We found that, in all three settings,  $G_2$  outperforms the baseline by 1 to 2 percentage points in F-measure. The best performance result achieved by  $G_2$  in our experiment is 81.7 in word-level F-measure, although this was obtained from search setting (c), using a heuristic  $\rho$  value 0.37. It is interesting to note that  $G_1$  (b) and  $G_2$  (b) also gave very close estimates to this heuristic value. Nevertheless, it remains an open issue whether there is a connection between the optimal  $\rho$  value and the true word/token ratio ( $\approx 0.35$  for Bernstein-Ratner corpus).

The result has led us to conclude that MDL-based grid search is efficient in optimizing segmentation accuracy. Minimization of description length is in general aligned with performance improvement, although under finer granularity MDL-based search may not be as effec-



| Method                                   |                                | P    | R    | F    |
|--|--------------------------------|------|------|------|
| Adaptors grammar, colloc3-syllable       | Johnson and Goldwater (2009)   | 86.1 | 88.4 | 87.2 |
| Regularized compression + MDL, $G_2$ (b) | —                              | 79.1 | 81.7 | 80.4 |
| Regularized compression + MDL            | Chen et al. (2012)             | 76.9 | 81.6 | 79.2 |
| Adaptors grammar, colloc                 | Johnson and Goldwater (2009)   | 78.4 | 75.7 | 77.1 |
| Particle filter, unigram                 | Börschinger and Johnson (2012) | —    | —    | 77.1 |
| Regularized compression + MDL, $G_1$ (b) | —                              | 73.4 | 80.2 | 76.8 |
| Bootstrap voting experts + MDL           | Hewlett and Cohen (2011)       | 79.3 | 73.4 | 76.2 |
| Nested Pitman-Yor process, bigram        | Mochihashi et al. (2009)       | 74.8 | 76.7 | 75.7 |
| Branching entropy + MDL                  | Zhikov et al. (2010)           | 76.3 | 74.5 | 75.4 |
| Particle filter, bigram                  | Börschinger and Johnson (2012) | —    | —    | 74.5 |
| Hierarchical Dirichlet process           | Goldwater et al. (2009)        | 75.2 | 69.6 | 72.3 |

Table 3: The performance chart on the Bernstein-Ratner corpus, in descending order of word-level F-measure. We deliberately reproduced the results for adaptors grammar and regularized compression. The other measurements came directly from the literature.

tive. In our experiment, search setting (b) won out on description length for both objectives, while the best performance was in fact achieved by the others. It would be interesting to confirm this by studying the correlation between description length and word-level F-measure.

In Table 3, we summarize many published results for segmentation methods ever tested on the Bernstein-Ratner corpus. Of the proposed methods, we include only setting (b) since it is more general than the others. From Table 3, we find that the performance of  $G_2$  (b) is competitive to other state-of-the-art hierarchical Bayesian models and MDL methods, though it still lags 7 percentage points behind the best result achieved by adaptors grammar with colloc3-syllable. We also compare adaptors grammar to regularized compressor on average running time, which is shown in Table 4. On our test machine, it took roughly 15 hours for one instance of adaptors grammar with colloc3-syllable to run to the finish. Yet an improved regularized compressor could deliver the result in merely 1.25 second. In other words, even in an  $100 \times 100$  grid search, the regularized compressor algorithm can still finish 4 to 5 times earlier than one single adaptors grammar instance.

## 6 Concluding Remarks

In this paper, we derive a new lower-bound approximate to the objective function used in the regularized compression algorithm. As computing the approximate no longer relies on the change in lexicon entropy, the new compressor algorithm is made more efficient than the original. Besides run-

| Method                             | Time (s) |
|------------------------------------|----------|
| Adaptors grammar, colloc3-syllable | 53826    |
| Adaptors grammar, colloc           | 10498    |
| Regularized compressor             | 1.51     |
| Regularized compressor, $G_1$ (b)  | 0.60     |
| Regularized compressor, $G_2$ (b)  | 1.25     |

Table 4: The average running time in seconds on the Bernstein-Ratner corpus for adaptors grammar (per fold, based on trace output) and regularized compressors, tested on an Intel Xeon 2.5GHz 8-core machine with 8GB RAM.

time efficiency, our experiment result also shows improved performance. Using MDL alone, one proposed method outperforms the original regularized compressor (Chen et al., 2012) in precision by 2 percentage points and in F-measure by 1. Its performance is only second to the state of the art, achieved by adaptors grammar with colloc3-syllable (Johnson and Goldwater, 2009).

A natural extension of this work is to reproduce this result on some other word segmentation benchmarks, specifically those in other Asian languages (Emerson, 2005; Zhikov et al., 2010). Furthermore, it would be interesting to investigate stochastic optimization techniques for regularized compression that simultaneously fit both  $\alpha$  and  $\rho$ . We believe this would be the key to adapt the algorithm to larger datasets.

## Acknowledgments

We thank the anonymous reviewers for their valuable feedback.

## References

- Nan Bernstein-Ratner. 1987. The phonology of parent child speech. *Children's language*, 6:159–174.
- Benjamin Borschinger and Mark Johnson. 2011. A particle filter algorithm for bayesian word segmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia, December.
- Benjamin Börschinger and Mark Johnson. 2012. Using rejuvenation to improve particle filtering for bayesian word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–89, Jeju Island, Korea, July. Association for Computational Linguistics.
- Michael R. Brent and Timothy A. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. In *Cognition*, pages 93–125.
- Ruey-Cheng Chen, Chiung-Min Tsai, and Jieh Hsiang. 2012. A regularized compression method to unsupervised word segmentation. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology, SIG-MORPHON '12*, pages 26–34, Montreal, Canada. Association for Computational Linguistics.
- Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 133. Jeju Island, Korea.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, July.
- Daniel Hewlett and Paul Cohen. 2009. Bootstrap voting experts. In *Proceedings of the 21st international joint conference on Artificial intelligence, IJCAI'09*, pages 1071–1076, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Daniel Hewlett and Paul Cohen. 2011. Fully unsupervised word segmentation with BVE and MDL. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages 540–545, Portland, Oregon. Association for Computational Linguistics.
- Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09*, pages 317–325, Boulder, Colorado. Association for Computational Linguistics.
- Brian MacWhinney and Catherine Snow. 1990. The child language data exchange system: an update. *Journal of child language*, 17(2):457–472, June.
- Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. 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 100–108, Suntec, Singapore. Association for Computational Linguistics.
- Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471, September.
- Kumiko Tanaka-Ishii. 2005. Entropy as an indicator of context boundaries: An experiment using a web search engine. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Kwong, editors, *Natural Language Processing IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, chapter 9, pages 93–105. Springer Berlin / Heidelberg, Berlin, Heidelberg.
- Ian H. Witten, Alistair Moffat, and Timothy C. Bell. 1999. *Managing gigabytes (2nd ed.): compressing and indexing documents and images*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Valentin Zhikov, Hiroya Takamura, and Manabu Okumura. 2010. An efficient algorithm for unsupervised word segmentation with branching entropy and MDL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 832–842, Cambridge, Massachusetts. Association for Computational Linguistics.

# Co-regularizing character-based and word-based models for semi-supervised Chinese word segmentation

Xiaodong Zeng<sup>†</sup> Derek F. Wong<sup>†</sup> Lidia S. Chao<sup>†</sup> Isabel Trancoso<sup>‡</sup>

<sup>†</sup>Department of Computer and Information Science, University of Macau

<sup>‡</sup>INESC-ID / Instituto Superior Técnico, Lisboa, Portugal

nlp2ct.samuel@gmail.com, {derekfw, lidiasc}@umac.mo,  
isabel.trancoso@inesc-id.pt

## Abstract

This paper presents a semi-supervised Chinese word segmentation (CWS) approach that co-regularizes character-based and word-based models. Similarly to multi-view learning, the “segmentation agreements” between the two different types of view are used to overcome the scarcity of the label information on unlabeled data. The proposed approach trains a character-based and word-based model on labeled data, respectively, as the initial models. Then, the two models are constantly updated using unlabeled examples, where the learning objective is maximizing their segmentation agreements. The agreements are regarded as a set of valuable constraints for regularizing the learning of both models on unlabeled data. The segmentation for an input sentence is decoded by using a joint scoring function combining the two induced models. The evaluation on the Chinese tree bank reveals that our model results in better gains over the state-of-the-art semi-supervised models reported in the literature.

## 1 Introduction

Chinese word segmentation (CWS) is a critical and a necessary initial procedure with respect to the majority of high-level Chinese language processing tasks such as syntax parsing, information extraction and machine translation, since Chinese scripts are written in continuous characters without explicit word boundaries. Although supervised CWS models (Xue, 2003; Zhao et al., 2006; Zhang and Clark, 2007; Sun, 2011) proposed in the past years showed some reasonably accurate results, the outstanding problem is that they rely heavily on a large amount of labeled data.

However, the production of segmented Chinese texts is time-consuming and expensive, since hand-labeling individual words and word boundaries is very hard (Jiao et al., 2006). So, one cannot rely only on the manually segmented data to build an everlasting model. This naturally provides motivation for using easily accessible raw texts to enhance supervised CWS models, in semi-supervised approaches. In the past years, however, few semi-supervised CWS models have been proposed. Xu et al. (2008) described a Bayesian semi-supervised model by considering the segmentation as the hidden variable in machine translation. Sun and Xu (2011) enhanced the segmentation results by interpolating the statistics-based features derived from unlabeled data to a CRFs model. Another similar trial via “feature engineering” was conducted by Wang et al. (2011).

The crux of solving semi-supervised learning problem is the learning on unlabeled data. Inspired by multi-view learning that exploits redundant views of the same input data (Ganchev et al., 2008), this paper proposes a semi-supervised CWS model of co-regularizing from two different views (intrinsically two different models), character-based and word-based, on unlabeled data. The motivation comes from that the two types of model exhibit different strengths and they are mutually complementary (Sun, 2010; Wang et al., 2010). The proposed approach begins by training a character-based and word-based model on labeled data respectively, and then both models are regularized from each view by their segmentation agreements, i.e., the identical outputs, of unlabeled data. This paper introduces segmentation agreements as gainful knowledge for guiding the learning on the texts without label information. Moreover, in order to better combine the strengths of the two models, the proposed approach uses a joint scoring function in a log-linear combination form for the decoding in the segmentation phase.

## 2 Segmentation Models

There are two classes of CWS models: character-based and word-based. This section briefly reviews two supervised models in these categories, a character-based CRFs model, and a word-based Perceptrons model, which are used in our approach.

### 2.1 Character-based CRFs Model

Character-based models treat word segmentation as a sequence labeling problem, assigning labels to the characters in a sentence indicating their positions in a word. A 4 tag-set is used in this paper: **B** (beginning), **M** (middle), **E** (end) and **S** (single character). Xue (2003) first proposed the use of CRFs model (Lafferty et al., 2001) in character-based CWS. Let  $x = (x^1 x^2 \dots x^{|x|}) \in \mathcal{X}$  denote a sentence, where each character and  $y = (y^1 y^2 \dots y^{|y|}) \in \mathcal{Y}$  denote a tag sequence,  $y^i \in \mathcal{T}$  being the tag assigned to  $x^i$ . The goal is to achieve a label sequence with the best score in the form,

$$p_{\theta_c}(y|x) = \frac{1}{Z(x; \theta_c)} \exp\{f(x, y) \cdot \theta_c\} \quad (1)$$

where  $Z(x; \theta_c)$  is a partition function that normalizes the exponential form to be a probability distribution, and  $f(x, y)$  are arbitrary feature functions. The aim of CRFs is to estimate the weight parameters  $\theta_c$  that *maximizes* the conditional likelihood of the training data:

$$\hat{\theta}_c = \operatorname{argmax}_{\theta_c} \sum_{i=1}^l \log p_{\theta_c}(y^i|x^i) - \gamma \|\theta_c\|_2^2 \quad (2)$$

where  $\gamma \|\theta_c\|_2^2$  is a regularizer on parameters to limit overfitting on rare features and avoid degeneracy in the case of correlated features. In this paper, this objective function is optimized by stochastic gradient method. For the decoding, the Viterbi algorithm is employed.

### 2.2 Word-based Perceptrons Model

Word-based models read a input sentence from left to right and predict whether the current piece of continuous characters is a word. After one word is identified, the method moves on and searches for a next possible word. Zhang and Clark (2007) first proposed a word-based segmentation model using a discriminative Perceptrons algorithm. Given a sentence  $x$ , let us denote a possible segmented sentence as  $w \in \mathbf{w}$ , and the function that

enumerates a set of segmentation candidates as  $\mathbf{w} = \text{GEN}(x)$  for  $x$ . The objective is to *maximize* the following problem for all sentences:

$$\hat{\theta}_w = \operatorname{argmax}_{\mathbf{w}=\text{GEN}(x)} \sum_{i=1}^{|\mathbf{w}|} \phi(x, w_i) \cdot \theta_w \quad (3)$$

where it maps the segmented sentence  $w$  to a global feature vector  $\phi$  and denotes  $\theta_w$  as its corresponding weight parameters. The parameters  $\theta_w$  can be estimated by using the Perceptrons method (Collins, 2002) or other online learning algorithms, e.g., Passive Aggressive (Crammer et al., 2006). For the decoding, a beam search decoding method (Zhang and Clark, 2007) is used.

### 2.3 Comparison Between Both Models

Character-based and word-based models present different behaviors and each one has its own strengths and weakness. Sun (2010) carried out a thorough survey that includes theoretical and empirical comparisons from four aspects. Here, two critical properties of the two models supporting the co-regularization in this study are highlighted. Character-based models present better prediction ability for new words, since they lay more emphasis on the internal structure of a word and thereby express more nonlinearity. On the other side, it is easier to define the word-level features in word-based models. Hence, these models have a greater representational power and consequently better recognition performance for in-of-vocabulary (IV) words.

## 3 Semi-supervised Learning via Co-regularizing Both Models

As mentioned earlier, the primary challenge of semi-supervised CWS concentrates on the unlabeled data. Obviously, the learning on unlabeled data does not come for “free”. Very often, it is necessary to discover certain gainful information, e.g., label constraints of unlabeled data, that is incorporated to guide the learner toward a desired solution. In our approach, we believe that the segmentation agreements (§ 3.1) from two different views, character-based and word-based models, can be such gainful information. Since each of the models has its own merits, their consensus signify high confidence segmentations. This naturally leads to a new learning objective that maximizes segmentation agreements between two models on unlabeled data.

This study proposes a co-regularized CWS model based on character-based and word-based models, built on a small amount of segmented sentences (labeled data) and a large amount of raw sentences (unlabeled data). The model induction process is described in Algorithm 1: given labeled dataset  $D_l$  and unlabeled dataset  $D_u$ , the first two steps are training a CRFs (character-based) and Perceptrons (word-based) model on the labeled data  $D_l$ , respectively. Then, the parameters of both models are continually updated using unlabeled examples in a learning cycle. At each iteration, the raw sentences in  $D_u$  are segmented by current character-based model  $\theta_c$  and word-based model  $\theta_w$ . Meanwhile, all the segmentation agreements  $\mathcal{A}$  are collected (§ 3.1). Afterwards, the agreements  $\mathcal{A}$  are used as a set of constraints to bias the learning of CRFs (§ 3.2) and Perceptron (§ 3.3) on the unlabeled data. The convergence criterion is the occurrence of a reduction of segmentation agreements or reaching the maximum number of learning iterations. In the final segmentation phase, given a raw sentence, the decoding requires both induced models (§ 3.4) in measuring a segmentation score.

---

**Algorithm 1** Co-regularized CWS model induction

---

**Require:**  $n$  labeled sentences  $D_l$ ;  $m$  unlabeled sentences  $D_u$   
**Ensure:**  $\theta_c$  and  $\theta_w$   
1:  $\theta_c^0 \leftarrow \text{crf.train}(D_l)$   
2:  $\theta_w^0 \leftarrow \text{perceptron.train}(D_l)$   
3: for  $t = 1 \dots T_{max}$  do  
4:  $\mathcal{A}^t \leftarrow \text{agree}(D_u, \theta_c^{t-1}, \theta_w^{t-1})$   
5:  $\theta_c^t \leftarrow \text{crf.train.constraints}(D_u, \mathcal{A}^t, \theta_c^{t-1})$   
6:  $\theta_w^t \leftarrow \text{perceptron.train.constraints}(D_u, \mathcal{A}^t, \theta_w^{t-1})$   
7: end for

---

### 3.1 Agreements Between Two Models

Given a raw sentence, e.g., “我正在北京看奥运会开幕式。(I am watching the opening ceremony of the Olympics in Beijing.)”, the two segmentations shown in Figure 1 are the predictions from a character-based and word-based model. The segmentation agreements between the two models correspond to the identical words. In this example, the five words, i.e. “我 (I)”, “北京 (Beijing)”, “看 (watch)”, “开幕式 (opening ceremony)” and “。(.)”, are the agreements.

### 3.2 CRFs with Constraints

For the character-based model, this paper follows (Täckström et al., 2013) to incorporate the segmentation agreements into CRFs. The main

idea is to constrain the size of the tag sequence lattice according to the agreements for achieving simplified learning. Figure 2 demonstrates an example of the constrained lattice, where the bold node represents that a definitive tag derived from the agreements is assigned to the current character, e.g., “我 (I)” has only one possible tag “S” because both models segmented it to a word with a single character. Here, if the lattice of all admissible tag sequences for the sentence  $x$  is denoted as  $\mathcal{Y}(x)$ , the constrained lattice can be defined by  $\hat{\mathcal{Y}}(x, \tilde{y})$ , where  $\tilde{y}$  refers to tags inferred from the agreements. Thus, the objective function on unlabeled data is modeled as:

$$\hat{\theta}'_c = \underset{\theta_c}{\operatorname{argmax}} \sum_{i=1}^m \log p_{\theta_c}(\hat{\mathcal{Y}}(x^i, \tilde{y}^i) | x^i) - \gamma \|\theta_c\|_2^2 \quad (4)$$

It is a marginal conditional probability given by the total probability of all tag sequences consistent with the constrained lattice  $\hat{\mathcal{Y}}(x, \tilde{y})$ . This objective can be optimized by using LBFGS-B (Zhu et al., 1997), a generic quasi-Newton gradient-based optimizer.

Character-based: 我 正在 北京 看 奥运会 开幕式。  
Word-based: 我 正在 北京 看 奥运会 开幕式。

Figure 1: The segmentations given by character-based and word-based model, where the words in “□” refer to the segmentation agreements.

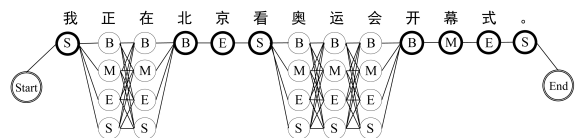


Figure 2: The constrained lattice representation for a given sentence, “我正在北京看奥运会开幕式。”.

### 3.3 Perceptrons with Constraints

For the word-based model, this study incorporates segmentation agreements by a modified parameter update criterion in Perceptrons online training, as shown in Algorithm 2. Because there are no “gold segmentations” for unlabeled sentences, the output sentence predicted by the current model is compared with the agreements instead of the “answers” in the supervised case. At each parameter

update iteration  $k$ , each raw sentence  $x_u$  is decoded with the current model into a segmentation  $z_u$ . If the words in output  $z_u$  do not match the agreements  $\mathcal{A}(x_u)$  of the current sentence  $x_u$ , the parameters are updated by adding the global feature vector of the current training example with the agreements and subtracting the global feature vector of the decoder output, as described in lines 3 and 4 of Algorithm 2.

---

**Algorithm 2** Parameter update in word-based model

---

```

1: for  $k = 1 \dots K, u = 1 \dots m$  do
2:   calculate  $z_u = \operatorname{argmax}_{\mathbf{w}=\text{GEN}(x)} \sum_{i=1}^{|w|} \phi(x_u, w_i) \cdot \theta_w^{k-1}$ 
3:   if  $z_u \neq \mathcal{A}(x_u)$ 
4:      $\theta_w^k = \theta_w^{k-1} + \phi(\mathcal{A}(x_u)) - \phi(z_u)$ 
5: end for

```

---

### 3.4 The Joint Score Function for Decoding

There are two co-regularized models as results of the previous induction steps. An intuitive idea is that both induced models are combined to conduct the segmentation, for the sake of integrating their strengths. This paper employs a log-linear interpolation combination (Bishop, 2006) to formulate a joint scoring function based on character-based and word-based models in the decoding:

$$\text{Score}(w) = \alpha \cdot \log(p_{\theta_c}(y|x)) + (1 - \alpha) \cdot \log(\phi(x, w) \cdot \theta_w) \quad (5)$$

where the two terms of the logarithm are the scores of character-based and word-based models, respectively, for a given segmentation  $w$ . This composite function uses a parameter  $\alpha$  to weight the contributions of the two models. The  $\alpha$  value is tuned using the development data.

## 4 Experiment

### 4.1 Setting

The experimental data is taken from the Chinese tree bank (CTB). In order to make a fair comparison with the state-of-the-art results, the versions of CTB-5, CTB-6, and CTB-7 are used for the evaluation. The training, development and testing sets are defined according to the previous works. For CTB-5, the data split from (Jiang et al., 2008) is employed. For CTB-6, the same data split as recommended in the CTB-6 official document is used. For CTB-7, the datasets are formed according to the way in (Wang et al., 2011). The corresponding statistic information on these data splits is reported in Table 1. The unlabeled data in

our experiments is from the XIN\_CMN portion of Chinese Gigaword 2.0. The articles published in 1991-1993 and 1999-2004 are used as unlabeled data, with 204 million words.

The feature templates in (Zhao et al., 2006) and (Zhang and Clark, 2007) are used in training the CRFs model and Perceptrons model, respectively. The experimental platform is implemented based on two popular toolkits: CRF++ (Kudo, 2005) and Zpar (Zhang and Clark, 2011).

| Data  | #Sent-train | #Sent-dev | #Sent-test | OOV-dev | OOV-test |
|-------|-------------|-----------|------------|---------|----------|
| CTB-5 | 18,089      | 350       | 348        | 0.0811  | 0.0347   |
| CTB-6 | 23,420      | 2,079     | 2,796      | 0.0545  | 0.0557   |
| CTB-7 | 31,131      | 10,136    | 10,180     | 0.0549  | 0.0521   |

Table 1: Statistics of CTB-5, CTB-6 and CTB-7 data.

### 4.2 Main Results

The development sets are mainly used to tune the values of the weight factor  $\alpha$  in Equation 5. We evaluated the performance (F-score) of our model on the three development sets by using different  $\alpha$  values, where  $\alpha$  is progressively increased in steps of 0.1 ( $0 < \alpha < 1.0$ ). The best performed settings of  $\alpha$  for CTB-5, CTB-6 and CTB-7 on development data are 0.7, 0.6 and 0.6, respectively. With the chosen parameters, the test data is used to measure the final performance.

Table 2 shows the F-score results of word segmentation on CTB-5, CTB-6 and CTB-7 testing sets. The line of “ours” reports the performance of our semi-supervised model with the tuned parameters. We first compare it with the supervised “baseline” method which joints character-based and word-based model trained only on the training set<sup>1</sup>. It can be observed that our semi-supervised model is able to benefit from unlabeled data and greatly improves the results over the supervised baseline. We also compare our model with two state-of-the-art semi-supervised methods of Wang ’11 (Wang et al., 2011) and Sun ’11 (Sun and Xu, 2011). The performance scores of Wang ’11 are directly taken from their paper, while the results of Sun ’11 are obtained, using the program provided by the author, on the same experimental data. The

---

<sup>1</sup>The “baseline” uses a different training configuration so that the  $\alpha$  values in the decoding are also need to be tuned on the development sets. The tuned  $\alpha$  values are {0.6, 0.6, 0.5} for CTB-5, CTB-6 and CTB-7.

bold scores indicate that our model does achieve significant gains over these two semi-supervised models. This outcome can further reveal that using the agreements from these two views to regularize the learning can effectively guide the model toward a better solution. The third comparison candidate is Hatori '12 (Hatori et al., 2012) which reported the best performance in the literature on these three testing sets. It is a supervised joint model of word segmentation, POS tagging and dependency parsing. Impressively, our model still outperforms Hatori '12 on all three datasets. Although there is only a 0.01 increase on CTB-5, it can be seen as a significant improvement when considering Hatori '12 employs much richer training resources, i.e., sentences tagged with syntactic information.

| Method     | CTB-5        | CTB-6        | CTB-7        |
|------------|--------------|--------------|--------------|
| Ours       | <b>98.27</b> | <b>96.33</b> | <b>96.72</b> |
| Baseline   | 97.58        | 94.71        | 94.87        |
| Wang '11   | 98.11        | 95.79        | 95.65        |
| Sun '11    | 98.04        | 95.44        | 95.34        |
| Hatori '12 | 98.26        | 96.18        | 96.07        |

Table 2: F-score (%) results of five CWS models on CTB-5, CTB-6 and CTB-7.

## 5 Conclusion

This paper proposed an alternative semi-supervised CWS model that co-regularizes a character- and word-based model by using their segmentation agreements on unlabeled data. We perform the agreements as valuable knowledge for the regularization. The experiment results reveal that this learning mechanism results in a positive effect to the segmentation performance.

## Acknowledgments

The authors are grateful to the Science and Technology Development Fund of Macau and the Research Committee of the University of Macau for the funding support for our research, under the reference No. 017/2009/A and MYRG076(Y1-L2)-FST13-WF. The authors also wish to thank the anonymous reviewers for many helpful comments.

## References

Christopher M. Bishop. 2006. *Pattern recognition and machine learning*.

- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1-8, Philadelphia, USA.
- 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.
- Kuzman Ganchev, Joao Graca, John Blitzer, and Ben Taskar. 2008. Multi-View Learning over Structured and Non-Identical Outputs. In *Proceedings of CUAJ*, pages 204-211, Helsinki, Finland.
- Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. In *Proceedings of ACL*, pages 1045-1053, Jeju, Republic of Korea.
- Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Liu. 2008. A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of ACL*, pages 897-904, Columbus, Ohio.
- Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging - A Case Study. In *Proceedings of ACL and the 4th IJCNLP of the AFNLP*, pages 522-530, Suntec, Singapore.
- Feng Jiao, Shaojun Wang and Chi-Hoon Lee. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of ACL and the 4th IJCNLP of the AFNLP*, pages 209-216, Stroudsburg, PA, USA.
- Taku Kudo. 2005. CRF++: Yet another CRF toolkit. Software available at <http://crfpp.sourceforge.net>.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Field: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, pages 282-289, Williams College, USA.
- Weiwei Sun. 2001. Word-based and character-based word segmentation models: comparison and combination. In *Proceedings of COLING*, pages 1211-1219, Beijing, China.
- 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, Portland, Oregon.
- Weiwei Sun and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of EMNLP*, pages 970-979, Scotland, UK.
- Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging. In *Transactions of the Association for Computational Linguistics*, 1:1-12.

- Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A Character-Based Joint Model for Chinese Word Segmentation. In *Proceedings of COLING*, pages 1173-1181, Beijing, China.
- 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, Hyderabad, India.
- Jia Xu, Jianfeng Gao, Kristina Toutanova and Hermann Ney. 2008. Bayesian semi-supervised chinese word segmentation for statistical machine translation. In *Proceedings of COLING*, pages 1017-1024, Manchester, UK.
- Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29-48.
- Yue Zhang and Stephen Clark. 2007. Chinese segmentation using a word-based perceptron algorithm. In *Proceedings of ACL*, pages 840-847, Prague, Czech Republic.
- 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, Massachusetts, USA.
- Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105-151.
- 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*, pages 87-94, Wuhan, China.
- Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 2006. L-BFGS-B: Fortran subroutines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23:550-560.



# Improving Chinese Word Segmentation on Micro-blog Using Rich Punctuations

Longkai Zhang Li Li Zhengyan He Houfeng Wang\* Ni Sun

Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China  
zhlongk@qq.com, li.l@pku.edu.cn, hezhengyan.hit@gmail.com,  
wanghf@pku.edu.cn, sunny.forwork@gmail.com

## Abstract

Micro-blog is a new kind of medium which is short and informal. While no segmented corpus of micro-blogs is available to train Chinese word segmentation model, existing Chinese word segmentation tools cannot perform equally well as in ordinary news texts. In this paper we present an effective yet simple approach to Chinese word segmentation of micro-blog. In our approach, we incorporate punctuation information of unlabeled micro-blog data by introducing characters behind or ahead of punctuations, for they indicate the beginning or end of words. Meanwhile a self-training framework to incorporate confident instances is also used, which prove to be helpful. Experiments on micro-blog data show that our approach improves performance, especially in OOV-recall.

## 1 INTRODUCTION

Micro-blog (also known as tweets in English) is a new kind of broadcast medium in the form of blogging. A micro-blog differs from a traditional blog in that it is typically smaller in size. Furthermore, texts in micro-blogs tend to be informal and new words occur more frequently. These new features of micro-blogs make the Chinese Word Segmentation (CWS) models trained on the source domain, such as news corpus, fail to perform equally well when transferred to texts from micro-blogs. For example, the most widely used Chinese segmenter "ICTCLAS" yields 0.95 f-score in news corpus, only gets 0.82 f-score on micro-blog data. The poor segmentation results will hurt subsequent analysis on micro-blog text.

Manually labeling the texts of micro-blog is time consuming. Luckily, punctuations provide useful information because they are used as indicators of the end of previous sentence and the beginning of the next one, which also indicate the start and the end of a word. These "natural boundaries" appear so frequently in micro-blog texts that we can easily make good use of them. TABLE 1 shows some statistics of the news corpus vs. the micro-blogs. Besides, English letters and digits are also more than those in news corpus. They all are natural delimiters of Chinese characters and we treat them just the same as punctuations.

We propose a method to enlarge the training corpus by using punctuation information. We build a semi-supervised learning (SSL) framework which can iteratively incorporate newly labeled instances from unlabeled micro-blog data during the training process. We test our method on micro-blog texts and experiments show good results.

This paper is organized as follows. In section 1 we introduce the problem. Section 2 gives detailed description of our approach. We show the experiment and analyze the results in section 3. Section 4 gives the related works and in section 5 we conclude the whole work.

## 2 Our method

### 2.1 Punctuations

Chinese word segmentation problem might be treated as a character labeling problem which gives each character a label indicating its position in one word. To be simple, one can use label 'B' to indicate a character is the beginning of a word, and use 'N' to indicate a character is not the beginning of a word. We also use the 2-tag in our work. Other tag sets like the 'BIES' tag set are not suitable because the punctuation information cannot decide whether a character after punctuation should be labeled as 'B' or 'S' (word with Single

\*Corresponding author

|            | Chinese | English | Number | Punctuation |
|------------|---------|---------|--------|-------------|
| News       | 85.7%   | 0.6%    | 0.7%   | 13.0%       |
| micro-blog | 66.3%   | 11.8%   | 2.6%   | 19.3%       |

Table 1: Percentage of Chinese, English, number, punctuation in the news corpus vs. the micro-blogs.

character).

Punctuations can serve as implicit labels for the characters before and after them. The character right after punctuations must be the first character of a word, meanwhile the character right before punctuations must be the last character of a word. An example is given in TABLE 2.

## 2.2 Algorithm

Our algorithm “ADD-N” is shown in TABLE 3. The initially selected character instances are those right after punctuations. By definition they are all labeled with ‘B’. In this case, the number of training instances with label ‘B’ is increased while the number with label ‘N’ remains unchanged. Because of this, the model trained on this unbalanced corpus tends to be biased. This problem can become even worse when there is inexhaustible supply of texts from the target domain. We assume that labeled corpus of the source domain can be treated as a balanced reflection of different labels. Therefore we choose to estimate the balanced point by counting characters labeling ‘B’ and ‘N’ and calculate the ratio which we denote as  $\eta$ . We assume the enlarged corpus is also balanced if and only if the ratio of ‘B’ to ‘N’ is just the same to  $\eta$  of the source domain.

Our algorithm uses data from source domain to make the labels balanced. When enlarging corpus using characters behind punctuations from texts in target domain, only characters labeling ‘B’ are added. We randomly reuse some characters labeling ‘N’ from labeled data until ratio  $\eta$  is reached. We do not use characters ahead of punctuations, because the single-character words ahead of punctuations take the label of ‘B’ instead of ‘N’. In summary our algorithm tackles the problem by duplicating labeled data in source domain. We denote our algorithm as “ADD-N”.

We also use baseline feature templates include the features described in previous works (Sun and Xu, 2011; Sun et al., 2012). Our algorithm is not necessarily limited to a specific tagger. For simplicity and reliability, we use a simple Maximum-Entropy tagger.

## 3 Experiment

### 3.1 Data set

We evaluate our method using the data from weibo.com, which is the biggest micro-blog service in China. We use the API provided by weibo.com<sup>1</sup> to crawl 500,000 micro-blog texts of weibo.com, which contains 24,243,772 characters. To keep the experiment tractable, we first randomly choose 50,000 of all the texts as unlabeled data, which contain 2,420,037 characters. We manually segment 2038 randomly selected micro-blogs. We follow the segmentation standard as the PKU corpus.

In micro-blog texts, the user names and URLs have fixed format. User names start with ‘@’, followed by Chinese characters, English letters, numbers and ‘\_’, and terminated when meeting punctuations or blanks. URLs also match fixed patterns, which are shortened using “http://t.cn/” plus six random English letters or numbers. Thus user names and URLs can be pre-processed separately. We follow this principle in following experiments.

We use the benchmark datasets provided by the second International Chinese Word Segmentation Bakeoff<sup>2</sup> as the labeled data. We choose the PKU data in our experiment because our baseline methods use the same segmentation standard.

We compare our method with three baseline methods. The first two are both famous Chinese word segmentation tools: ICTCLAS<sup>3</sup> and Stanford Chinese word segmenter<sup>4</sup>, which are widely used in NLP related to word segmentation. Stanford Chinese word segmenter is a CRF-based segmentation tool and its segmentation standard is chosen as the PKU standard, which is the same to ours. ICTCLAS, on the other hand, is a HMM-based Chinese word segmenter. Another baseline is Li and Sun (2009), which also uses punctuation in their semi-supervised framework. F-score

<sup>1</sup><http://open.weibo.com/wiki>

<sup>2</sup><http://www.sighan.org/bakeoff2005/>

<sup>3</sup><http://ictclas.org/>

<sup>4</sup><http://nlp.stanford.edu/projects/chinese-nlp.shtml\#cws>

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 评 | 论 | 是 | 风 | 格 | , | 评 | 论 | 是 | 能 | 力 | 。 |
| B | - | - | - | - | - | B | - | - | - | - | - |
| B | N | B | B | N | B | B | N | B | B | N | B |

Table 2: The first line represents the original text. The second line indicates whether each character is the Beginning of sentence. The third line is the tag sequence using "BN" tag set.

| ADD-N algorithm  |
|--|
| <b>Input:</b> labeled data $\{(x_i, y_i)_{i=1}^l\}$ , unlabeled data $\{x_j\}_{j=l+1}^{l+u}$ .<br>1. Initially, let $L = \{(x_i, y_i)_{i=1}^l\}$ and $U = \{x_j\}_{j=l+1}^{l+u}$ .<br>2. Label instances behind punctuations in $U$ as 'B' and add them into $L$ .<br>3. Calculate 'B', 'N' ratio $\eta$ in labeled data.<br>4. Randomly duplicate characters whose labels are 'N' in $L$ to make 'B'/'N' = $\eta$<br>5. Repeat:<br>5.1 Train a classifier $f$ from $L$ using supervised learning.<br>5.2 Apply $f$ to tag the unlabeled instances in $U$ .<br>5.3 Add confident instances from $U$ to $L$ . |

Table 3: ADD-N algorithm.

is used as the accuracy measure. The recall of out-of-vocabulary is also taken into consideration, which measures the ability of the model to correctly segment out of vocabulary words.

### 3.2 Main results

| Method     | P     | R     | F     | OOV-R |
|------------|-------|-------|-------|-------|
| Stanford   | 0.861 | 0.853 | 0.857 | 0.639 |
| ICTCLAS    | 0.812 | 0.861 | 0.836 | 0.602 |
| Li-Sun     | 0.707 | 0.820 | 0.760 | 0.734 |
| Maxent     | 0.868 | 0.844 | 0.856 | 0.760 |
| No-punc    | 0.865 | 0.829 | 0.846 | 0.760 |
| No-balance | 0.869 | 0.877 | 0.873 | 0.757 |
| Our method | 0.875 | 0.875 | 0.875 | 0.773 |

Table 4: Segmentation performance with different methods on the development data.

TABLE 4 summarizes the segmentation results. In TABLE 4, Li-Sun is the method in Li and Sun (2009). Maxent only uses the PKU data for training, with neither punctuation information nor self-training framework incorporated. The next 4 methods all require a 100 iteration of self-training. No-punc is the method that only uses self-training while no punctuation information is added. No-balance is similar to ADD N. The only difference between No-balance and ADD-N is that the former does not balance label 'B' and label 'N'.

The comparison of Maxent and No-punctuation

shows that naively adding confident unlabeled instances does not guarantee to improve performance. The writing style and word formation of the source domain is different from target domain. When segmenting texts of the target domain using models trained on source domain, the performance will be hurt with more false segmented instances added into the training set.

The comparison of Maxent, No-balance and ADD-N shows that considering punctuation as well as self-training does improve performance. Both the f-score and OOV-recall increase. By comparing No-balance and ADD-N alone we can find that we achieve relatively high f-score if we ignore tag balance issue, while slightly hurt the OOV-Recall. However, considering it will improve OOV-Recall by about +1.6% and the f-score +0.2%.

We also experimented on different size of unlabeled data to evaluate the performance when adding unlabeled target domain data. TABLE 5 shows different f-scores and OOV-Recalls on different unlabeled data set.

We note that when the number of texts changes from 0 to 50,000, the f-score and OOV both are improved. However, when unlabeled data changes to 200,000, the performance is a bit decreased, while still better than not using unlabeled data. This result comes from the fact that the method 'ADD-N' only uses characters behind punctua-

| Size   | P     | R     | F     | OOV-R |
|--------|-------|-------|-------|-------|
| 0      | 0.864 | 0.846 | 0.855 | 0.754 |
| 10000  | 0.872 | 0.869 | 0.871 | 0.765 |
| 50000  | 0.875 | 0.875 | 0.875 | 0.773 |
| 100000 | 0.874 | 0.879 | 0.876 | 0.772 |
| 200000 | 0.865 | 0.865 | 0.865 | 0.759 |

Table 5: Segmentation performance with different size of unlabeled data

tions from target domain. Taking more texts into consideration means selecting more characters labeling 'N' from source domain to simulate those in target domain. If too many 'N's are introduced, the training data will be biased against the true distribution of target domain.

### 3.3 Characters ahead of punctuations

In the "BN" tagging method mentioned above, we incorporate characters after punctuations from texts in micro-blog to enlarge training set. We also try an opposite approach, "EN" tag, which uses 'E' to represent "End of word", and 'N' to represent "Not the end of word". In this contrasting method, we only use characters just ahead of punctuations. We find that the two methods show similar results. Experiment results with ADD-N are shown in TABLE 6.

| Unlabeled<br>Data size | "BN" tag |       | "EN" tag |       |
|------------------------|----------|-------|----------|-------|
|                        | F        | OOV-R | F        | OOV-R |
| 50000                  | 0.875    | 0.773 | 0.870    | 0.763 |

Table 6: Comparison of BN and EN.

## 4 Related Work

Recent studies show that character sequence labeling is an effective formulation of Chinese word segmentation (Low et al., 2005; Zhao et al., 2006a,b; Chen et al., 2006; Xue, 2003). These supervised methods show good results, however, are unable to incorporate information from new domain, where OOV problem is a big challenge for the research community. On the other hand unsupervised word segmentation Peng and Schuurmans (2001); Goldwater et al. (2006); Jin and Tanaka-Ishii (2006); Feng et al. (2004); Maosong et al. (1998) takes advantage of the huge amount of raw text to solve Chinese word segmentation problems. However, they usually are less accurate and more complicated than supervised ones.

Meanwhile semi-supervised methods have been applied into NLP applications. Bickel et al. (2007) learns a scaling factor from data of source domain and use the distribution to resemble target domain distribution. Wu et al. (2009) uses a Domain adaptive bootstrapping (DAB) framework, which shows good results on Named Entity Recognition. Similar semi-supervised applications include Shen et al. (2004); Daumé III and Marcu (2006); Jiang and Zhai (2007); Weinberger et al. (2006). Besides, Sun and Xu (2011) uses a sequence labeling framework, while unsupervised statistics are used as discrete features in their model, which prove to be effective in Chinese word segmentation.

There are previous works using punctuations as implicit annotations. Riley (1989) uses it in sentence boundary detection. Li and Sun (2009) proposed a compromising solution to by using a classifier to select the most confident characters. We do not follow this approach because the initial errors will dramatically harm the performance. Instead, we only add the characters after punctuations which are sure to be the beginning of words (which means labeling 'B') into our training set. Sun and Xu (2011) uses punctuation information as discrete feature in a sequence labeling framework, which shows improvement compared to the pure sequence labeling approach. Our method is different from theirs. We use characters after punctuations directly.

## 5 Conclusion

In this paper we have presented an effective yet simple approach to Chinese word segmentation on micro-blog texts. In our approach, punctuation information of unlabeled micro-blog data is used, as well as a self-training framework to incorporate confident instances. Experiments show that our approach improves performance, especially in OOV-recall. Both the punctuation information and the self-training phase contribute to this improvement.

## Acknowledgments

This research was partly supported by National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), National Natural Science Foundation of China (No.91024009) and Major National Social Science Fund of China(No. 12&ZD227).

## References

- Bickel, S., Brückner, M., and Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM.
- Chen, W., Zhang, Y., and Isahara, H. (2006). Chinese named entity recognition with conditional random fields. In *5th SIGHAN Workshop on Chinese Language Processing, Australia*.
- Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126.
- Feng, H., Chen, K., Deng, X., and Zheng, W. (2004). Accessor variety criteria for chinese word extraction. *Computational Linguistics*, 30(1):75–93.
- Goldwater, S., Griffiths, T., and Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 673–680. Association for Computational Linguistics.
- Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 264.
- Jin, Z. and Tanaka-Ishii, K. (2006). Unsupervised segmentation of chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 428–435. Association for Computational Linguistics.
- Li, Z. and Sun, M. (2009). Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512.
- Low, J., Ng, H., and Guo, W. (2005). A maximum entropy approach to chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 164. Jeju Island, Korea.
- Maosong, S., Dayang, S., and Tsou, B. (1998). Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1265–1271. Association for Computational Linguistics.
- Pan, S. and Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.
- Peng, F. and Schuurmans, D. (2001). Self-supervised chinese word segmentation. *Advances in Intelligent Data Analysis*, pages 238–247.
- Riley, M. (1989). Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics.
- Shen, D., Zhang, J., Su, J., Zhou, G., and Tan, C. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics.
- Sun, W. and Xu, J. (2011). Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 970–979. Association for Computational Linguistics.
- Sun, X., Wang, H., and Li, W. (2012). Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea. Association for Computational Linguistics.
- Weinberger, K., Blitzer, J., and Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In *In NIPS*. Citeseer.
- Wu, D., Lee, W., Ye, N., and Chieu, H. (2009). Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1523–1532. Association for Computational Linguistics.
- Xue, N. (2003). Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.
- Zhao, H., Huang, C., and Li, M. (2006a). An improved chinese word segmentation system with

conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, volume 117. Sydney: July.

Zhao, H., Huang, C., Li, M., and Lu, B. (2006b). Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of PACLIC*, volume 20, pages 87–94.

# Accurate Word Segmentation using Transliteration and Language Model Projection

Masato Hagiwara

Satoshi Sekine

Rakuten Institute of Technology, New York

215 Park Avenue South, New York, NY

{masato.hagiwara, satoshi.b.sekine}@mail.rakuten.com

## Abstract

Transliterated compound nouns not separated by whitespaces pose difficulty on word segmentation (WS). *Offline* approaches have been proposed to split them using word statistics, but they rely on static lexicon, limiting their use. We propose an *online* approach, integrating source LM, and/or, back-transliteration and English LM. The experiments on Japanese and Chinese WS have shown that the proposed models achieve significant improvement over state-of-the-art, reducing 16% errors in Japanese.

## 1 Introduction

Accurate word segmentation (WS) is the key components in successful language processing. The problem is pronounced in languages such as Japanese and Chinese, where words are not separated by whitespaces. In particular, compound nouns pose difficulties to WS since they are productive, and often consist of unknown words.

In Japanese, transliterated foreign compound words written in Katakana are extremely difficult to split up into components without proper lexical knowledge. For example, when splitting a compound noun ブラキシシュレット *burakisshureddo*, a traditional word segmenter can easily segment this as ブラキシ/シュレット “\*blacki shred” since シュレット *shureddo* “shred” is a known, frequent word. It is only the knowledge that ブラキシ *buraki* (“\*blacki”) is not a valid word which prevents this. Knowing that the back-transliterated unigram “blacki” and bigram “blacki shred” are unlikely in English can promote the correct WS, ブラキシシュ/レット “blackish red”. In Chinese, the problem can be more severe since

the language does not have a separate script to represent transliterated words.

Kaji and Kitsuregawa (2011) tackled Katakana compound splitting using back-transliteration and paraphrasing. Their approach falls into an *offline* approach, which focuses on creating dictionaries by extracting new words from large corpora separately before WS. However, offline approaches have limitation unless the lexicon is constantly updated. Moreover, they only deal with Katakana, but their method is not directly applicable to Chinese since the language lacks a separate script for transliterated words.

Instead, we adopt an *online* approach, which deals with unknown words simultaneously as the model analyzes the input. Our approach is based on semi-Markov discriminative structure prediction, and it incorporates English back-transliteration and English language models (LMs) into WS in a seamless way. We refer to this process of transliterating unknown words into another language and using the target LM as *LM projection*. Since the model employs a general transliteration model and a general English LM, it achieves robust WS for unknown words. To the best of our knowledge, this paper is the first to use transliteration and projected LMs in an online, seamlessly integrated fashion for WS.

To show the effectiveness of our approach, we test our models on a Japanese balanced corpus and an electronic commerce domain corpus, and a balanced Chinese corpus. The results show that we achieved a significant improvement in WS accuracy in both languages.

## 2 Related Work

In Japanese WS, unknown words are usually dealt with in an online manner with the *unknown word model*, which uses heuristics

depending on character types (Kudo et al., 2004). Nagata (1999) proposed a Japanese unknown word model which considers PoS (part of speech), word length model and orthography. Uchimoto et al. (2001) proposed a maximum entropy morphological analyzer robust to unknown words. In Chinese, Peng et al. (2004) used CRF confidence to detect new words.

For offline approaches, Mori and Nagao (1996) extracted unknown word and estimated their PoS from a corpus through distributional analysis. Asahara and Matsumoto (2004) built a character-based chunking model using SVM for Japanese unknown word detection.

Kaji and Kitsuregawa (2011)’s approach is the closest to ours. They built a model to split Katakana compounds using back-transliteration and paraphrasing mined from large corpora. Nakazawa et al. (2005) is a similar approach, using a Ja-En dictionary to translate compound components and check their occurrence in an English corpus. Similar approaches are proposed for other languages, such as German (Koehn and Knight, 2003) and Urdu-Hindi (Lehal, 2010). Correct splitting of compound nouns has a positive effect on MT (Koehn and Knight, 2003) and IR (Braschler and Ripplinger, 2004).

A similar problem can be seen in Korean, German etc. where compounds may not be explicitly split by whitespaces. Koehn and Knight (2003) tackled the splitting problem in German, by using word statistics in a monolingual corpus. They also used the information whether translations of compound parts appear in a German-English bilingual corpus. Lehal (2010) used Urdu-Devnagri transliteration and a Hindi corpus for handling the space omission problem in Urdu compound words.

### 3 Word Segmentation Model

Our baseline model is a semi-Markov structure prediction model which estimates WS and the PoS sequence simultaneously (Kudo et al., 2004; Zhang and Clark, 2008). This model finds the best output  $\mathbf{y}^*$  from the input sentence string  $x$  as:  $\mathbf{y}^* = \arg \max_{\mathbf{y} \in Y(x)} \mathbf{w} \cdot \phi(\mathbf{y})$ . Here,  $Y(x)$  denotes all the possible sequences of words derived from  $x$ . The best analysis is determined by the feature function  $\phi(\mathbf{y})$  the

| ID  | Feature                   | ID  | Feature  |
|-----|---------------------------|-----|--|
| 1   | $w_i$                     | 13  | $w_{i-1}w_i$   |
| 2   | $t_i^1$                   | 14  | $t_{i-1}^1t_i^1$   |
| 3*  | $t_i^1t_i^2$              | 15* | $t_{i-1}^1t_{i-1}^2t_i^1t_i^2$                             |
| 4*  | $t_i^1t_i^2t_i^3$         | 16* | $t_{i-1}^1t_{i-1}^2t_{i-1}^3t_i^1t_i^2t_i^3$               |
| 5*  | $t_i^1t_i^2t_i^5t_i^6$    | 17* | $t_{i-1}^1t_{i-1}^2t_{i-1}^5t_{i-1}^6t_i^1t_i^2t_i^5t_i^6$ |
| 6*  | $t_i^1t_i^2t_i^6$         | 18* | $t_{i-1}^1t_{i-1}^2t_{i-1}^6t_i^1t_i^2t_i^6$               |
| 7   | $w_it_i^1$                | 19  | $\phi_1^{LMS}(w_i)$  |
| 8*  | $w_it_i^1t_i^2$           | 20  | $\phi_2^{LMS}(w_{i-1}, w_i)$                               |
| 9*  | $w_it_i^1t_i^2t_i^3$      | 21  | $\phi_1^{LMP}(w_i)$  |
| 10* | $w_it_i^1t_i^2t_i^5t_i^6$ | 22  | $\phi_2^{LMP}(w_{i-1}, w_i)$                               |
| 11* | $w_it_i^1t_i^2t_i^6$      |     |  |
| 12  | $c(w_i)l(w_i)$            |     |  |

Table 1: Features for WS & PoS tagging

weight vector  $\mathbf{w}$ . WS is conducted by standard Viterbi search based on lattice, which is illustrated in Figure 1. We limit the features to word unigram and bigram features, i.e.,  $\phi(\mathbf{y}) = \sum_i [\phi_1(w_i) + \phi_2(w_{i-1}, w_i)]$  for  $\mathbf{y} = w_1 \dots w_n$ . By factoring the feature function into these two subsets, argmax can be efficiently searched by the Viterbi algorithm, with its computational complexity proportional to the input length. We list all the baseline features in Table 1<sup>1</sup>. The asterisks (\*) indicate the feature is used for Japanese (JA) but not for Chinese (ZH) WS. Here,  $w_i$  and  $w_{i-1}$  denote the current and previous word in question, and  $t_i^j$  and  $t_{i-1}^j$  are level- $j$  PoS tags assigned to them.  $l(w)$  and  $c(w)$  are the length and the set of character types of word  $w$ .

If there is a substring for which no dictionary entries are found, the *unknown word model* is invoked. In Japanese, our unknown word model relies on heuristics based on character types and word length to generate word nodes, similar to that of McCab (Kudo et al., 2004). In Chinese, we aggregated consecutive 1 to 4 characters add them as “n (common noun)”, “ns (place name)”, “nr (personal name)”, and “nz (other proper nouns),” since most of the unknown words in Chinese are proper nouns. Also, we aggregated up to 20 consecutive numerical characters, making them a single node, and assign “m (number)”. For other character types, a single node with PoS “w (others)” is created.

<sup>1</sup>The Japanese dictionary and the corpus we used have 6 levels of PoS tag hierarchy, while the Chinese ones have only one level, which is why some of the PoS features are not included in Chinese. As character type, Hiragana (JA), Katakana (JA), Latin alphabet, Number, Chinese characters, and Others, are distinguished. Word length is in Unicode.



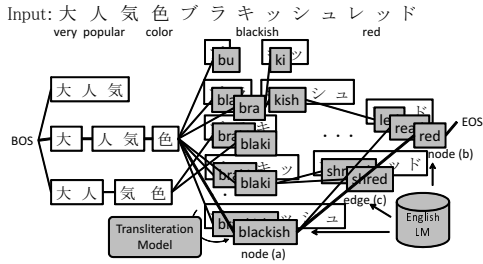


Figure 1: Example lattice with LM projection

## 4 Use of Language Model

**Language Model Augmentation** Analogous to Koehn and Knight (2003), we can exploit the fact that レッド *reddo* (red) in the example ブラキシユレット is such a common word that one can expect it appears frequently in the training corpus. To incorporate this intuition, we used log probability of  $n$ -gram as features, which are included in Table 1 (ID 19 and 20):  $\phi_1^{LMS}(w_i) = \log p(w_i)$  and  $\phi_2^{LMS}(w_{i-1}, w_i) = \log p(w_{i-1}, w_i)$ . Here the empirical probability  $p(w_i)$  and  $p(w_{i-1}, w_i)$  are computed from the source language corpus. In Japanese, we applied this source language augmentation only to Katakana words. In Chinese, we did not limit the target.

### 4.1 Language Model Projection

As we mentioned in Section 2, English LM knowledge helps split transliterated compounds. We use (LM) projection, which is a combination of back-transliteration and an English model, by extending the normal lattice building process as follows:

Firstly, when the lattice is being built, each node is back-transliterated and the resulting nodes are associated with it, as shown in Figure 1 as the shaded nodes. Then, edges are spanned between these extended English nodes, instead of between the original nodes, by additionally taking into consideration English LM features (ID 21 and 22 in Table 1):  $\phi_1^{LMP}(w_i) = \log p(w_i)$  and  $\phi_2^{LMP}(w_{i-1}, w_i) = \log p(w_{i-1}, w_i)$ . Here the empirical probability  $p(w_i)$  and  $p(w_{i-1}, w_i)$  are computed from the English corpus. For example, Feature 21 is set to  $\phi_1^{LMP}$ (“blackish”) for node (a), to  $\phi_1^{LMP}$ (“red”) for node (b), and Feature 22 is set to  $\phi_2^{LMP}$ (“blackish”, “red”) for edge (c) in Figure 1. If no transliterations were generated, or the  $n$ -grams do not appear in the English

corpus, a small frequency  $\varepsilon$  is assumed.

Finally, the created edges are traversed from EOS, and associated original nodes are chosen as the WS result. In Figure 1, the bold edges are traversed at the final step, and the corresponding nodes “大 - 人気 - 色 - ブラキシユレット” are chosen as the final WS result.

For Japanese, we only expand and project Katakana noun nodes (whether they are known or unknown words) since transliterated words are almost always written in Katakana. For Chinese, only “ns (place name)”, “nr (personal name)”, and “nz (other proper noun)” nodes whose surface form is more than 1-character long are transliterated. As the English LM, we used Google Web 1T 5-gram Version 1 (Brants and Franz, 2006), limiting it to unigrams occurring more than 2000 times and bigrams occurring more than 500 times.

## 5 Transliteration

For transliterating Japanese/Chinese words back to English, we adopted the Joint Source Channel (JSC) Model (Li et al., 2004), a generative model widely used as a simple yet powerful baseline in previous research e.g., (Hagiwara and Sekine, 2012; Finch and Sumita, 2010).<sup>2</sup> The JSC model, given an input of source word  $s$  and target word  $t$ , defines the transliteration probability based on transliteration units (TUs)  $u_i = \langle s_i, t_i \rangle$  as:  $P_{JSC}(\langle s, t \rangle) = \prod_{i=1}^f P(u_i | u_{i-n+1}, \dots, u_{i-1})$ , where  $f$  is the number of TUs in a given source / target word pair. TUs are atomic pair units of source / target words, such as “la/ラ” and “ish/イッシュ”. The TU  $n$ -gram probabilities are learned from a training corpus by following iterative updates similar to the EM algorithm<sup>3</sup>. In order to generate transliteration candidates, we used a stack decoder described in (Hagiwara and Sekine, 2012). We used the training data of the NEWS 2009 workshop (Li et al., 2009a; Li et al., 2009b).

As reference, we measured the performance on its own, using NEWS 2009 (Li et al., 2009b) data. The percentage of correctly transliterated words are 37.9% for Japanese and 25.6%

<sup>2</sup>Note that one could also adopt other generative / discriminative transliteration models, such as (Jiampojamarn et al., 2007; Jiampojamarn et al., 2008).

<sup>3</sup>We only allow TUs whose length is shorter than or equal to 3, both in the source and target side.

for Chinese. Although the numbers seem low at a first glance, Chinese back-transliteration itself is a very hard task, mostly because Chinese phonology is so different from English that some sounds may be dropped when transliterated. Therefore, we can regard this performance as a lower bound of the transliteration module performance we used for WS.

## 6 Experiments

### 6.1 Experimental Settings

**Corpora** For Japanese, we used (1) EC corpus, consists of 1,230 product titles and descriptions randomly sampled from Rakuten (Rakuten-Inc., 2012). The corpus is manually annotated with the BCCWJ style WS (Ogura et al., 2011). It consists of 118,355 tokens, and has a relatively high percentage of Katakana words (11.2%). (2) BCCWJ (Maekawa, 2008) CORE (60,374 sentences, 1,286,899 tokens, out of which approx. 3.58% are Katakana words). As the dictionary, we used UniDic (Den et al., 2007). For Chinese, we used LCMC (McEnery and Xiao, 2004) (45,697 sentences and 1,001,549 tokens). As the dictionary, we used CC-CEDICT (MDGB, 2011)<sup>4</sup>.

**Training and Evaluation** We used Averaged Perceptron (Collins, 2002) (3 iterations) for training, with five-fold cross-validation. As for the evaluation metrics, we used Precision (Prec.), Recall (Rec.), and F-measure (F). We additionally evaluated the performance limited to Katakana (JA) or proper nouns (ZH) in order to see the impact of compound splitting. We also used word error rate (WER) to see the relative change of errors.

### 6.2 Japanese WS Results

We compared the baseline model, the augmented model with the source language (+LM-S) and the projected model (+LM-P). Table 3 shows the result of the proposed models and major open-source Japanese WS systems, namely, MeCab 0.98 (Kudo et al., 2004), JUMAN 7.0 (Kurohashi and Nagao, 1994),

<sup>4</sup>Since the dictionary is not explicitly annotated with PoS tags, we firstly took the intersection of the training corpus and the dictionary words, and assigned all the possible PoS tags to the words which appeared in the corpus. All the other words which do not appear in the training corpus are discarded.

and KyTea 0.4.2 (Neubig et al., 2011)<sup>5</sup>. We observed slight improvement by incorporating the source LM, and observed a 0.48 point F-value increase over baseline, which translates to 4.65 point Katakana F-value change and 16.0% (3.56% to 2.99 %) WER reduction, mainly due to its higher Katakana word rate (11.2%). Here, MeCab+UniDic achieved slightly better Katakana WS than the proposed models. This may be because it is trained on a much larger training corpus (the whole BCCWJ). The same trend is observed for BCCWJ corpus (Table 2), where we gained statistically significant 1 point F-measure increase on Katakana word.

Many of the improvements of +LM-S over Baseline come from finer grained splitting, for example, \* レインスーツ *reinsuutsu* “rain suits” to レイン/スーツ, while there is wrong over-splitting, e.g., テレキャスター *terekyasutaa* “Telecaster” to \* テレ/キャスター. This type of error is reduced by +LM-P, e.g., \* プラス/チック *purasu chikku* “\*plus tick” to プラスチック *purasuchikku* “plastic” due to LM projection. +LM-P also improved compounds whose components do not appear in the training data, such as \* ルーカスフィルム *ruukasufirumu* to ルーカス/フィルム “Lucus Film.” Indeed, we randomly extracted 30 Katakana differences between +LM-S and +LM-P, and found out that 25 out of 30 (83%) are true improvement. One of the proposed method’s advantages is that it is very robust to variations, such as アクティベイト *akutibeitiddo* “activated,” even though only the original form, アクティベイト *akutibeito* “activate” is in the dictionary.

One type of errors can be attributed to non-English words such as スノコベッド *sunokobeddo*, which is a compound of Japanese word スノコ *sunoko* “duckboard” and an English word ベッド *beddo* “bed.”

### 6.3 Chinese WS Results

We compare the results on Chinese WS, with Stanford Segmenter (Tseng et al., 2005) (Table 4)<sup>6</sup>. Including +LM-S *decreased* the

<sup>5</sup>Because MeCab+UniDic and KyTea models are actually trained on BCCWJ itself, this evaluation is not meaningful but just for reference. The WS granularity of IPADic, JUMAN, and KyTea is also different from the BCCWJ style.

<sup>6</sup>Note that the comparison might not be fair since (1) Stanford segmenter’s criteria are different from

| Model         | Prec. (O) | Rec. (O) | F (O)   | Prec. (K) | Rec. (K) | F (K)   | WER     |
|---------------|-----------|----------|---------|-----------|----------|---------|---------|
| MeCab+IPADic  | 91.28     | 89.87    | 90.57   | 88.74     | 82.32    | 85.41   | 12.87   |
| MeCab+UniDic* | (98.84)   | (99.33)  | (99.08) | (96.51)   | (97.34)  | (96.92) | (1.31)  |
| JUMAN         | 85.66     | 78.15    | 81.73   | 91.68     | 88.41    | 90.01   | 23.49   |
| KyTea*        | (81.84)   | (90.12)  | (85.78) | (99.57)   | (99.73)  | (99.65) | (20.02) |
| Baseline      | 96.36     | 96.57    | 96.47   | 84.83     | 84.36    | 84.59   | 4.54    |
| +LM-S         | 96.36     | 96.57    | 96.47   | 84.81     | 84.36    | 84.59   | 4.54    |
| +LM-S+LM-P    | 96.39     | 96.61    | 96.50   | 85.59     | 85.40    | 85.50   | 4.50    |

Table 2: Japanese WS Performance (%) on BCCWJ — Overall (O) and Katakana (K)

| Model        | Prec. (O) | Rec. (O) | F (O) | Prec. (K) | Rec. (K) | F (K) | WER   |
|--------------|-----------|----------|-------|-----------|----------|-------|-------|
| MeCab+IPADic | 84.36     | 87.31    | 85.81 | 86.65     | 73.47    | 79.52 | 20.34 |
| MeCab+UniDic | 95.14     | 97.55    | 96.33 | 93.88     | 93.22    | 93.55 | 5.46  |
| JUMAN        | 90.99     | 87.13    | 89.2  | 92.37     | 88.02    | 90.14 | 14.56 |
| KyTea        | 82.00     | 86.53    | 84.21 | 93.47     | 90.32    | 91.87 | 21.90 |
| Baseline     | 97.50     | 97.00    | 97.25 | 89.61     | 85.40    | 87.45 | 3.56  |
| +LM-S        | 97.79     | 97.37    | 97.58 | 92.58     | 88.99    | 90.75 | 3.17  |
| +LM-S+LM-P   | 97.90     | 97.55    | 97.73 | 93.62     | 90.64    | 92.10 | 2.99  |

Table 3: Japanese WS Performance (%) on the EC domain corpus

| Model              | Prec. (O) | Rec. (O) | F (O) | Prec. (P) | Rec. (P) | F (P) | WER   |
|--------------------|-----------|----------|-------|-----------|----------|-------|-------|
| Stanford Segmenter | 87.06     | 86.38    | 86.72 | —         | —        | —     | 17.45 |
| Baseline           | 90.65     | 90.87    | 90.76 | 83.29     | 51.45    | 63.61 | 12.21 |
| +LM-S              | 90.54     | 90.78    | 90.66 | 72.69     | 43.28    | 54.25 | 12.32 |
| +LM-P              | 90.90     | 91.48    | 91.19 | 75.04     | 52.11    | 61.51 | 11.90 |

Table 4: Chinese WS Performance (%) — Overall (O) and Proper Nouns (P)

performance, which may be because one cannot limit where the source LM features are applied. This is why the result of +LM-S+LM-P is not shown for Chinese. On the other hand, replacing LM-S with LM-P improved the performance significantly. We found positive changes such as \* 欧麦/尔萨利赫 *oumai/ersalihe* to 欧麦尔/萨利赫 *oumaier/salihe* “Umar Saleh” and \* 领导/人曼德拉 *lingdao/renmandela* to 领导人/曼德拉 *lingdaoren/mandela* “Leader Mandela”. However, considering the overall F-measure increase and proper noun F-measure decrease suggests that the effect of LM projection is not limited to proper nouns but also promoted finer granularity because we observed proper noun recall increase.

One of the reasons which make Chinese LM projection difficult is the corpus allows single tokens with a transliterated part and Chinese affixes, e.g., 马克思主义者 *makesizhuyizhe* “Marxists” (马克思 *makesi* “Marx” + 主义者 *zhuyizhe* “-ist (believers)”) and 尼罗河 *niluohe* “Nile River” (尼罗 *niluo* “Nile” + 河 *he* “-river”). Another source of errors is transliteration accuracy. For example, no ap-

ours, and (2) our model only uses the intersection of the training set and the dictionary. Proper noun performance for the Stanford segmenter is not shown since it does not assign PoS tags.

propriate transliterations were generated for 维娜斯 *weinasi* “Venus,” which is commonly spelled 维纳斯 *weinasi*. Improving the JSC model could improve the LM projection performance.

## 7 Conclusion and Future Works

In this paper, we proposed a novel, on-line WS model for the Japanese/Chinese compound word splitting problem, by seamlessly incorporating the knowledge that back-transliteration of properly segmented words also appear in an English LM. The experimental results show that the model achieves a significant improvement over the baseline and LM augmentation, achieving 16% WER reduction in the EC domain.

The concept of LM projection is general enough to be used for splitting other compound nouns. For example, for Japanese personal names such as 仲里依紗 *Naka Riisa*, if we could successfully estimate the pronunciation *Nakarīisa* and look up possible splits in an English LM, one is expected to find a correct WS *Naka Riisa* because the first and/or the last name are mentioned in the LM. Seeking broader application of LM projection is a future work.

## References

- Masayuki Asahara and Yuji Matsumoto. 2004. Japanese unknown word identification by character-based chunking. In *Proceedings of COLING 2004*, pages 459–465.
- Thorsten Brants and Alex Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium.
- Martin Braschler and Bärbel Ripplinger. 2004. How effective is stemming and compounding for german text retrieval? *Information Retrieval*, pages 291–316.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8.
- Yasuharu Den, Toshinobu Ogiso, Hideki Ogura, Atsushi Yamada, Nobuaki Minematsu, Kiyotaka Uchimoto, and Hanae Koiso. 2007. The development of an electronic dictionary for morphological analysis and its application to Japanese corpus linguistics (in Japanese). *Japanese linguistics*, 22:101–122.
- Andrew Finch and Eiichiro Sumita. 2010. A bayesian model of bilingual segmentation for transliteration. In *Proceedings of IWSLT 2010*, pages 259–266.
- Masato Hagiwara and Satoshi Sekine. 2012. Latent class transliteration based on source language origin. In *Proceedings of NEWS 2012*, pages 30–37.
- Sittichai Jiampojarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of NAACL-HLT 2007*, pages 372–379.
- Sittichai Jiampojarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL 2008*, pages 905–913.
- Nobuhiro Kaji and Masaru Kitsuregawa. 2011. Splitting noun compounds via monolingual and bilingual paraphrasing: A study on japanese katakana words. In *Proceedings of the EMNLP 2011*, pages 959–969.
- Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of EACL 2003*, pages 187–193.
- Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of EMNLP 2004*, pages 230–237.
- Sadao Kurohashi and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer juman. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 22–38.
- Gurpreet Singh Lehal. 2010. A word segmentation system for handling space omission problem in urdu script. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, pages 43–50.
- Haizhou Li, Zhang Min, and Su Jian. 2004. A joint source-channel model for machine transliteration. In *Proceedings of ACL 2004*, pages 159–166.
- Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009a. Report of news 2009 machine transliteration shared task. In *Proceedings of NEWS 2009*, pages 1–18.
- Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine. 2009b. Whitepaper of news 2009 machine transliteration shared task. In *Proceedings of NEWS 2009*, pages 19–26.
- Kikuo Maekawa. 2008. Compilation of the Kotonoha-BCCWJ corpus (in Japanese). *Nihongo no kenkyu (Studies in Japanese)*, 4(1):82–95.
- Anthony McEnery and Zhonghua Xiao. 2004. The lancaster corpus of mandarin chinese: A corpus for monolingual and contrastive language study. In *Proceedings of LREC 2004*, pages 1175–1178.
- MDGB. 2011. *CC-CEDICT*, Retrieved August, 2012 from <http://www.mdbg.net/chindict/chindict.php?page=cedict>.
- Shinsuke Mori and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proceedings of COLING 2006*, pages 1119–1122.
- Masaaki Nagata. 1999. A part of speech estimation method for Japanese unknown words using a statistical model of morphology and context. In *Proceedings of ACL 1999*, pages 277–284.
- Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2005. Automatic acquisition of basic katakana lexicon from a given corpus. In *Proceedings of IJCNLP 2005*, pages 682–693.
- Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of ACL-HLT 2011*, pages 529–533.
- Hideki Ogura, Hanae Koiso, Yumi Fujike, Sayaka Miyauchi, and Yutaka Hara. 2011. *Morphological Information Guideline for BCCWJ: Balanced Corpus of Contemporary Written*

*Japanese, 4th Edition*. National Institute for Japanese Language and Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings COLING 2004*.

Rakuten-Inc. 2012. *Rakuten Ichiba*  
<http://www.rakuten.co.jp/>.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. Morphological analysis based on a maximum entropy model — an approach to the unknown word problem — (in Japanese). *Journal of Natural Language Processing*, 8:127–141.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proceedings of ACL 2008*, pages 888–896.

# Broadcast News Story Segmentation Using Manifold Learning on Latent Topic Distributions

Xiaoming Lu<sup>1,2</sup>, Lei Xie<sup>1\*</sup>, Cheung-Chi Leung<sup>2</sup>, Bin Ma<sup>2</sup>, Haizhou Li<sup>2</sup>

<sup>1</sup>School of Computer Science, Northwestern Polytechnical University, China

<sup>2</sup>Institute for Infocomm Research, A\*STAR, Singapore

luxiaomingnpu@gmail.com, lxie@nwpu.edu.cn, {cclleung,mabin,hli}@i2r.a-star.edu.sg

## Abstract

We present an efficient approach for broadcast news story segmentation using a manifold learning algorithm on latent topic distributions. The latent topic distribution estimated by Latent Dirichlet Allocation (LDA) is used to represent each text block. We employ Laplacian Eigenmaps (LE) to project the latent topic distributions into low-dimensional semantic representations while preserving the intrinsic local geometric structure. We evaluate two approaches employing LDA and probabilistic latent semantic analysis (PLSA) distributions respectively. The effects of different amounts of training data and different numbers of latent topics on the two approaches are studied. Experimental results show that our proposed LDA-based approach can outperform the corresponding PLSA-based approach. The proposed approach provides the best performance with the highest F1-measure of 0.7860.

## 1 Introduction

Story segmentation refers to partitioning a multimedia stream into homogenous segments each embodying a main topic or coherent story (Allan, 2002). With the explosive growth of multimedia data, it becomes difficult to retrieve the most relevant components. For indexing broadcast news programs, it is desirable to divide each of them into a number of independent stories. Manual segmentation is accurate but labor-intensive and costly. Therefore, automatic story segmentation approaches are highly demanded.

Lexical-cohesion based approaches have been widely studied for automatic broadcast news story segmentation (Beeferman et al., 1997; Choi, 1999; Hearst, 1997; Rosenberg and Hirschberg, 2006;

Lo et al., 2009; Malioutov and Barzilay, 2006; Yamron et al., 1999; Tur et al., 2001). In this kind of approaches, the audio portion of the data stream is passed to an automatic speech recognition (ASR) system. Lexical cues are extracted from the ASR transcripts. Lexical cohesion is the phenomenon that different stories tend to employ different sets of terms. Term repetition is one of the most common appearances.

These rigid lexical-cohesion based approaches simply take term repetition into consideration, while term association in lexical cohesion is ignored. Moreover, polysemy and synonymy are not considered. To deal with these problems, some topic model techniques which provide conceptual level matching have been introduced to text and story segmentation task (Hearst, 1997). Probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) is a typical instance and used widely. PLSA is the probabilistic variant of latent semantic analysis (LSA) (Choi et al., 2001), and offers a more solid statistical foundation. PLSA provides more significant improvement than LSA for story segmentation (Lu et al., 2011; Blei and Moreno, 2001).

Despite the success of PLSA, there are concerns that the number of parameters in PLSA grows linearly with the size of the corpus. This makes PLSA not desirable if there is a considerable amount of data available, and causes serious over-fitting problems (Blei, 2012). To deal with this issue, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) has been proposed. LDA has been proved to be effective in many segmentation tasks (Arora and Ravindran, 2008; Hall et al., 2008; Sun et al., 2008; Riedl and Biemann, 2012; Chien and Chueh, 2012).

Recent studies have shown that intrinsic dimensionality of natural text corpus is significantly lower than its ambient Euclidean space (Belkin and Niyogi, 2002; Xie et al., 2012). Therefore,

\*corresponding author

Laplacian Eigenmaps (LE) was proposed to compute corresponding natural low-dimensional structure. LE is a geometrically motivated dimensionality reduction method. It projects data into a low-dimensional representation while preserving the intrinsic local geometric structure information (Belkin and Niyogi, 2002). The locality preserving property attempts to make the low-dimensional data representation more robust to the noise from ASR errors (Xie et al., 2012).

To further improve the segmentation performance, using latent topic distributions and LE instead of term frequencies to represent text blocks is studied in this paper. We study the effects of the size of training data and the number of latent topics on the LDA-based and the PLSA-based approaches. Another related work (Lu et al., 2013) is to use local geometric information to regularize the log-likelihood computation in PLSA.

## 2 Our Proposed Approach

In this paper, we propose to apply LE on the LDA topic distributions, each of which is estimated from a text block. The low-dimensional vectors obtained by LE projection are used to detect story boundaries through dynamic programming. Moreover, as in (Xie et al., 2012), we incorporate the temporal distances between block pairs as a penalty factor in the weight matrix.

### 2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is a generative probabilistic model of a corpus. It considers that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over terms.

In LDA, given a corpus  $D = \{d_1, d_2, \dots, d_M\}$  and a set of terms  $W = (w_1, w_2, \dots, w_V)$ , the generative process can be summarized as follows:

1) For each document  $d$ , pick a multinomial distribution  $\theta$  from a Dirichlet distribution parameter  $\alpha$ , denoted as  $\theta \sim Dir(\alpha)$ .

2) For each term  $w$  in document  $d$ , select a topic  $z$  from the multinomial distribution  $\theta$ , denoted as  $z \sim Multinomial(\theta)$ .

3) Select a term  $w$  from  $P(w|z, \beta)$ , which is a multinomial probability conditioned on the topic.

An LDA model is characterized by two sets of prior parameters  $\alpha$  and  $\beta$ .  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$  represents the Dirichlet prior distributions for each  $K$  latent topics.  $\beta$  is a  $K \times V$  matrix, which defines the latent topic distributions over terms.

### 2.2 Construction of weight matrix in Laplacian Eigenmaps

Laplacian Eigenmaps (LE) is introduced to project high-dimensional data into a low-dimensional representation while preserving its locality property. Given the ASR transcripts of  $N$  text blocks, we apply LDA algorithm to compute the corresponding latent topic distributions  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  in  $\mathbb{R}^K$ , where  $K$  is the number of latent topics, namely the dimensionality of LDA distributions.

We use  $G$  to denote an  $N$ -node ( $N$  is number of LDA distributions) graph which represents the relationship between all the text block pairs. If distribution vectors  $\mathbf{x}_i$  and  $\mathbf{x}_j$  come from the same story, we put an edge between nodes  $i$  and  $j$ . We define a weight matrix  $\mathbf{S}$  of the graph  $G$  to denote the cohesive strength between the text block pairs. Each element of this weight matrix is defined as:

$$s_{ij} = \cos(\mathbf{x}_i, \mathbf{x}_j) \mu^{|i-j|}, \quad (1)$$

where  $\mu^{|i-j|}$  serves the penalty factor for the distance between  $i$  and  $j$ .  $\mu$  is a constant lower than 1.0 that we tune from a set of development data. It makes the cohesive strength of two text blocks dramatically decrease when their distance is much larger than the normal length of a story.

### 2.3 Data projection in Laplacian Eigenmaps

Given the weight matrix  $\mathbf{S}$ , we define  $\mathbf{C}$  as the diagonal matrix with its element:

$$c_{ij} = \sum_{i=1}^K s_{ij}. \quad (2)$$

Finally, we obtain the Laplacian matrix  $\mathbf{L}$ , which is defined as:

$$\mathbf{L} = \mathbf{C} - \mathbf{S}. \quad (3)$$

We use  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$  ( $\mathbf{y}_i$  is a column vector) to indicate the low-dimensional representation of the latent topic distributions  $\mathbf{X}$ . The projection from the latent topic distribution space to the target space can be defined as:

$$f : \mathbf{x}_i \Rightarrow \mathbf{y}_i. \quad (4)$$

A reasonable criterion for computing an optimal mapping is to minimize the objective as follows:

$$\sum_{i=1}^K \sum_{j=1}^K \|\mathbf{y}_i - \mathbf{y}_j\|^2 s_{ij}. \quad (5)$$

Under this constraint condition, we can preserve the local geometrical property in LDA distributions. The objective function can be transformed

as:

$$\sum_{i=1}^K \sum_{j=1}^K (\mathbf{y}_i - \mathbf{y}_j) s_{ij} = \text{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y}). \quad (6)$$

Meanwhile, zero matrix and matrices with its rank less than  $K$  are meaningless solutions for our task. We impose  $\mathbf{Y}^T \mathbf{L} \mathbf{Y} = \mathbf{I}$  to prevent this situation, where  $\mathbf{I}$  is an identity matrix. By the Reyleigh-Ritz theorem (Lutkepohl, 1997), the solution can be obtained by the  $Q$  smallest eigenvalues of the generalized eigenmaps problem:

$$\mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{y} = \lambda \mathbf{X} \mathbf{C} \mathbf{X}^T \mathbf{y}. \quad (7)$$

With this formula, we calculate the mapping matrix  $\mathbf{Y}$ , and its row vectors  $\mathbf{y}'_1, \mathbf{y}'_2, \dots, \mathbf{y}'_Q$  are in the order of their eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_Q$ .  $\mathbf{y}'_i$  is a  $Q$ -dimensional ( $Q < K$ ) eigenvectors.

## 2.4 Story boundary detection

In story boundary detection, dynamic programming (DP) approach is adopted to obtain the global optimal solution. Given the low-dimensional semantic representation of the test data, an objective function can be defined as follows:

$$\mathfrak{S} = \sum_{t=1}^{N_s} \left( \sum_{i,j \in \text{Seg}_t} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \right), \quad (8)$$

where  $\mathbf{y}_i$  and  $\mathbf{y}_j$  are the latent topic distributions of text blocks  $i$  and  $j$  respectively, and  $\|\mathbf{y}_i - \mathbf{y}_j\|^2$  is the Euclidean distance between them.  $\text{Seg}_t$  indicates these text blocks assigned to a certain hypothesized story.  $N_s$  is the number of hypothesized stories.

The story boundaries which minimize the objective function  $\mathfrak{S}$  in Eq.(8) form the optimal result. Compared with classical local optimal approach, DP can more effectively capture the smooth story shifts, and achieve better segmentation performance.

## 3 Experimental setup

Our experiments were evaluated on the ASR transcripts provided in TDT2 English Broadcast news corpus<sup>1</sup>, which involved 1033 news programs. We separated this corpus into three non-overlapping sets: a training set of 500 programs for parameter estimation in topic modeling and LE, a development set of 133 programs for empirical tuning and a test set of 400 programs for performance evaluation.

In the training stage, ASR transcripts with manually labeled boundary tags were provided. Text

<sup>1</sup><http://projects.ldc.upenn.edu/TDT2/>

streams were broken into block units according to the given boundary tags, with each text block being a complete story. In the segmentation stage, we divided test data into text blocks using the time labels of pauses in the transcripts. If the pause duration between two blocks last for more than 1.0 sec, it was considered as a boundary candidate. To avoid the segmentation being suffered from ASR errors and the out-of-vocabulary issue, phoneme bigram was used as the basic term unit (Xie et al., 2012). Since the ASR transcripts were at word level, we performed word-to-phoneme conversion to obtain the phoneme bigram basic units. The following approaches, in which DP was used in story boundary detection, were evaluated in the experiments:

- PLSA-DP: PLSA topic distributions were used to compute sentence cohesive strength.
- LDA-DP: LDA topic distributions were used to compute sentence cohesive strength.
- PLSA-LE-DP: PLSA topic distributions followed by LE projection were used to compute sentence cohesive strength.
- LDA-LE-DP: LDA topic distributions followed by LE projection were used to compute sentence cohesion strength.

For LDA, we used the implementation from David M. Blei’s webpage<sup>2</sup>. For PLSA, we used the Lemur Toolkit<sup>3</sup>.

F1-measure was used as the evaluation criterion. We followed the evaluation rule: a detected boundary candidate is considered correct if it lies within a 15 sec tolerant window on each side of a reference boundary. A number of parameters were set through empirical tuning on the development set. The penalty factor was set to 0.8. When evaluating the effects of different size of the training set, the number of latent topics in topic modeling process was set to 64. After the number of latent topics was fixed, the dimensionality after LE projection was set to 32. When evaluating the effects of different number of latent topics in topic modeling computation, we fixed the size of the training set to 500 news programs and changed the number of latent topics from 16 to 256.

## 4 Experimental results and analysis

### 4.1 Effect of the size of training dataset

We used the training set from 100 programs to 500 programs (adding 100 programs in each step) to e-

<sup>2</sup><http://www.cs.princeton.edu/blei/lda-c/>

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



evaluate the effects of different size of training data in both PLSA-based and LDA-based approaches. Figure 1 shows the results on the development set and the test set.

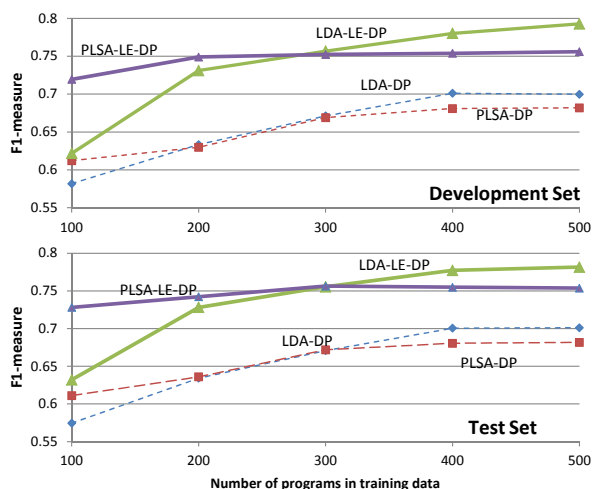


Figure 1: Segmentation performance with different amounts of training data

LDA-LE-DP approach achieved the best result (0.7927 and 0.7860) on both the development and the test sets, when there were 500 programs in the training set. This demonstrates that LDA model and LE projection used in combination is excellent for the story segmentation task. The LE projection applied on the latent topic representations made relatively 9.88% and 10.93% improvement over the LDA-based approach and the PLSA-based approach, respectively on the test set. We can reveal that employing LE on PLSA and LDA topic distributions achieves much better performance than the corresponding approaches without using LE.

We have compared the performances between PLSA and LDA. We found that when the training data size was small, PLSA performed better than LDA. Both PLSA-based and LDA-based approaches got better with the increase in the size of the training data set. All the four approaches had similar performances on the development set and the test set.

With the increase in the size of the training data, the LDA-based approaches were improved dramatically. They even outperformed the PLSA-based approaches when the training data contained more than 300 programs. This may be attributed to the fact that LDA needs more training data to estimate the parameters. When the training data is not enough, its parameters estimated in the training stage is not stable for the development and the

test data. Moreover, compared with PLSA, the parameters in LDA do not grow linearly with the size of the corpus.

## 4.2 Effect of the number of latent topics

We evaluated the F1-measure of the four approaches with different number of latent topics prior to LE projection. Figure 2 shows the corresponding results.

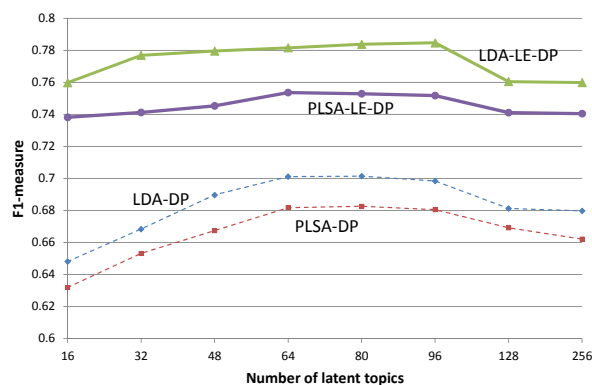


Figure 2: Segmentation performance with different numbers of latent topics

The best performances (0.7816-0.7847) were achieved at the number of latent topics between 64 and 96. When the number of latent topics was increased from 16 to 64, F1-measure increased. When the number of latent topics was larger than 96, F1-measure decreased gradually. We found that the best results were achieved when the number of topics was close to the real number of topics. There are 80 manually labeled main topics in the test set.

We observe that LE projection makes the topic model more stable with different numbers of latent topics. The best and the worst performances differed by relatively 9.12% in LDA-DP and 7.97% in PLSA-DP. However, the relative difference of 2.79% and 2.46% were observed in LDA-LE-DP and PLSA-LE-DP respectively.

## 5 Conclusions

Our proposed approach achieves the best F1-measure of 0.7860. In the task of story segmentation, we believe that LDA can avoid data overfitting problem when there is a sufficient amount of training data. This is also applicable to LDA-LE-LP. Moreover, we find that when we apply LE projection to latent topic distributions, the segmentation performances become less sensitive to the predefined number of latent topics.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61175018), the Natural Science Basic Research Plan of Shaanxi Province (2011JM8009) and the Fok Ying Tung Education Foundation (131059).

## References

- J. Allan. 2002. *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publisher, Norwell, MA.
- Doug Beeferman, Adam Berger, and John Lafferty. 1997. *A Model of Lexical Attraction and repulsion*. In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pp.373-380.
- Freddy Y. Y. Choi. 2000. *Advances in Domain Independent Linear Text Segmentation*. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference (NAACL)*, pp.26-33.
- Thomas Hofmann. 1999. *Probabilistic Latent Semantic Indexing*. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp.20-57.
- Mimi Lu, Cheung-Chi Leung, Lei Xie, Bin Ma, Haizhou Li. 2011. *Probabilistic Latent Semantic Analysis for Broadcast News Segmentation*. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp.1109-1112.
- David M. Blei. 2012. *Probabilistic topic models*. *Communication of the ACM*, vol. 55, pp.77-84.
- David M. Blei, Andrew Y. Ng, Michael I. Jordan. 2003. *Latent Dirichlet Allocation*. *the Journal of Machine Learning Research*, vol. 3, pp.993-1022.
- Marti A. Hearst. 1997. *TextTiling: Segmenting Text into Multiparagraph subtopic passages*. *Computational Linguistic*, vol. 23, pp.33-64.
- Gokhan Tur, Dilek Hakkani-Tur, Andreas Stolcke, Elizabeth Shriberg. 2001. *Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation*. *Computational Linguistic*, vol. 27, pp.31-57.
- Andrew Rosenberg and Julia Hirschberg. 2006. *Story Segmentation of Broadcast News in English, Mandarin and Arabic*. In *Proceedings of the 7th North American Chapter of the Association for Computational Linguistics Conference (NAACL)*, pp.125-128.
- David M. Blei and Pedro J. Moreno. 2001. *Topic Segmentation with An Aspect Hidden Markov Model*. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp.343-348.
- Wai-Kit Lo, Wenying Xiong, Helen Meng. 2009. *Automatic Story Segmentation Using a Bayesian Decision Framework for Statistical Models of Lexical Chain Feature*. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.357-364.
- Igor Malioutov and Regina Barzilay. 2006. *Minimum Cut Model for Spoken Lecture Segmentation*. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.25-32.
- Freddy Y. Y. Choi, Peter Wiemer-Hastings, Juhanna Moore. 2001. *Latent Semantic Analysis for Text Segmentation*. In *Proceedings of the 2001 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pp.109-117.
- Rachit Arora and Balaraman Ravindran. 2008. *Latent Dirichlet Allocation Based Multi-document Summarization*. In *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data (AND)*, pp.91-97.
- David Hall, Daniel Jurafsky, Christopher D. Manning. 2008. *Latent Studying the History Ideas Using Topic Models*. In *Proceedings of the 2008 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pp.363-371.
- Qi Sun, Runxin Li, Dingsheng Luo, Xihong Wu. 2008. *Text Segmentation with LDA-based Fisher Kernel*. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT-ACL)*, pp.269-272.
- Mikhail Belkin and Partha Niyogi. 2002. *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*. *Neural Computation*, vol. 15, pp.1383-1396.
- Lei Xie, Lilei Zheng, Zihan Liu and Yanning Zhang. 2012. *Laplacian Eigenmaps for Automatic Story Segmentation of Broadcast News*. *IEEE Transaction on Audio, Speech and Language Processing*, vol. 20, pp.264-277.
- Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. 2008. *Modeling Hidden Topics on Document Manifold*. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pp.911-120.
- Xiaoming Lu, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li. 2013. *Broadcast News Story Segmentation Using Latent Topics on Data Manifold*. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

- J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. 1999. *A Hidden Markov Model Approach to Text Segmentation and Event Tracking*. In *Proceedings of the 1999 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.333-336.
- Martin Riedl and Chris Biemann. 2012. *Text Segmentation with Topic Models*. the Journal for Language Technology and Computational Linguistics, pp.47-69.
- P. Fragkou , V. Petridis , Ath. Kehagias. 2002. *A Dynamic Programming algorithm for Linear Text Story Segmentation*. the Journal of Intelligent Information Systems, vol. 23, pp.179-197.
- H. Lutkepohl. 1997. *Handbook of Matrices*. Wiley, Chichester, UK.
- Jen-Tzung Chien and Chuang-Hua Chueh. 2012. *Topic-Based Hierarchical Segmentation*. IEEE Transaction on Audio, Speech and Language Processing, vol. 20, pp.55-66.

# Is word-to-phone mapping better than phone-phone mapping for handling English words?

**Naresh Kumar Elluru**

Speech and Vision Lab  
IIIT Hyderabad, India  
nareshkumar.elluru@  
research.iiit.ac.in

**Anandaswarup Vadapalli**

Speech and Vision Lab  
IIIT Hyderabad, India  
anandaswarup.vadapalli@  
research.iiit.ac.in

**Raghavendra Elluru**

Speech and Vision Lab  
IIIT Hyderabad, India  
raghavendra.veera@  
gmail.com

**Hema Murthy**

Department of CSE  
IIT Madras, India  
hema@iitm.ac.in

**Kishore Prahallad**

Speech and Vision Lab  
IIIT Hyderabad, India  
kishore@iiit.ac.in

## Abstract

In this paper, we relook at the problem of pronunciation of English words using native phone set. Specifically, we investigate methods of pronouncing English words using Telugu phoneset in the context of Telugu Text-to-Speech. We compare phone-phone substitution and word-phone mapping for pronunciation of English words using Telugu phones. We are not considering other than native language phoneset in all our experiments. This differentiates our approach from other works in polyglot speech synthesis.

## 1 Introduction

The objective of a Text-to-Speech (TTS) system is to convert a given text input into a spoken waveform. Text processing and waveform generation are the two main components of a TTS system. The objective of the text processing component is to convert the given input text into an appropriate sequence of valid phonemic units. These phonemic units are then realized by the waveform generation component. For high quality speech synthesis, it is necessary that the text processing unit produce the appropriate sequence of phonemic units, for the given input text.

There has been a rise in the phenomenon of “code mixing” (Romaine and Kachru, 1992). This is a phenomenon where lexical items of two languages appear in a single sentence. In a multilingual country such as India, we commonly find Indian language text being freely interspersed with English words and phrases. This is particularly noticeable in the case of text from web sources like

blogs, tweets etc. An informal analysis of a Telugu blog on the web showed that around 20-30% of the text is in English (ASCII) while the remaining is in Telugu (Unicode). Due to the growth of “code mixing” it has become necessary to develop strategies for dealing with such multilingual text in TTS systems. These multilingual TTS systems should be capable of synthesizing utterances which contain foreign language words or word groups, without sounding unnatural.

The different ways of achieving multilingual TTS synthesis are as follows (Traber et al., 1999; Latorre et al., 2006; Campbell, 1998; Campbell, 2001).

### 1. Separate TTS systems for each language:

In this paradigm, a separate TTS system is built for each language under consideration. When the language of the input text changes, the TTS system also has to be changed. This can only be done between two sentences/utterances and not in the middle of a sentence.

### 2. Polyglot speech synthesis:

This is a type of multilingual speech synthesis achieved using a single TTS system. This method involves recording a multi language speech corpus by someone who is fluent in multiple languages. This speech corpus is then used to build a multilingual TTS system. The primary issue with polyglot speech synthesis is that it requires development of a combined phoneset, incorporating phones from all the languages under consideration. This is a time consuming process requiring linguistic knowledge of both languages. Also, finding a speaker fluent in mul-

tiple languages is not an easy task.

### 3. Phone mapping:

This type of multilingual synthesis is based upon phone mapping, whereby the phones of the foreign language are substituted with the closest sounding phones of the primary language. This method results in a strong foreign accent while synthesizing the foreign words. This may not always be acceptable. Also, if the sequence of the mapped phones does not exist or is not frequently occurring in the primary language, then the synthesized output quality would be poor. Hence, an average polyglot synthesis technique using HMM based synthesis and speaker adaptation has been proposed (Latorre et al., 2006). Such methods make use of speech data from different languages and different speakers.

In this paper, we relook at the problem of pronunciation of English words using native phone set. Specifically, we investigate methods of pronouncing English words using Telugu phoneset in the context of Telugu Text-to-Speech. *Our motivation for doing so, comes from our understanding of how humans pronounce foreign words while speaking. The speaker maps the foreign words to a sequence of phones of his/her native language while pronouncing that foreign word. For example, a native speaker of Telugu, while pronouncing an English word, mentally maps the English word to a sequence of Telugu phones as opposed to simply substituting English phones with the corresponding Telugu phones.* Also, the receiver of the synthesized speech would be a Telugu native speaker, who may not have the knowledge of English phone set. Hence, approximating an English word using Telugu phone sequence may be more acceptable for a Telugu native speaker.

We compare phone-phone substitution and word-phone mapping (also referred to LTS rules) for the pronunciation of English words using Telugu phones. We are not considering other than native language phoneset in all our experiments. This differentiates our work from other works in polyglot speech synthesis.

## 2 Comparison of word-phone and phone-phone mapping

Table 1 shows an example of the word *computer* represented as a US English phone sequence, En-

| Computer            |  |
|---------------------|--|
| US English Phones   | $\frac{/k \text{ ax m p y uw t er/}}{[k \text{ ə m p j u t ʔ}]}$     |
| phone-phone mapping | $\frac{/k \text{ e m p y uu t: r/}}{[k \text{ e m p j u: t r}]}$     |
| word-phone mapping  | $\frac{/k \text{ a m p y uu t: a r/}}{[k \text{ a m p j u: t a r}]}$ |

Table 1: English word *computer* represented as US English phone sequence, US English phone-Telugu phone mapping and English word-Telugu phone mapping

glish phone-Telugu phone mapping and English word-Telugu phone mapping, along with the corresponding IPA transcription. The English word-Telugu phone mapping is not a one to one mapping, as it is in the case of English phone-Telugu phone mapping. Each letter has a correspondence with one or more than one phones. As some letters do not have a equivalent pronunciation sound (the letter is not mapped to any phone) the term `_epsilon_` is used whenever there is a letter which does not have a mapping with a phone.

To compare word-phone (W-P) mapping and phone-phone (P-P) mapping, we manually prepared word-phone and phone-phone mappings for 10 bilingual utterances and synthesized them using our baseline Telugu TTS system. We then performed perceptual listening evaluations on these synthesized utterances, using five native speakers of Telugu as the subjects of the evaluations. The perceptual listening evaluations were setup both as MOS (mean opinion score) evaluations and as ABX evaluations. An explanation of MOS and ABX evaluations is given in Section 4. Table 2 shows that results of these evaluations.

| MOS  |      | ABX   |      |          |
|------|------|-------|------|----------|
| W-P  | P-P  | W-P   | P-P  | No. Pref |
| 3.48 | 2.66 | 32/50 | 4/50 | 14/50    |

Table 2: Perceptual evaluation scores for baseline Telugu TTS system with different pronunciation rules for English

An examination of the results in Table 2 shows that manually prepared word-phone mapping is preferred perceptually when compared to manual phone-phone mapping. The MOS score of 3.48 indicates that native speakers accept W-P mapping for pronouncing English words in Telugu TTS.

For the remainder of this paper, we focus exclusively on word-phone mapping. We propose a method of automatically generating these word-phone mapping from data. We experiment our approach by generating a word-phone mapping which maps each English word to a Telugu phone sequence (henceforth called EW-TP mapping). We report the accuracy of learning the word-phone mappings both on a held out test set and on a test set from a different domain. Finally, we incorporate this word-phone mapping in our baseline Telugu TTS system and demonstrate its usefulness by means of perceptual listening tests.

### 3 Automatic generation of word-phone mapping

We have previously mentioned that letter to phone mapping is not a one to one mapping. Each letter may have a correspondence with one or more than one phones, or it may not have correspondence with any phone. As we require a fixed sized learning vector to build a model for learning word-phone mapping rules, we need to align the letter (graphemic) and phone sequences. For this we use the automatic epsilon scattering method.

#### 3.1 Automatic Epsilon Scattering Method

The idea in automatic epsilon scattering is to estimate the probabilities for one letter (grapheme)  $G$  to match with one phone  $P$ , and then use string alignment to introduce epsilons maximizing the probability of the word's alignment path. Once the all the words have been aligned, the association probability is calculated again and so on until convergence. The algorithm for automatic epsilon scattering is given below (Pagel et al., 1998).

#### 3.2 Evaluation and Results

Once the alignment between the each word and the corresponding phone sequence was complete, we built two phone models using Classification and Regression Trees (CART). For the first model, we used data from the CMU pronunciation dictionary where each English word had been aligned to a sequence of US English phones (EW-EP mapping).

---

Algorithm for Epsilon Scattering :

```

/*Initialize  $prob(G, P)$  the probability of  $G$ 
matching  $P$ */
1. for each  $word_i$  in training_set
count with string alignment all possible  $G/P$ 
association for all possible epsilon positions in the
phonetic transcription
/* EM loop */
2. for each  $word_i$  in training_set
alignment_path =  $argmax \prod_{i,j} P(G_i, P_j)$ 
compute  $prob_{new}(G, P)$  on alignment_path
3. if( $prob \neq prob_{new}$ ) go to 2

```

---

The second model was the EW-TP mapping.

Once both the models had been built, they were used to predict the mapped phone sequences for each English word in the test data. For the purposes of testing, we performed the prediction on both held out test data as well as on test data from a different domain. The held out test data was prepared by removing every ninth word from the lexicon.

As we knew the correct phone sequence for each word in the test data, a ground truth against which to compute the accuracy of prediction was available. We measured the accuracy of the prediction both at the letter level and at the word level. At the letter level, the accuracy was computed by counting the number of times the predicted letter to phone mapping matched with the ground truth. For computing the accuracy at the word level, we counted the number of times the predicted phone sequence of each word in the test data matched with the actual phone sequence for that word (derived from the ground truth). We also varied the size of the training data and then computed the prediction accuracy for each model. We did so in order to study the effect of training data size on the prediction accuracy.

Tables 3, 4 show the accuracy of the models. An examination of the results in the two tables shows that incrementally increasing the size of the training data results in an increase of the prediction accuracy. The native speakers of Indian languages prefer to speak what is written. As a result there are fewer variations in word-phone mapping as compared to US English. This is reflected in our results, which show that the word level prediction accuracy is higher for EW-TP mapping as compared to EW-EP mapping.

| Training set size | Held-out(%) |       | Testing(%) |       |
|-------------------|-------------|-------|------------|-------|
|                   | Letters     | words | Letters    | words |
| 1000              | 92.04       | 39    | 81.43      | 16.6  |
| 2000              | 94.25       | 44.98 | 82.47      | 17.5  |
| 5000              | 94.55       | 47    | 84.40      | 25.1  |
| 10000             | 95.82       | 59.86 | 89.46      | 44.7  |
| 100000            | 94.09       | 56.37 | 93.27      | 55.10 |

Table 3: Accuracy of prediction for English word - English phone mapping

| Training set size | Held-out(%) |       | Testing(%) |       |
|-------------------|-------------|-------|------------|-------|
|                   | Letters     | words | Letters    | words |
| 1000              | 92.37       | 28    | 82.22      | 18.8  |
| 2000              | 94.34       | 45.45 | 83.79      | 25.1  |
| 5000              | 95.89       | 68.2  | 88.40      | 42.7  |
| 10000             | 96.54       | 71.67 | 94.74      | 70.9  |

Table 4: Accuracy of prediction for English word-Telugu phone mapping

#### 4 Integrating word-phone mapping rules in TTS

For the purpose of perceptual evaluations we built a baseline TTS systems for Telugu using the HMM based speech synthesis technique (Zen et al., 2007).

To conduct perceptual evaluations of the word-phone mapping rules built from data in 3.2, we incorporated these rules in our Telugu TTS system. This system is henceforth referred to as T\_A. A set of 25 bilingual sentences were synthesized by the Telugu TTS, and ten native speakers of Telugu performed perceptual evaluations on the synthesized utterances. As a baseline, we also synthesized the same 25 sentences by incorporating manually written word-phone mapping for the English words, instead of using the automatically generated word-phone mapping rules. We refer to this system as T\_M.

The perceptual evaluations were set up both as MOS (mean opinion score) evaluations and as ABX evaluations. In the MOS evaluations, the listeners were asked to rate the synthesized utterances from all systems on a scale of 1 to 5 (1 being worst and 5 best), and the average scores for each system was calculated. This average is the MOS score for that system. In a typical ABX evaluation, the listeners are presented with the the same

set of utterances synthesized using two systems A and B, and are asked to mark their preference for either A or B. The listeners also have an option of marking no preference. In this case, the listeners were asked to mark their preference between T\_A and T\_M. The results of the perceptual evaluations are shown in Table 5.

| MOS  |      | ABX Test |        |          |
|------|------|----------|--------|----------|
| T_M  | T_A  | T_M      | T_A    | No. Pref |
| 3.48 | 3.43 | 51/250   | 38/250 | 161/250  |

Table 5: Perceptual results comparing systems T\_M and T\_A

An examination of the results shows that perceptually there is no significant preference for the manual system over the automated system. The MOS scores also show that there is not much significant difference between the ratings of the manual and the automated system.

#### 5 Conclusions

In this paper we present a method of automatically learning word-phone mapping rules for synthesizing foreign words occurring in text. We show the effectiveness of the method by computing the accuracy of prediction and also by means of perceptual evaluations. The synthesized multilingual wave files are available for download at <https://www.dropbox.com/s/7hja51r5rpkz5mz/ACL-2013.zip>.

#### 6 Acknowledgements

This work is partially supported by MCIT-TTS consortium project funded by MCIT, Government of India. The authors would also like to thank all the native speakers who participated in the perceptual evaluations.

#### References

- A.W. Black and K. Lenzo. 2004. Multilingual Text to Speech synthesis. In *Proceedings of ICASSP*, Montreal, Canada.
- N. Campbell. 1998. Foreign language speech synthesis. In *Proceedings ESCA/COCOSDA workshop on speech synthesis*, Jenolan Caves, Australia.
- N. Campbell. 2001. Talking foreign. Concatenative speech synthesis and the language barrier. In *Proceedings Eurospeech*, pages 337–340, Aalborg, Denmark.

- J. Latorre, K. Iwano, and S. Furui. 2006. New approach to polygot speech generation by means of an HMM based speaker adaptable synthesizer. *Speech Communication*, 48:1227–1242.
- V. Pagel, K. Lenzo, and A.W. Black. 1998. Letter to sound rules for accented lexicon compression. In *Proceedings of ICSLP 98*, volume 5, Sydney, Australia.
- Suzanne Romaine and Braj Kachru. 1992. *The Oxford Companion to the English Language*. Oxford University Press.
- C. Traber, K. Huber, K. Nedir, B. Pfister, E. Keller, and B. Zellner. 1999. From multilingual to polyglot speech synthesis. In *Proceedings of Eurospeech 99*, pages 835–838.
- H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A.W. Black, and K. Tokuda. 2007. The HMM-based speech synthesis system version 2.0. In *Proceedings of ISCA SSW6*, Bonn, Germany.



# Enriching Entity Translation Discovery using Selective Temporality

Gae-won You, Young-rok Cha, Jinhan Kim, and Seung-won Hwang

Pohang University of Science and Technology, Republic of Korea  
{gwyou, line0930, wlsqks08, swhwang}@postech.edu

## Abstract

This paper studies named entity translation and proposes “selective temporality” as a new feature, as using temporal features may be harmful for translating “atemporal” entities. Our key contribution is building an automatic classifier to distinguish temporal and atemporal entities then align them in separate procedures to boost translation accuracy by 6.1%.

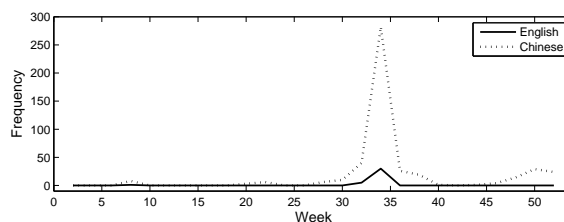
## 1 Introduction

Named entity translation discovery aims at mapping entity names for people, locations, *etc.* in source language into their corresponding names in target language. As many new named entities appear every day in newspapers and web sites, their translations are non-trivial yet essential.

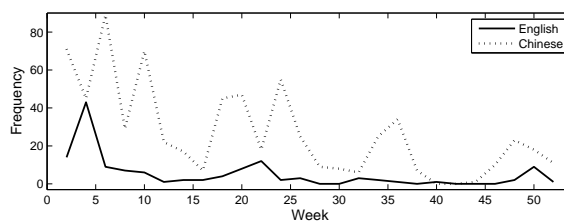
Early efforts of named entity translation have focused on using *phonetic feature* (called PH) to estimate a phonetic similarity between two names (Knight and Graehl, 1998; Li et al., 2004; Virga and Khudanpur, 2003). In contrast, some approaches have focused on using *context feature* (called CX) which compares surrounding words of entities (Fung and Yee, 1998; Diab and Finch, 2000; Laroche and Langlais, 2010).

Recently, holistic approaches combining such similarities have been studied (Shao and Ng, 2004; You et al., 2010; Kim et al., 2011). (Shao and Ng, 2004) rank translation candidates using PH and CX independently and return results with the highest average rank. (You et al., 2010) compute initial translation scores using PH and iteratively update the scores using *relationship feature* (called R). (Kim et al., 2011) boost You’s approach by additionally leveraging CX.

More recent approaches consider *temporal feature* (called T) of entities in two corpora (Klementiev and Roth, 2006; Tao et al., 2006; Sproat et



(a) Temporal entity: “Usain Bolt”



(b) Atemporal entity: “Hillary Clinton”

Figure 1: Illustration on temporality

al., 2006; Kim et al., 2012). T is computed using frequency vectors for entities and combined with PH (Klementiev and Roth, 2006; Tao et al., 2006). (Sproat et al., 2006) extend Tao’s approach by iteratively updating overall similarities using R. (Kim et al., 2012) holistically combine all the features: PH, CX, T, and R.

However, T used in previous approaches is a good feature only if temporal behaviors are “symmetric” across corpora. In contrast, Figure 1 illustrates asymmetry, by showing the frequencies of “Usain Bolt,” a Jamaican sprinter, and “Hillary Clinton,” an American politician, in comparable news articles during the year 2008. The former is mostly mentioned in the context of some temporal events, *e.g.*, Beijing Olympics, while the latter is not. In such case, as Hillary Clinton is a famous female leader, she may be associated with other Chinese female leaders in Chinese corpus, while such association is rarely observed in English corpus, which causes asymmetry. That is, Hillary Clinton is “atemporal,” as Figure 1(b) shows, such that using such dissimilarity against deciding this pair as a correct translation would be harmful. In clear contrast, for Usain Bolt, similarity of temporal dis-

tributions in Figure 1(a) is a good feature for concluding this pair as a correct one.

To overcome such problems, we propose a new notion of “selective temporality” (called this feature **ST** to distinguish from **T**) to automatically distinguish temporal and atemporal entities. Toward this goal, we design a classifier to distinguish temporal entities from atemporal entities, based on which we align temporal projections of entity graphs for the temporal ones and the entire entity graphs for the atemporal ones. We also propose a method to identify the optimal window size for temporal entities. We validate this “selective” use of temporal features boosts the accuracy by 6.1%.

## 2 Preliminaries

Our approach follows a graph alignment framework proposed in (You et al., 2010). Our graph alignment framework consists of 4 steps.

### 2.1 Step 1: Graph Construction

We first build a graph  $G = (V, E)$  from each language corpus, where  $V$  is a set of entities (nodes) and  $E$  is a set of co-occurrence relationships (unweighted edges) between entities. We consider entities occurring more than  $\eta$  times as nodes and entity pairs co-occurring more than  $\sigma$  times as edges.

To identify entities, we use a CRF-based named entity tagger (Finkel et al., 2005) and a Chinese word breaker (Gao et al., 2003) for English and Chinese corpora, respectively.

### 2.2 Step 2: Initialization

Given two graphs  $G_e = (V_e, E_e)$  and  $G_c = (V_c, E_c)$ , we initialize  $|V_e|$ -by- $|V_c|$  initial similarity matrix  $R^0$  using **PH** and **CX** for every pair  $(e, c)$  where  $e \in V_e$  and  $c \in V_c$ .

For **PH**, we use a variant of Edit-Distance (You et al., 2010) between English entity and a romanized representation of Chinese entity called Pinyin. For **CX**, the context similarity is computed based on entity context which is defined as a set of words near to the entity (we ignore some words such as stop words and other entities). We compute similarity of the most frequent 20 words for each entity using a variant of Jaccard index. To integrate two similarity scores, we adopt an average as a composite function.

We finally compute initial similarity scores for all pairs  $(e, c)$  where  $e \in V_e$  and  $c \in V_c$ , and build the initial similarity matrix  $R^0$ .

### 2.3 Step 3: Reinforcement

We reinforce  $R^0$  by leveraging **R** and obtain a converged matrix  $R^\infty$  using the following model:

$$R_{(i,j)}^{t+1} = \lambda R_{(i,j)}^0 + (1 - \lambda) \sum_{(u,v)_k \in B^t(i,j,\theta)} \frac{R_{(u,v)}^t}{2^k}$$

This model is a linear combination of (a) the initial similarity  $R_{(i,j)}^0$  of entity pair  $(i, j) \in V_e \times V_c$  and (b) the similarities  $R_{(u,v)}^t$  of their matched neighbors  $(u, v) \in V_e \times V_c$  where  $t$  indicates iteration,  $B^t(i, j, \theta)$  is an ordered set of the matched neighbors, and  $k$  is the rank of the matched neighbors.  $\lambda$  is the coefficient for balancing two terms.

However, as we cannot assure the correctly matched neighbors  $(u, v)$ , a chicken-and-egg dilemma, we take advantage of the current similarity  $R^t$  to estimate the next similarity  $R^{t+1}$ . Algorithm 1 describes the process of matching the neighbors where  $N(i)$  and  $N(j)$  are the sets of neighbor nodes of  $i \in V_e$  and  $j \in V_c$ , respectively, and  $H$  is a priority queue sorting the matched pairs in non-increasing order of similarities. To guarantee that the neighbors are correctly matched, we use only the matches such that  $R_{(u,v)}^t \geq \theta$ .

---

#### Algorithm 1 $B^t(i, j, \theta)$

---

```

1:  $M \leftarrow \{\}; H \leftarrow \{\}$ 
2:  $\forall u \in N(i), \forall v \in N(j)$   $H.\text{push}(u, v)$  such that
    $R_{(u,v)}^t \geq \theta$ 
3: while  $H$  is not empty do
4:    $(u, v) \leftarrow H.\text{pop}()$ 
5:   if neither  $u$  nor  $v$  are matched yet then
6:      $M \leftarrow M \cup \{(u, v)\}$ 
7:   end if
8: end while
9: return  $M$ 

```

---

### 2.4 Step 4: Extraction

From  $R^\infty$ , we finally extract one-to-one matches by using simple greedy approach of three steps: (1) choosing the pair with the highest similarity score; (2) removing the corresponding row and column from  $R^\infty$ ; (3) repeating (1) and (2) until the matching score is not less than a threshold  $\delta$ .

## 3 Entity Translation Discovery using Selective Temporality

**Overall Framework:** We propose our framework by putting together two separate procedures for temporal and atemporal entities to compute the overall similarity matrix  $R$

We first build two temporal graphs from the corpora within every time window, optimized in Section 3.1. We then compute the reinforced matrix  $R_s^\infty$  obtained from the window starting at the time-stamp  $s$ . To keep the best match scores among all windows, we update  $R$  using the best similarity among  $\forall s, R_s^\infty$ . We then extract the candidate translation pairs  $M_{ours}$  by running step 4.

As there can exist atemporal entities in  $M_{ours}$ , we classify them (Section 3.2). Specifically, we build two entire graphs and compute  $R^\infty$ . We then distinguish temporal entities from atemporal ones using our proposed metric for each matched pair  $(i, j) \in M_{ours}$  and, if the pair is atemporal,  $R_{(i,j)}$  is updated as the atemporal similarity  $R_{(i,j)}^\infty$ .

From the final matrix  $R$ , we extract the matched pairs by running step 4 with  $R$  once again.

### 3.1 Projecting Graph for Temporal Entities

We first project graphs temporally to improve translation quality for temporal entities. As the optimal projection would differ across entities, we generate many projected graphs by shifting time window over all periods, and then identify the best window for each entity.

The rest of this section describes how we set the right window size  $w$ . Though each entity may have its own optimal  $w$ , we find optimizing for each entity may negatively influence on considering relationships with entities of different window sizes. Thus, we instead find the optimal window size  $\hat{w}$  to maximize the global ‘‘symmetry’’ of the given two graphs.

We now define ‘‘symmetry’’ with respect to the truth translation pair  $M$ . We note it is infeasible to assume we have  $M$  during translation, and will later relax to consider how  $M$  can be approximated.

Given a set of graph pairs segmented by the shifted windows

$$\{(G_e^{(0,w)}, G_c^{(0,w)}), \dots, (G_e^{(s,s+w)}, G_c^{(s,s+w)}), (G_e^{(s+\Delta s, s+\Delta s+w)}, G_c^{(s+\Delta s, s+\Delta s+w)}), \dots\},$$

where  $s$  is the time-stamp, our goal is to find the window size  $\hat{w}$  maximizing the average symmetry  $S$  of graph pairs:

$$\hat{w} = \arg \max_{\forall w} \left( \frac{\sum_s S(G_e^{(s,s+w)}, G_c^{(s,s+w)}; M)}{N} \right)$$

Given  $M$ , symmetry  $S$  can be defined for (1) *node* and (2) *edge* respectively. We first define the

*node symmetry*  $S_n$  as follows:

$$S_n(G_e, G_c; M) = \frac{\sum_{(e,c) \in V_e \times V_c} I(e, c; M)}{\max\{|V_e|, |V_c|\}}$$

where  $I(u, v; M)$  to be 1 if  $(u, v) \in M$ , 0 otherwise. High node symmetry leads to accurate translation in  $R^0$  (Initialization step). Similarly, we define the *edge symmetry*  $S_e$  as follows:

$$S_e(G_e, G_c; M) = \frac{\sum_{(e_1, e_2) \in E_e} \sum_{(c_1, c_2) \in E_c} I(e_1, c_1; M) I(e_2, c_2; M)}{\max\{|E_e|, |E_c|\}}$$

In contrast, high edge symmetry leads to accurate translation in  $R^\infty$  (Reinforcement step).

We finally define the symmetry  $S$  as the weighted sum of  $S_n$  and  $S_e$  with parameter  $\alpha$  (empirically tuned to 0.8 in our experiment).

$$S(G_e, G_c; M) = \alpha S_n(G_e, G_c; M) + (1 - \alpha) S_e(G_e, G_c; M)$$

However, as it is infeasible to assume we have the truth translation pair  $M$ , we approximate  $M$  using intermediate translation results  $M_{ours}$  computed at step 4. To insert only true positive pairs in  $M_{ours}$ , we set threshold higher than the optimized value from the step 4. We found out that symmetry from  $M_{ours}$  closely estimates that from  $M$ :

$$S(G_e, G_c; M) \approx S(G_e, G_c; M_{ours})$$

Specifically, observe from Table 1 that, given a manually built ground-truth set  $M_g \subset M$  as described in Section 4.1,  $S(G_e, G_c; M_{ours})$  returns the best symmetry value in two weeks for person entities, which is expectedly the same as the result of  $S(G_e, G_c; M_g)$ . This suggests that we can use  $M_{ours}$  for optimizing window size.

| Weeks      | 26    | 13    | 4     | <b>2</b>     | 1     |
|------------|-------|-------|-------|--------------|-------|
| $M_g$      | .0264 | .0276 | .0303 | <b>.0318</b> | .0315 |
| $M_{ours}$ | .0077 | .0084 | .0102 | <b>.0113</b> | .0107 |

Table 1: Symmetry of window size

### 3.2 Building Classifier

We then classify temporal/atemporal entities. As a first step, we observe their characteristics: **Temporal entities** have peaks in the frequency distribution of both corpora and these peaks are aligned, while such distribution of **atemporal entities** are more uniform and less aligned.

Based on these observations, we identify the following criteria for temporal entities: (1) Their two distributions  $\mathbf{m}$  in English corpus and  $\mathbf{n}$  in Chinese corpus should have aligned peaks. (2) Frequencies at the peaks are the higher the better.

For the *first criterion*, we first normalize the two vectors  $\hat{\mathbf{m}}$  and  $\hat{\mathbf{n}}$  since two corpora have different scales, *i.e.*, different number of documents. We then calculate the inner product of the two vectors  $\mathbf{x} = \langle \hat{\mathbf{m}}, \hat{\mathbf{n}} \rangle$ , such that this aggregated distribution  $\mathbf{x}$  peaks, only if both  $\hat{\mathbf{m}}$  and  $\hat{\mathbf{n}}$  peak at the same time.

For the *second criterion*, we have a spectrum of option from taking the frequencies at all peaks in one extreme, to taking only the maximum frequency in another extreme. A metric representing such a spectrum is  $p$ -norm, which represents sum when  $p = 1$  and maximum when  $p = \infty$ . We empirically tune the right balance to distinguish temporal and atemporal entities, which turns out to be  $p = 2.2$ .

Overall, we define a metric  $d(\mathbf{m}, \mathbf{n})$  which satisfies both criteria as follow:

$$d(\mathbf{m}, \mathbf{n}) = \left( \sum_{i=1}^n (\hat{\mathbf{m}}_i \hat{\mathbf{n}}_i)^p \right)^{\frac{1}{p}}$$

For instance, this measure returns 0.50 and 0.03 for the distributions in Figure 1(a) and (b), respectively, from which we can determine the translation of Figure 1(a) is temporal and the one of Figure 1(b) is atemporal.

## 4 Experimental Evaluation

### 4.1 Experimental Settings

We obtained comparable corpora from English and Chinese Gigaword Corpora (LDC2009T13 and LDC2009T27) published by the Xinhua News Agency during the year 2008. From them, we extracted person entities and built two graphs,  $G_e = (V_e, E_e)$  and  $G_c = (V_c, E_c)$  by setting  $\eta = 20$  which was used in (Kim et al., 2011).

Next, we built a ground truth translation pair set  $M_g$  for person entities. We first selected 500 person names randomly from English corpus. We then hired a Chinese annotator to translate them into their Chinese names. Among them, only 201 person names were matched to our Chinese corpus. We used all such pairs to identify the best parameters and compute the evaluation measures.

We implemented and compared the following approaches denoted as the naming convention of listing of the used features in a parenthesis ():

- (PH+R) in (You et al., 2010).
- (PH+CX+R) in (Kim et al., 2011).
- (PH+CX+R+T) in (Kim et al., 2012).
- (PH+CX+R+ST): This is our approach.

We evaluated the effectiveness of our new approach using four measures: MRR, precision, recall, and F1-score, where MRR (Voorhees, 2001) is the average of the reciprocal ranks of the query results defined as follows:

$$\text{MRR} = \frac{1}{|Q|} \sum_{(u,v) \in Q} \frac{1}{\text{rank}_{(u,v)}},$$

where  $Q$  is a set of ground-truth matched pairs  $(u, v)$  such that  $u \in V_e$  and  $v \in V_c$ , and  $\text{rank}_{(u,v)}$  is the rank of  $R_{(u,v)}$  among all  $R_{(u,w)}$ 's such that  $w \in V_c$ . We performed a 5-fold cross validation by dividing ground truth into five groups. We used four groups to training the parameters to maximize F1-scores, used the remaining group as a test-set using trained parameters, and computed average of five results. (**bold numbers** indicate the best performance for each metric.)

### 4.2 Experimental Results

#### Effect of window size

We first validated the effectiveness of our approach for various window sizes (Table 2). Observe that it shows the best performance in two weeks for MRR and F1 measures. Interestingly, this result also corresponds to our optimization result  $\hat{w}$  of Table 1 in Section 3.1.

| Weeks     | 26    | 13    | 4            | <b>2</b>     | 1            |
|-----------|-------|-------|--------------|--------------|--------------|
| MRR       | .7436 | .8066 | .8166        | <b>.8233</b> | .8148        |
| Precision | .7778 | .7486 | .8126        | .8306        | <b>.8333</b> |
| Recall    | .6617 | .6875 | <b>.7320</b> | .7295        | .7214        |
| F1        | .7151 | .7165 | .7701        | <b>.7765</b> | .7733        |

Table 2: Optimality of window size

#### Overall performance

Table 3 shows the results of four measures. Observe that (PH+CX+R+T) and (PH+CX+R+ST) outperform the others in all our settings. We can also observe the effect of selective temporality, which maximizes the symmetry between two graphs as shown in Table 1, *i.e.*, (PH+CX+R+ST)

| Method       | MRR          | Precision    | Recall       | F1           |
|--------------|--------------|--------------|--------------|--------------|
| (PH+R)       | .6500        | .7230        | .4548        | .5552        |
| (PH+CX+R)    | .7499        | .7704        | .6623        | .7120        |
| (PH+CX+R+T)  | .7658        | .8223        | .6608        | .7321        |
| (PH+CX+R+ST) | <b>.8233</b> | <b>.8306</b> | <b>.7295</b> | <b>.7765</b> |

Table 3: MRR, Precision, Recall, and F1-score

| English Name | TL+CX+R | TL+CX+R+T | TL+CX+R+ST |
|--------------|---------|-----------|------------|
| Hu Jintao    | 胡锦涛     | 胡锦涛       | 胡锦涛        |
| Kim Yong Nam | 殷永建     | 金永南       | 金永南        |
| Karzai       | 盖茨      | 拉克        | 卡尔扎伊       |

Figure 2: The translation examples where shaded cells indicate the correctly translated pairs.

outperforms (PH+CX+R+T) by 6.1%. These improvements were statistically significant according to the Student’s t-test at  $P < 0.05$  level.

Figure 2 shows representative translation examples. All approaches found famous entities such as “Hu Jintao,” a former leader of China, but (PH+CX+R) failed to find translation of lesser known entities, such as “Kim Yong Nam.” Using temporal features help both (PH+CX+R+T) and (PH+CX+R+ST) identify the right translation, as Kim’s temporal occurrence is strong and symmetric in both corpora. In contrast, (PH+CX+R+T) failed to find the translation of “Karzai”, the president of Afghanistan, as it only appears weakly and transiently during a short period time, for which only (PH+CX+R+ST) applying varying sizes of window per entity is effective.

## 5 Conclusion

This paper validated that considering temporality selectively is helpful for improving the translation quality. We developed a classifier to distinguish temporal/atemporal entities and our proposed method outperforms the state-of-the-art approach by 6.1%.

## Acknowledgment

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea and Microsoft Research, under IT/SW Creative research program supervised by the NIPA (National IT Industry Promotion Agency). (NIPA-2012- H0503-12-1036).

## References

- Mona Diab and Steve Finch. 2000. A statistical word level translation model for comparable corpora. In *RIAO '00*.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL*.
- Pascale Fung and Lo Yuen Yee. 1998. An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In *COLING*.
- Jianfeng Gao, Mu Li, and Chang-Ning Huang. 2003. Improved Source-channel Models for Chinese Word Segmentation. In *ACL*.
- Jinhan Kim, Long Jiang, Seung-won Hwang, Young-In Song, and Ming Zhou. 2011. Mining Entity Translations from Comparable Corpora: A Holistic Graph Mapping Approach. In *CIKM*.
- Jinhan Kim, Seung won Hwang, Long Jiang, Young-In Song, and Ming Zhou. 2012. Entity Translation Mining from Comparable Corpora: Combining Graph Mapping with Corpus Latent Features. *IEEE TKDE*.
- Alexandre Klementiev and Dan Roth. 2006. Named entity transliteration and discovery from multilingual comparable corpora. In *HLT-NAACL '06*.
- Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics*.
- Audrey Laroche and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *COLING*.
- Haizhou Li, Zhang Min, and Su Jian. 2004. A Joint Source-Channel Model for Machine Transliteration. In *ACL*.
- Li Shao and Hwee Tou Ng. 2004. Mining New Word Translations from Comparable Corpora. In *COLING*.
- Richard Sproat, Tao Tao, and ChengXiang Zhai. 2006. Named Entity Transliteration with Comparable Corpora. In *ACL*.
- Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. Unsupervised Named Entity Transliteration Using Temporal and Phonetic Correlation. In *EMNLP*.
- Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-language applications. In *SIGIR '03*.
- Ellen M. Voorhees. 2001. The TREC Question Answering Track. *Natural Language Engineering*, 7(4):361–378.
- Gae-won You, Seung-won Hwang, Young-In Song, Long Jiang, and Zaiqing Nie. 2010. Mining Name Translations from Entity Graph Mapping. In *Proceedings of EMNLP*, pages 430–439.

# Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling

Heike Adel

Ngoc Thang Vu

Tanja Schultz

Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT)

heike.adel@student.kit.edu

thang.vu@kit.edu

tanja.schultz@kit.edu

## Abstract

In this paper, we investigate the application of recurrent neural network language models (RNNLM) and factored language models (FLM) to the task of language modeling for Code-Switching speech. We present a way to integrate part-of-speech tags (POS) and language information (LID) into these models which leads to significant improvements in terms of perplexity. Furthermore, a comparison between RNNLMs and FLMs and a detailed analysis of perplexities on the different backoff levels are performed. Finally, we show that recurrent neural networks and factored language models can be combined using linear interpolation to achieve the best performance. The final combined language model provides 37.8% relative improvement in terms of perplexity on the SEAME development set and a relative improvement of 32.7% on the evaluation set compared to the traditional n-gram language model.

**Index Terms:** multilingual speech processing, code switching, language modeling, recurrent neural networks, factored language models

## 1 Introduction

Code-Switching (CS) speech is defined as speech that contains more than one language ('code'). It is a common phenomenon in multilingual communities (Auer, 1999a). For the automated processing of spoken communication in these scenarios, a speech recognition system must be able to handle code switches. However, the components of speech recognition systems are usually trained on monolingual data. Furthermore, there is a lack of bilingual training data. While there

have been promising research results in the area of acoustic modeling, only few approaches so far address Code-Switching in the language model. Recently, it has been shown that recurrent neural network language models (RNNLMs) can improve perplexity and error rates in speech recognition systems in comparison to traditional n-gram approaches (Mikolov et al., 2010; Mikolov et al., 2011). One reason for that is their ability to handle longer contexts. Furthermore, the integration of additional features as input is rather straightforward due to their structure. On the other hand, factored language models (FLMs) have been used successfully for languages with rich morphology due to their ability to process syntactical features, such as word stems or part-of-speech tags (Bilmes and Kirchhoff, 2003; El-Desoky et al., 2010). The main contribution of this paper is the application of RNNLMs and FLMs to the challenging task of Code-Switching. Furthermore, the two different models are combined using linear interpolation. In addition, a comparison between them is provided including a detailed analysis to explain their results.

## 2 Related Work

For this work, three different topics are investigated and combined: linguistic investigation of Code-Switching, recurrent neural network language modeling and factored language models. In (Muysken, 2000; Poplack, 1978; Bokamba, 1989), it is observed that code switches occur at positions in an utterance where they do not violate the syntactical rules of the involved languages. On the one hand, Code-Switching can be regarded as a speaker dependent phenomenon (Auer, 1999b; Vu, Adel et al., 2013). On the other hand, particular Code-Switching patterns are shared across speakers (Poplack, 1980). It can be observed that part-of-speech tags may predict Code-Switching points more reliable than words themselves. The

authors of (Solorio et al., 2008a) predict Code-Switching points using several linguistic features, such as word form, language ID, part-of-speech tags or the position of the word relative to the phrase (BIO). The best result is obtained by combining those features. In (Chan et al., 2006), four different kinds of n-gram language models are compared to predict Code-Switching. It is discovered that clustering all foreign words into their part-of-speech classes leads to the best performance.

In the last years, neural networks have been used for a variety of tasks, including language modeling (Mikolov et al., 2010). Recurrent neural networks are able to handle long-term contexts since the input vector does not only contain the current word but also the previous hidden layer. It is shown that these networks outperform traditional language models, such as n-grams which only contain very limited histories. In (Mikolov et al., 2011), the network is extended by factorizing the output layer into classes to accelerate the training and testing processes. The input layer can be augmented to model features, such as part-of-speech tags (Shi et al., 2011; Adel, Vu et al., 2013). In (Adel, Vu et al., 2013), recurrent neural networks are applied to Code-Switching speech. It is shown that the integration of POS tags into the neural network, which predicts the next language as well as the next word, leads to significant perplexity reductions.

A factored language model refers to a word as a vector of features, such as the word itself, morphological classes, POS tags or word stems. Hence, it provides another possibility to integrate syntactical features into the language modeling process. In (Bilmes and Kirchhoff, 2003), it is shown that factored language models are able to outperform standard n-gram techniques in terms of perplexity. In the same paper, generalized parallel backoff is introduced. This technique can be used to generalize traditional backoff methods and to improve the performance of factored language models. Due to the integration of various features, it is possible to handle rich morphology in languages like Arabic or Turkish (Duh and Kirchhoff, 2004; El-Desoky et al., 2010).

### 3 Code-Switching Language Modeling

#### 3.1 Motivation

Since there is a lack of Code-Switching data, language modeling is a challenging task. Traditional n-gram approaches may not provide reliable estimates. Hence, more general features than words should be integrated into the language models. Therefore, we apply recurrent neural networks and factored language models. As features, we use part-of-speech tags and language identifiers.

#### 3.2 Using Recurrent Neural Networks As Language Model

This section describes the structure of the recurrent neural network (RNNLM) that we use as Code-Switching language model. It has been proposed in (Adel, Vu et al., 2013) and is illustrated in figure 1.

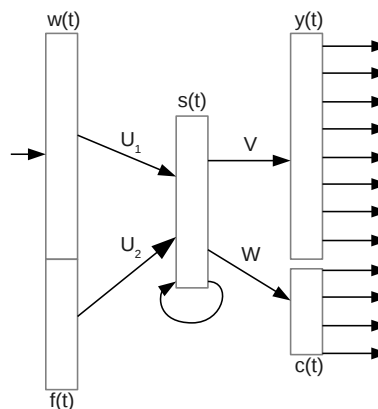


Figure 1: RNNLM for Code-Switching (based upon a figure in (Mikolov et al., 2011))

Vector  $w(t)$ , which represents the current word using 1-of-N coding, forms the input of the recurrent neural network. Thus, its dimension equals the size of the vocabulary. Vector  $s(t)$  contains the state of the network and is called 'hidden layer'. The network is trained using back-propagation through time (BPTT), an extension of the back-propagation algorithm for recurrent neural networks. With BPTT, the error is propagated through recurrent connections back in time for a specific number of time steps  $t$ . Hence, the network is able to remember information for several time steps. The matrices  $U_1$ ,  $U_2$ ,  $V$ , and  $W$  contain the weights for the connections between the layers. These weights are learned during the training phase. Moreover, the output layer is factorized

into classes which provide language information. In this work, four classes are used: English, Mandarin, other languages and particles. Vector  $c(t)$  contains the probabilities for each class and vector  $y(t)$  provides the probabilities for each word given its class. Hence, the probability  $P(w_i|history)$  is computed as shown in equation 1.

$$P(w_i|history) = P(c_i|s(t))P(w_i|c_i, s(t)) \quad (1)$$

It is intended to not only predict the next word but also the next language. Hence according to equation 1, the probability of the next language is computed first and then the probability of each word given the language. Furthermore, a vector  $f(t)$  is added to the input layer. It provides features (in this work part-of-speech tags) corresponding to the current word. Thus, not only the current word is activated but also its features. Since the POS tags are integrated into the input layer, they are also propagated into the hidden layer and back-propagated into its history  $s(t)$ . Hence, not only the previous feature is stored in the history but also features from several time steps in the past.

### 3.3 Using Factored Language Models

Factored language models (FLM) are another approach to integrate syntactical features, such as part-of-speech tags or language identifiers into the language modeling process. Each word is regarded as a sequence of features which are used for the computation of the n-gram probabilities. If a particular sequence of features has not been detected in the training data, backoff techniques will be applied. For our task of Code-Switching, we develop two different models: One model with only part-of-speech tags as features and one model including also language information tags. Unfortunately, the number of possible parameters is rather high: Different feature combinations from different time steps can be used to predict the next word (conditioning factors), different back-off paths and different smoothing methods may be applied. To detect useful parameters, the genetic algorithm described in (Duh and Kirchhoff, 2004) is used. It is an evolution-inspired technique that encodes the parameters of an FLM as binary strings (genes). First, an initializing set of genes is generated. Then, a loop follows that evaluates the fitness of the genes and mutates them until their average fitness is not improved any more. As fitness value, the inverse perplexity of the FLM corresponding to the gene on the development set is

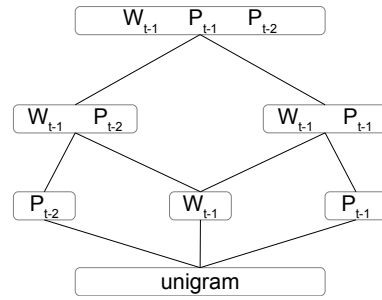


Figure 2: Backoff graph of the FLM

used. Hence, parameter solutions with lower perplexities are preferred in the selection of the genes for the following iteration. In (Duh and Kirchhoff, 2004), it is shown that this genetic method outperforms both knowledge-based and randomized choices. For the case of part-of-speech tags as features, the method results in three conditioning factors: the previous word  $W_{t-1}$  and the two previous POS tags  $P_{t-1}$  and  $P_{t-2}$ . The backoff graph obtained by the algorithm is illustrated in figure 2. According to the result of the genetic algorithm, different smoothing methods are used at different backoff levels: For the backoff from three factors to two factors, Kneser-Ney discounting is applied. If the probabilities for the factor combination  $W_{t-1}P_{t-2}$  could not be estimated reliably, absolute discounting is used. In all other cases, Witten-Bell discounting is applied. An overview of the different smoothing methods can be found in (Rosenfeld, 2000).

## 4 Experiments and Results

### 4.1 Data Corpus

SEAME (South East Asia Mandarin-English) is a conversational Mandarin-English Code-Switching speech corpus recorded from Singaporean and Malaysian speakers (D.C. Lyu et al., 2011). It was used for the research project 'Code-Switch' jointly performed by Nanyang Technological University (NTU) and Karlsruhe Institute of Technology (KIT). The recordings consist of spontaneously spoken interviews and conversations of about 63 hours of audio data. For this task, we deleted all hesitations and divided the transcribed words into four categories: English words, Mandarin words, particles (Singaporean and Malaysian discourse particles) and others (other languages). These categories are used as language information in the language models. The average number of Code-Switching points between Mandarin and English



is 2.6 per utterance and the duration of monolingual segments is quite short: The average duration of English and Mandarin segments is only 0.67 seconds and 0.81 seconds respectively. In total, the corpus contains 9,210 unique English and 7,471 unique Mandarin vocabulary words. We divided the corpus into three disjoint sets (training, development and test set) and assigned the data based on several criteria (gender, speaking style, ratio of Singaporean and Malaysian speakers, ratio of the four categories, and the duration in each set). Table 1 lists the statistics of the corpus in these sets.

|               | Train set | Dev set | Eval set |
|---------------|-----------|---------|----------|
| # Speakers    | 139       | 8       | 8        |
| Duration(hrs) | 59.2      | 2.1     | 1.5      |
| # Utterances  | 48,040    | 1,943   | 1,018    |
| # Token       | 525,168   | 23,776  | 11,294   |

Table 1: Statistics of the SEAME corpus

## 4.2 POS Tagger for Code-Switching Speech

To be able to assign part-of-speech tags to our bilingual text corpus, we apply the POS tagger described in (Schultz et al., 2010) and (Adel, Vu et al., 2013). It consists of two different monolingual (Stanford log-linear) taggers (Toutanova et al., 2003; Toutanova et al., 2000) and a combination of their results. While (Solorio et al., 2008b) passes the whole Code-Switching text to both monolingual taggers and combines their results using different heuristics, in this work, the text is splitted into different languages first. The tagging process is illustrated in figure 3.

Mandarin is determined as matrix language (the main language of an utterance) and English as embedded language. If three or more words of the embedded language are detected, they are passed to the English tagger. The rest of the text is passed to the Mandarin tagger, even if it contains foreign words. The idea is to provide the tagger as much context as possible. Since most English words in the Mandarin segments are falsely tagged as nouns by the Mandarin tagger, a postprocessing step is applied. It passes all foreign words of the Mandarin segments to the English tagger in order to replace the wrong tags with the correct ones.

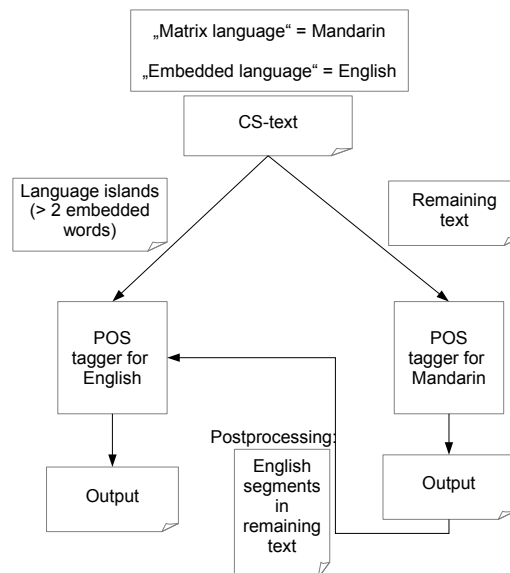


Figure 3: Tagging of Code-Switching speech

## 4.3 Evaluation

For evaluation, we compute the perplexity of each language model on the SEAME development and evaluation set and perform an analysis of the different back-off levels to understand in detail the behavior of each language model. A traditional 3-gram LM trained with the SEAME transcriptions serves as baseline.

### 4.3.1 LM Performance

The language models are evaluated in terms of perplexity. Table 2 presents the results on the development and test set.

| Model             | dev set       | test set      |
|-------------------|---------------|---------------|
| Baseline 3-gram   | 285.87        | 285.25        |
| FLM (pos)         | <b>263.57</b> | <b>271.57</b> |
| FLM (pos + lid)   | 263.84        | 276.99        |
| RNNLM (pos)       | 233.50        | 268.05        |
| RNNLM (pos + lid) | <b>219.85</b> | <b>239.21</b> |

Table 2: Perplexity results

It can be noticed that both the RNNLM and the FLM model outperform the traditional 3-gram model. Hence, adding syntactical features improves the word prediction. For the FLM, it leads to no improvement to add the language identifier as feature. In contrast, clustering the words into their languages on the output layer of the RNNLM leads to lower perplexities.

### 4.3.2 Backoff Level Analysis

To understand the different results of the RNNLM and the FLM, an analysis similar to the one described in (Oparin et al., 2012) is performed. For each word, the backoff-level of the n-gram model is observed. Then, a level-dependent perplexity is computed for each model as shown in equation 2.

$$PPL_k = 10^{-\frac{1}{N_k} \sum_{w_k} \log_{10} P(w_k|h_k)} \quad (2)$$

In the equation,  $k$  denotes the backoff-level,  $N_k$  the number of words on this level,  $w_k$  the current word and  $h_k$  its history. Table 3 shows how often each backoff-level is used and presents the level-dependent perplexities of each model on the development set.

|                 | 1-gram          | 2-gram        | 3-gram       |
|-----------------|-----------------|---------------|--------------|
| # occurrences   | 6894            | 11628         | 6226         |
| Baseline 3-gram | 5,786.24        | 165.82        | 28.28        |
| FLM (pos)       | 4,950.31        | <b>147.70</b> | 30.99        |
| RNNLM           | <b>3,231.02</b> | 151.67        | <b>21.24</b> |

Table 3: Backoff-level-dependent PPLs

In case of backoff to the 2-gram, the FLM provides the best perplexity, while for the 3-gram and backoff to the 1-gram, the RNNLM performs best. This may be correlated with the better over-all perplexity of the RNNLM in comparison to the FLM. Nevertheless, the backoff to the 2-gram is used about twice as often as the backoff to the 1-gram or the 3-gram.

### 4.4 LM Interpolation

The different results of RNNLM and FLM show that they provide different estimates of the next word. Thus, a combination of them may reduce the perplexities of table 2. Hence, we apply linear interpolation to the probabilities of each two models as shown in equation 3.

$$P(w|h) = \lambda \cdot P_{M1}(w|h) + (1-\lambda) \cdot P_{M2}(w|h) \quad (3)$$

The equation shows the computation of the probability for word  $w$  given its history  $h$ .  $P_{M1}$  denotes the probability provided by the first model and  $P_{M2}$  the probability from the second model. Table 4 shows the results of this experiment. The weights are optimized on the development set. The interpolation of RNNLM and FLM leads to the best results. This may be caused by the superior backoff-level-dependent PPLs in comparison

| Model          | weight   | PPL on dev    | PPL on eval   |
|----------------|----------|---------------|---------------|
| FLM + 3-gram   | 0.7, 0.3 | 211.13        | 227.57        |
| RNNLM + 3-gram | 0.8, 0.2 | 206.49        | 227.08        |
| RNNLM + FLM    | 0.6, 0.4 | <b>177.79</b> | <b>192.08</b> |

Table 4: Perplexities after interpolation

to the 3-gram model. While the RNNLM performs better for the 3-gram and for the backoff to the 1-gram, the FLM performs the best in case of backoff to the 2-gram which is used more often than the other levels (table 3).

## 5 Conclusions

In this paper, we presented two different methods for language modeling of Code-Switching speech: Recurrent neural networks and factored language models. We integrated part-of-speech tags and language information to improve the performance of the language models. In addition, we analyzed their behavior on the different backoff levels. While the FLM performed better in case of backoff to the 2-gram, the RNNLM led to a better over-all performance. Finally, the models were combined using linear interpolation. The combined language model provided 37.8% relative improvement in terms of perplexity on the SEAME development set and a relative improvement of 32.7% on the evaluation set compared to the traditional n-gram LM.

## References

- H. Adel, N.T. Vu, F. Kraus, T. Schlippe, and T. Schultz. 2013 *Recurrent Neural Network Language Modeling for Code Switching Conversational Speech* In: Proceedings of ICASSP 2013.
- P. Auer 1999 *Code-Switching in Conversation*, Routledge.
- P. Auer 1999 *From codeswitching via language mixing to fused lects toward a dynamic typology of bilingual speech* In: International Journal of Bilingualism, vol. 3, no. 4, pp. 309-332.
- J.A. Bilmes and K. Kirchhoff. 2003 *Factored Language Models and Generalized Parallel Backoff* In: Proceedings of NAACL, 2003.
- E.G. Bokamba 1989 *Are there syntactic constraints on code-mixing?* In: World Englishes, vol. 8, no. 3, pp. 277-292.
- J.Y.C. Chan, PC Ching, T. Lee, and H. Cao 2006 *Automatic speech recognition of Cantonese-English*

- code-mixing utterances* In: Proceeding of Interspeech 2006.
- K. Duh and K. Kirchhoff. 2004. *Automatic Learning of Language Model Structure*, pg 148. In: Proceedings of the 20th international conference on Computational Linguistics.
- A. El-Desoky, R. Schlüter, H.Ney 2010 *A Hybrid Morphologically Decomposed Factored Language Models for Arabic LVCSR* In: NAACL 2010.
- D.C. Lyu, T.P. Tan, E.S. Cheng, H. Li 2011 *An Analysis of Mandarin-English Code-Switching Speech Corpus: SEAME* In: Proceedings of Interspeech 2011.
- M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. 1993 *Building a large annotated corpus of english: The penn treebank* In: Computational Linguistics, vol. 19, no. 2, pp. 313330.
- T. Mikolov, M. Karafiat, L. Burget, J. Jernocky and S. Khudanpur. 2010 *Recurrent Neural Network based Language Model* In: Proceedings of Interspeech 2010.
- T. Mikolov, S. Kombrink, L. Burget, J. Jernocky and S. Khudanpur. 2011 *Extensions of Recurrent Neural Network Language Model* In: Proceedings of ICASSP 2011.
- P. Muysken 2000 *Bilingual speech: A typology of code-mixing* In: Cambridge University Press, vol. 11.
- I. Oparin, M. Sundermeyer, H. Ney, J.-L. Gauvain 2012 *Performance analysis of Neural Networks in combination with n-gram language models* In: ICASSP, 2012.
- S. Poplack 1978 *Syntactic structure and social function of code-switching* , Centro de Estudios Puertorriquenos, City University of New York.
- S. Poplack 1980 *Sometimes ill start a sentence in spanish y termino en espanol: toward a typology of code-switching* In: Linguistics, vol. 18, no. 7-8, pp. 581-618.
- R. Rosenfeld 2000 *Two decades of statistical language modeling: Where do we go from here?* In: Proceedings of the IEEE 88.8 (2000): 1270-1278.
- T. Schultz, P. Fung, and C. Burgmer, 2010 *Detecting code-switch events based on textual features*.
- Y. Shi, P. Wiggers, M. Jonker 2011 *Towards Recurrent Neural Network Language Model with Linguistics and Contextual Features* In: Proceedings of Interspeech 2011.
- T. Solorio, Y. Liu 2008 *Part-of-speech tagging for English-Spanish code-switched text* In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.
- T. Solorio, Y. Liu 2008 *Learning to predict code-switching points* In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.
- K. Toutanova, C.D. Manning 2000 *Enriching the knowledge sources used in a maximum entropy part-of-speech tagger* In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, vol. 13.
- K. Toutanova, D. Klein, C.D. Manning, and Y. Singer 2003 *Feature-rich part-of-speech tagging with a cyclic dependency network* In: Proceedings of NAACL 2003.
- N.T. Vu, D.C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.S. Chng, T. Schultz, H. Li 2012 *A First Speech Recognition System For Mandarin-English Code-Switch Conversational Speech* In: Proceedings of Interspeech 2012.
- N.T. Vu, H. Adel, T. Schultz 2013 *An Investigation of Code-Switching Attitude Dependent Language Modeling* In: In Statistical Language and Speech Processing, First International Conference, 2013.
- N. Xue, F. Xia, F.D. Chiou, and M. Palmer 2005 *The penn chinese treebank: Phrase structure annotation of a large corpus* In: Natural Language Engineering, vol. 11, no. 2, pp. 207.

# Latent Semantic Matching: Application to Cross-language Text Categorization without Alignment Information

Tsutomu Hirao and Tomoharu Iwata and Masaaki Nagata

NTT Communication Science Laboratories, NTT Corporation

2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan

{hirao.tsutomu, iwata.tomoharu, nagata.masaaki}@lab.ntt.co.jp

## Abstract

Unsupervised object matching (UOM) is a promising approach to cross-language natural language processing such as bilingual lexicon acquisition, parallel corpus construction, and cross-language text categorization, because it does not require labor-intensive linguistic resources. However, UOM only finds one-to-one correspondences from data sets with the same number of instances in source and target domains, and this prevents us from applying UOM to real-world cross-language natural language processing tasks. To alleviate these limitations, we propose *latent semantic matching*, which embeds objects in both source and target language domains into a shared latent topic space. We demonstrate the effectiveness of our method on cross-language text categorization. The results show that our method outperforms conventional unsupervised object matching methods.

## 1 Introduction

Unsupervised object matching is a method for finding one-to-one correspondences between objects across different domains without knowledge about the relation between the domains. Kernelized sorting (Novi et al., 2010) and canonical correlation analysis based methods (Haghighi et al., 2008; Tripathi et al., 2010) are two such examples of unsupervised object matching, which have been shown to be quite useful for cross-language natural language processing (NLP) tasks. One of the most important properties of the unsupervised object matching is that it does not require any linguistic resources which connects between the languages. This distinguishes it from other cross-language NLP methods such as machine transla-

tion based and projection based approaches (Dumas et al., 1996; Gliozzo and Strapparava, 2005; Platt et al., 2010), which we need bilingual dictionaries or parallel sentences.

When we apply unsupervised object matching methods to cross-language NLP tasks, there are two critical problems. The first is that they only find one-to-one matching. The second is they require the same size of source- and target-data. For example, the correct translation of a word is not always unique. French words ‘*maison*’, ‘*appartement*’ and ‘*domicile*’ can be regarded as translation of an English word ‘home’. In addition, English vocabulary size is not equal to that of French.

These discussions motivate us to introduce a shared space in which both source and target domain objects will reside. If we can obtain such a shared space, we can match objects within the space, because we can use standard distance metrics on this space. This will also enable us to use various kinds of non-strict matching. For example,  $k$ -nearest objects in the source domain will be retrieved for a query object in the target domain. In this paper, we propose a simple but effective method to find the shared space by assuming that two languages have common latent topics, which we call *latent semantic matching*. With latent semantic matching, we first find latent topics in two domains independently. Then, the topics in two domains are aligned by kernelized sorting, and objects are embedded in a shared latent topic space. Latent topic representations are successfully used in a wide range of NLP tasks, such as information retrieval and text classification, because they represent intrinsic information of documents (Deerwester et al., 1990). By matching latent topics, we can find relation between source and target domains, and additionally we can handle different numbers of objects in two domains.

We compared latent semantic matching with conventional unsupervised object matching meth-

ods on the task of cross-language text categorization, *i.e.* classifying target side unlabeled documents by label information obtained from source side documents. The results show that, with more source side documents, our method achieved the highest classification accuracy.

## 2 Related work

Many cross-language text processing methods have been proposed that require correspondences between source and target languages. For example, (Dumais et al., 1996) proposed cross-lingual latent semantic indexing, and (Platt et al., 2010) employed oriented principle component analysis and canonical correlation analysis (CCA). They concatenate the document pairs (source document and its translation) obtained from a document-level parallel corpus. They then apply multivariate analysis to acquire the translational projection. There are extensions of latent Dirichlet allocation (LDA) (Blei et al., 2003) for cross-language analysis, such as multilingual topic models (Boyd-Graber and Blei, 2009), joint LDA (Jagadeesh and Daume III, 2010) and multilingual LDA (Xiao-chuan et al., 2011). They require a bilingual dictionary or document-level parallel corpora.

Unsupervised object matching methods have been proposed recently (Novi et al., 2010; Haghighi et al., 2008; Yamada and Sugiyama, 2011). These methods are promising in terms of language portability because they do not require external language resources. (Novi et al., 2010) proposed kernelized sorting (KS); it finds one-to-one correspondences between objects in different domains by permuting a set to maximize the dependence between two sets. Here, the Hilbert-Schmidt independence criterion is used for measuring dependence. (Djuric et al., 2012) proposed convex kernelized sorting as an extension of KS. (Yamada and Sugiyama, 2011) proposed least-squares object matching which maximizes the squared-loss mutual information between matched pairs. (Haghighi et al., 2008) proposed another framework, matching CCA (MCCA), based on a probabilistic interpretation of CCA (Bach and Jordan, 2005). MCCA simultaneously finds latent variables that represent correspondences and latent features so that the latent features of corresponding examples exhibit the maximum correlation. However, these unsupervised object matching methods have limitations. They require that

the source and target domains have the same data size, and they find one-to-one correspondences. There are critical weaknesses of these methods when we attempt to apply them to real world cross-language NLP applications.

## 3 Latent Semantic Matching

We propose latent semantic matching to find a shared latent space by assuming that two languages have common latent topics. Our method consists of following four steps: (1) for both source and target domains, we map the documents to a  $K$ -dimensional latent topic space independently, (2) we find the one-to-one correspondences between topics across source and target domains by unsupervised object matching, (3) we permute topics of the target side according to the correspondences, while fixing the topics of the source side, and (4) finally, we map documents in the source and target domains to a shared latent space by using permuted and fixed topics.

### 3.1 Topic Extraction as Dimension Reduction

Suppose that we have  $N$  documents in the source domain.  $\mathbf{s}_n = (s_{ni})_{i=1}^I$  is the  $n$ th document represented as a multi-dimensional column vector in the domain, *i.e.* each document is represented as a bag-of-words vector. Here, each element of the vectors indicates the TF-IDF score of the corresponding word in the document.  $I$  is the size of the feature set, *i.e.*, the vocabulary size in the source domain. Also, we have  $M$  documents in the target domain.  $\mathbf{t}_m = (t_{mj})_{j=1}^J$  is the  $m$ th document represented as a multi-dimensional vector.  $J$  is the vocabulary size in the target domain. Thus, the data set in the source domain is represented by an  $I \times N$  matrix,  $\mathbf{S} = (\mathbf{s}_1, \dots, \mathbf{s}_N)$ , the data set in the target is represented by a  $J \times M$  matrix,  $\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_M)$ .

We factorize these matrices using nonnegative matrix factorization (Lee and Seung, 2000) to find topics as follows:

$$\mathbf{S} \approx \mathbf{W}_S \mathbf{H}_S, \quad (1)$$

$$\mathbf{T} \approx \mathbf{W}_T \mathbf{H}_T. \quad (2)$$

$\mathbf{W}_S$  is an  $I \times K$  matrix that represents a set of topics, *i.e.* each column vector denotes word weights for each topic.  $\mathbf{H}_S$  is a  $K \times N$  matrix that denotes a set of latent semantic representations of documents in the source domain, *i.e.* each row

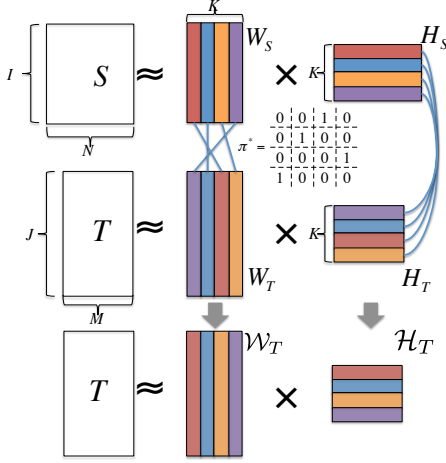


Figure 1: Topic alignments.

vector denotes an embedding of a document in the  $K$ -dimensional latent space. Similarly,  $\mathbf{W}_T$  is an  $I \times K$  matrix that represents a set of topics in the target domain, and  $\mathbf{H}_T$  is a  $K \times M$  matrix that denotes a set of latent semantic representations of target documents.  $K$  is less than  $I$  and  $J$ .

By factorizing the original matrices, we can independently map the documents in the source and target domains to the latent topic spaces whose dimensionality is  $K$ .

### 3.2 Finding Optimal Topic Alignments by Unsupervised Object Matching

To connect the different latent spaces, topics extracted from the source language must be aligned to one from the target language. This is reasonable because we can assume that both languages share the same latent concept.

However, we cannot quantify the similarity between the topics because we do not have any external language resources such as a dictionary. Therefore, we utilize unsupervised object matching method to find one-to-one correspondences between topics. In this paper, we employ kernelized sorting (KS) (Novi et al., 2010). KS finds the best one-to-one matching as follows:

$$\begin{aligned} \boldsymbol{\pi}^* &= \arg \max_{\boldsymbol{\pi} \in \Pi_K} \text{tr}(\bar{\mathcal{G}}_S \boldsymbol{\pi}^\top \bar{\mathcal{G}}_T \boldsymbol{\pi}), \\ \text{s.t. } &\boldsymbol{\pi} \mathbf{1}_K = \mathbf{1}_K \text{ and } \boldsymbol{\pi}^\top \mathbf{1}_K = \mathbf{1}_K. \end{aligned} \quad (3)$$

Here,  $\boldsymbol{\pi}$  is a  $K \times K$  matrix that represents the one-to-one correspondence between topics, *i.e.*  $\pi_{ij}=1$  indicates that the  $i$ th topic in the source language corresponds to the  $j$ th one of the target language.

|          | Overall Average                     |
|----------|-------------------------------------|
| KS       | 0.252 $\pm$ 0.112                   |
| CKS      | 0.249 $\pm$ 0.033                   |
| LSOM     | 0.278 $\pm$ 0.086                   |
| LSM(300) | 0.298 $\pm$ 0.077                   |
| LSM(600) | <b>0.359 <math>\pm</math> 0.062</b> |

Table 1: Average accuracy over all language pairs

$\Pi_K$  indicates the set of all possible matrices storing one-to-one correspondences.  $\mathcal{G}$  denotes the  $K \times K$  kernel matrix obtained from topic proportion,  $\mathcal{G}_{ij} = \mathcal{K}(\mathbf{W}_{i,:}^\top, \mathbf{W}_{:,j})$ , and  $\bar{\mathcal{G}}$  is the centered matrix of  $\mathcal{G}$ .  $\mathcal{K}(\cdot)$  is a kernel function.  $\mathbf{1}_K$  is a  $K$ -dimensional column vector of all ones.  $\boldsymbol{\pi}^*$  is obtained by iterative procedure.

According to  $\boldsymbol{\pi}^*$ , we obtain permuted matrices,  $\mathcal{W}_T = \mathbf{W}_T \boldsymbol{\pi}^*$  and  $\mathcal{H}_T = \boldsymbol{\pi}^{*\top} \mathbf{H}_T$ , and the product of permuted matrices is the same with that of unpermuted matrices as follows:

$$\mathbf{T} \approx \mathbf{W}_T \mathbf{H}_T = \mathcal{W}_T \mathcal{H}_T. \quad (4)$$

Fig. 1 shows the topic alignment procedure.

Since documents from both domains are represented in a shared latent space, we can directly calculate the similarity between the  $n$ th document in the source domain and the  $m$ th document in the target domain based on  $H_{T:,m}$  ( $m$ th column vector of  $H_T$ ) and  $\mathcal{H}_{S:,n}$  ( $n$ th column vector of  $\mathcal{H}_S$ ).

## 4 Cross-language Text Categorization via Latent Semantic Matching

Cross-language text categorization is the task of exploiting labeled documents in the source language (e.g. English) to classify documents in the target language (e.g. French). Suppose we have training data set  $\{s_n, y_n\}_{n=1}^N$  in the source language domain.  $y_n \in Y$  is the class label for the  $n$ th document. We can train a classifier in the  $K$ -dimensional latent space with data set  $\{\mathbf{H}_{S:,n}^\top, y_n\}_{n=1}^N$ .  $\mathbf{H}_{S:,n}$  is the projected vector of  $s_n$ . Also, the  $m$ th document in the target language domain  $t_m$  is projected into the latent space as  $\mathcal{H}_{T:,m}^\top$ . Here, the documents in both domains are projected into the same size latent space and the basis vectors of the spaces are aligned. Therefore, we can classify a document in the target domain  $t_m$  by a classifier trained with  $\{\mathbf{H}_{S:,n}^\top, y_n\}_{n=1}^N$ .

| <b>Books</b>       |  |
|--------------------|--|
| English            | Hack, Parent, tale, subversion, Interesting, centre, Paper, T., prejudice, Murphy  |
| German             | Lydia, Sebastian, Seelenbrecher, Patient, Fitzek, Patrick, Fiktion, Patientenakte, Realitt, Klinik                                 |
| <b>Electronics</b> |  |
| English            | SD800, Angle, Digital, Optical, Silver, understnad, camra, 7.1MP, P3N, 10MP  |
| German             | *****, 550D, 600D, Objektiv, Canon, ablichten, Body, Werkzeug, Kamera, einliet   |
| <b>Kitchen</b>     |  |
| English            | Briel, Electra-Craft, Chamonix, machine, Due, crema, supervisor, technician, espresso, tamp  |
| German             | ESGE, Prierkopf, Zauberstab, Gummikupplung, Suppe/Sauce, Braun , Bolognese, prieren, Testsieger, Topf                              |
| <b>Music</b>       |  |
| English            | Amy, Poison, Doherty, Schottin, Mid, Prince, Song, ausdrucksstark , Tempo, knocking  |
| German             | Norah, mini, 'Little, 'Rome, 'Come, Gardot, Lana, listenings , dreamlike, digipak  |
| <b>Watch</b>       |  |
| English            | watch, indicate, timex, HRM, month, icon, Timex, datum, troubleshooting, reasonable  |
| German             | Orient, Diver, Lnette, Leuchtpunkt, Zahlenringes, Handgelenksdurchmesser, Stoppsekunde, Uhrforum, Konsumbereiche, Schwingungen/Std |

Table 2: Examples of aligned latent topics

## 5 Experimental Evaluation

### 5.1 Experimental Settings

We compared our method, latent semantic matching (LSM), with three unsupervised object matching methods: Kernelized Sorting (KS), Convex Kernelized Sorting (CKS), Least-Squares Object Matching (LSOM). We set the number of the latent topics  $K$  to 100 and employed the  $k$ -nearest neighbor method ( $k=10$ ) as the classifier.

For, KS, CKS and LSOM, we find the one-to-one correspondence between documents in the source language and documents in the target language. Then, we assign class labels of the target documents according to the correspondence.

In order to build a corpus with various language pairs for evaluation, we crawled product reviews from Amazon U.S., German, France and Japan with five categories: 'Books', 'Electronics', 'Music', 'Kitchen', 'Watch'. The corpus is neither sentence level parallel nor comparable. For each category, we randomly select 60 documents as the test data ( $M=300$ ) for all methods and 60 documents as the training data ( $N=300$ ) for KS, CKS, LSOM and LSM(300). We also compared latent semantic matching with 120 training documents for each category ( $N=600$ ), and called this method LSM(600). Note that since KS, CKS and LSOM require that the data sizes are the same for source and target domains, they cannot use training data more than test data. To avoid local optimum solutions of NMF, we executed our methods with 100 different initialization values and chose the solution that achieved the best objective func-

tion of KS.

### 5.2 Results and Discussion

Table 1 shows average accuracies with standard division over all language pairs. From the table, classification accuracy of all methods significantly outperformed random classifier (accuracy=0.2). The results showed the effectiveness of both unsupervised object matching and latent semantic matching. When comparing LSM(300) with KS, CKS and LSOM, LSM(300) obtained better results than these unsupervised object matching methods. The result supports the effectiveness of the latent topic matching. Moreover, LSM(600) achieved the highest accuracy. There are large differences between LSM(600) and the others. This result implies not only the effectiveness of the latent topic matching but also increasing the number of source side documents (labeled training data) contributes to improving classification accuracy. This is natural in terms of supervised learning but only our method can deal with source side documents that are larger in number.

Table 2 shows examples of latent topics in English and German extracted and aligned by LSM(600). We can see that some author names, words related to camera, and cooking equipment appear in 'Books', 'Electronics' and 'Kitchen' topics, respectively. Similarity, there are some artists' names in 'Music' and watch brands in 'Watch'.

## 6 Conclusion

As an extension of unsupervised object matching, this paper proposed latent semantic matching that considers the shared latent space between two language domains. To generate such a space, topics of the target space are permuted by exploiting unsupervised object matching. We can measure distances between objects by standard metrics, which enable us retrieving k-nearest objects in the source domain for a query object in the target domain. This is a significant advantage over conventional unsupervised object matching methods. We used Amazon review corpus to demonstrate the effectiveness of our method on cross-language text categorization. The results showed that our method outperformed conventional object matching methods with the same number of training samples. Moreover, our method achieved even higher performance by utilizing more documents in the source domain.

## Acknowledgements

The authors would like to thank Nemanja Djuric for providing code for Convex Kernelized Sorting and the three anonymous reviewers for thoughtful suggestions.

## References

- Francis Bach and Michael Jordan. 2005. A probabilistic interpretation of canonical correlation analysis. Technical report, Department of Statistics, University of California, Berkeley.
- David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *JMLR*, 3(Jan.):993–1022.
- Jordan Boyd-Graber and David Blei. 2009. Multilingual topic model for unaligned text. In *Proc. of the 25th UAI*, pages 75–82.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Nemanja Djuric, Mihajlo Grbovic, and Slobodan Vucetic. 2012. Convex kernelized sorting. In *Proc. of the 26th AAAI*, pages 893–899.
- Susan Dumais, Lanauer Thomas, and Michael Littman. 1996. Automatic cross-linguistic information retrieval using latent semantic indexing. In *Proc. of the Workshop on Cross-Linguistic Information Retrieval in SIGIR*, pages 16–23.
- Alfio Gliozzo and Carlo Strapparava. 2005. Cross language text categorization by acquiring multilingual domain models from comparable corpora. In *Proc. of the ACL Workshop on Building and Using Parallel Texts*, pages 9–16.
- Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proc. of ACL-08: HLT*, pages 771–779.
- Jagarlamudi Jagadeesh and Hal Daume III. 2010. Extracting multilingual topics from unaligned corpora. In *Proc of the 32nd ECIR*, pages 444–456.
- Daniel Lee and Sebastian Seung. 2000. Algorithm for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562.
- Quadrianto Novi, Smola Alexander, Song Le, and Tuytelaars Tinne. 2010. Kernelized sorting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(10):1809–1821.
- Jhon Platt, Kristina Toutanova, and Wen-tau Yih. 2010. Translingual document representation from discriminative projections. In *Proc. of the 2010 Conference on EMNLP*, pages 251–261.
- Abhishek Tripathi, Arto Klami, and Sami Virpioja. 2010. Bilingual sentence matching using kernel CCA. In *Proc. of the 2010 IEEE International Workshop on MLSP*, pages 130–135.
- Ni Xiaochuan, Sun Lian-Tao, Hu Jian, and Chen Zheng. 2011. Cross lingual text classification by mining multilingual topics from wikipedia. In *Proc. of the 4th WSDM*, pages 375–384.
- Makoto Yamada and Masashi Sugiyama. 2011. Cross-domain object matching with model selection. In *Proc. of the 14th AISTATS*, pages 807–815.



# TopicSpam: a Topic-Model-Based Approach for Spam Detection

Jiwei Li , Claire Cardie  
School of Computer Science  
Cornell University  
Ithaca, NY, 14853  
jl3226@cornell.edu  
cardie@cs.cornell.edu

Sujian Li  
Laboratory of Computational Linguistics  
Peking University  
Beijing, P.R.China, 150001  
lisujian@pku.edu.cn

## Abstract

Product reviews are now widely used by individuals and organizations for decision making (Litvin et al., 2008; Jansen, 2010). And because of the profits at stake, people have been known to try to game the system by writing fake reviews to promote target products. As a result, the task of deceptive review detection has been gaining increasing attention. In this paper, we propose a generative LDA-based topic modeling approach for fake review detection. Our model can aptly detect the subtle differences between deceptive reviews and truthful ones and achieves about 95% accuracy on review spam datasets, outperforming existing baselines by a large margin.

## 1 Introduction

Consumers rely increasingly on user-generated online reviews to make purchase decisions. Positive opinions can result in significant financial gains. This gives rise to *deceptive opinion spam* (Ott et al., 2011; Jindal et al., 2008), fake reviews written to sound authentic and deliberately mislead readers. Previous research has shown that humans have difficulty distinguishing fake from truthful reviews, operating for the most part at chance (Ott et al., 2011). Consider, for example, the following two hotel reviews. One is truthful and the other is deceptive<sup>1</sup>:

1. *My husband and I stayed for two nights at the Hilton Chicago. We were very pleased with the accommodations and enjoyed the service every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free wifi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided. Their service was amazing,*

<sup>1</sup>The first example is a deceptive review.

- and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.*
2. *We stayed at the Sheraton by Navy Pier the first weekend of November. The view from both rooms was spectacular (as you can tell from the picture attached). They also left a plate of cookies and treats in the kids room upon check-in made us all feel very special. The hotel is central to both Navy Pier and Michigan Ave. so we walked, trolleyed, and cabbied all around the area. We ate the breakfast buffet on both mornings and thought it was pretty good. The eggs were a little runny. Our six year old ate free and our two eleven year old were \$14 (instead of the adult \$20). The rooms were clean, the concierge and reception staff were both friendly and helpful...we will definitely visit this Sheraton again when we stay in Chicago next time.*

Because of the difficulty of recognizing deceptive opinions, there has been a widespread and growing interest in developing automatic, usually learning-based methods to help users identify deceptive reviews (Ott et al., 2011; Jindal et al., 2008; Jindal et al., 2010; Li et al., 2011; Lim et al., 2011; Wang et al., 2011).

The state-of-the-art approach treats the task of spam detection as a *text categorization* problem and was first introduced by Jindal and Liu (2009) who trained a supervised classifier to distinguish duplicated reviews (assumed deceptive) from original ones (assumed truthful). Since then, many supervised approaches have been proposed for spam detection. Ott et al. (2011) employed standard word and part-of-speech (POS) n-gram features for supervised learning and built a *gold – standard* opinion dataset of 800 reviews. Lim et al. (2010) proposed the inclusion of user behavior-based features and found that behavior abnormalities of reviewers could predict spammers, without using any textual features. Li et al. (2011) carefully explored review-related features based on content and sentiment, training a semi-supervised classifier for opinion spam detection. However, the disadvantages of standard supervised learning methods are obvious. First, they do not generally provide readers with a clear probabilistic pre-

diction of how likely a review is to be deceptive vs. truthful. Furthermore, identifying features that provide direct evidence against deceptive reviews is always a hard problem.

LDA topic models (Blei et al., 2003) have widely been used for their ability to model latent topics in document collection. In LDA, each document is presented as a mixture distribution of topics and each topic is presented as a mixture distribution of words. Researchers also integrated different levels of information into LDA topic models to model the specific knowledge that they are interested in, such as user-specific information (Rosen-zvi et al., 2006), document-specific information (Li et al., 2010) and time-specific information (Diao et al., 2012). Ramage et al. (2009) developed a Labeled LDA model to define a one-to-one correspondence between LDA latent topics and tags. Chemudugunta et al. (2008) illustrated that by considering background information and document-specific information, we can largely improve the performance of topic modeling.

In this paper, we propose a Bayesian approach called TopicSpam for deceptive review detection. Our approach, which is a variation of Latent Dirichlet Allocation (LDA) (Blei et al., 2003), aims to detect the subtle differences between the topic-word distributions of deceptive reviews vs. truthful ones. In addition, our model can give a clear probabilistic prediction on how likely a review should be treated as deceptive or truthful. Performance is tested on dataset from Ott et al.(2011) that contains 800 reviews of 20 Chicago hotels. Our model achieves more than 94% accuracy on that dataset.

## 2 TopicSpam

We are presented with four subsets of hotel reviews,  $M = \{M_i\}_{i=1}^4$ , representing *deceptive train*, *truthful train*, *deceptive test* and *truthful test* data, respectively. Each review  $r$  is comprised of a number of words  $r = \{w_t\}_{t=1}^{t=n_r}$ . Input for the TopicSpam algorithm is the datasets  $M$ ; output is the label (deceptive, truthful) for each review in  $M_3$  and  $M_4$ .  $V$  denotes vocabulary size.

### 2.1 Details of TopicSpam

In TopicSpam, each document is modeled as a bag of words, which are assumed to be generated from a mixture of latent topics. Each word is associated with a latent variable that specifies

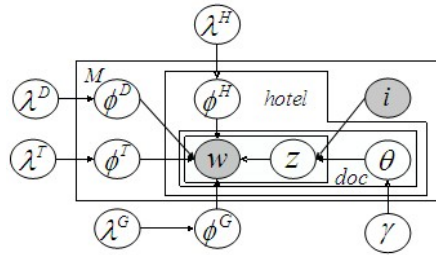


Figure 1: Graphical Model for TopicSpam

the topic from which it is generated. Words in a document are assumed to be conditionally independent given the hidden topics. A general background distribution  $\phi^B$  and hotel-specific distributions  $\phi^{H_j}$  ( $j = 1, \dots, 20$ ) are first introduced to capture the background information and hotel-specific information. To capture the difference between deceptive reviews and truthful reviews, TopicSpam also learns a deceptive topic distribution  $\phi^D$  and truthful topic distribution  $\phi^T$ . The generative model of TopicSpam is shown as follows:

- For a training review in  $r_{1j} \in M_1$ , words are originated from one of the three different topics:  $\phi^B$ ,  $\phi^{H_j}$  and  $\phi^D$ .
- For a training review in  $r_{2j} \in M_2$ , words are originated from one of the three different topics:  $\phi^B$ ,  $\phi^{H_j}$  and  $\phi^T$ .
- For a test review in  $r_{mj} \in M_m, m = 3, 4$ , words are originated from one of the four different topics:  $\phi^B$ ,  $\phi^{H_j}$ ,  $\phi^D$  and  $\phi^T$ .

The generation process of TopicSpam is shown in Figure 1 and the corresponding graphical model is illustrated in Figure 2. We use  $\lambda = (\lambda_G, \lambda_{H_i}, \lambda_D, \lambda_T)$  to represent the asymmetric priors for topic-word distribution generation. In our experiments, we set  $\lambda_G = 0.1$ , and  $\lambda_{H_i} = \lambda_D = \lambda_T = 0.01$ . The intuition for the asymmetric priors is that there should be more words assigned to the background topic.  $\gamma = [\gamma_B, \gamma_{H_i}, \gamma_D, \gamma_T]$  denotes the priors for the document-level topic distribution in the LDA model. We set  $\gamma_B = 2$  and  $\gamma_T = \gamma_D = \gamma_{H_i} = 1$ , reflecting the intuition that more words in each document should cover the background topic.

### 2.2 Inference

We adopt the collapsed Gibbs sampling strategy to infer the latent parameters in TopicSpam. In Gibbs

- 
1. sample  $\phi^G \sim Dir(\lambda^G)$
  2. sample  $\phi^D \sim Dir(\lambda^D)$
  3. sample  $\phi^T \sim Dir(\lambda^T)$
  4. for each hotel  $j \in [1, N]$ : sample  $\phi^{H_j} \sim \lambda^H$
  5. for each review  $r$ 
    - if  $i=1$ : sample  $\theta_r \sim Dir(\gamma_B, \gamma_{H_j}, \gamma_D)$
    - if  $i=2$ : sample  $\theta_r \sim Dir(\gamma_B, \gamma_{H_j}, \gamma_T)$
    - if  $i=3$ : sample  $\theta_r \sim Dir(\gamma_B, \gamma_{H_j}, \gamma_D, \gamma_T)$
    - if  $i=4$ : sample  $\theta_r \sim Dir(\gamma_B, \gamma_{H_j}, \gamma_D, \gamma_T)$
- for each word  $w$  in  $R$
- sample  $z \sim \theta_r$     sample  $w \sim \phi^z$
- 

Figure 2: Generative Model for TopicSpam

sampling, for each word  $w$  in review  $r$ , we need to calculate  $P(z_w|w, z_{-w}, \gamma, \lambda)$  in each iteration, where  $z_{-w}$  denotes the topic assignments for all words except that of the current word  $z_w$ .

$$P(z_w = m|z_{-w}, i, j, \gamma, \lambda) = \frac{N_r^m + \gamma_m}{\sum_{m'} (N_r^{m'} + \gamma_{m'})} \cdot \frac{E_m^w + \lambda_m}{\sum_{w'} E_m^w + V\lambda_m} \quad (1)$$

where  $N_r^m$  denotes the number of times that topic  $m$  appears in current review  $r$  and  $E_m^w$  denotes the number of times that word  $w$  is assigned to topic  $m$ . After each sampling iteration, the latent parameters can be estimated using the following formulas:

$$\theta_r^m = \frac{N_r^m + \gamma_m}{\sum_{m'} (N_r^{m'} + \gamma_{m'})} \quad \phi_m^{(w)} = \frac{E_m^w + \lambda_m}{\sum_{w'} E_m^w + V\lambda_m} \quad (2)$$

### 2.3 Labeling the Test Data

For each review  $r$  in the test data, let  $N_r^D$  denote the number of words generated from the deceptive topic and  $N_r^T$ , the number of words generated from the truthful topic. The decision for whether a review is deceptive or truthful is made as follows:

- if  $N_r^D > N_r^T$ ,  $r$  is deceptive.
- if  $N_r^D < N_r^T$ ,  $r$  is truthful.
- if  $N_r^D = N_r^T$ , it is hard to decide.

Let  $P(D)$  denote the probability that  $r$  is deceptive and  $P(T)$  denote the probability that  $r$  is truthful.

$$P(D) = \frac{N_r^D}{N_r^D + N_r^T} \quad P(T) = \frac{N_r^T}{N_r^D + N_r^T} \quad (3)$$

## 3 Experiments

### 3.1 System Description

Our experiments are conducted on the dataset from Ott et al.(2011), which contains reviews of the 20 most popular hotels on TripAdvisor in the Chicago areas. There are 20 truthful and 20 deceptive reviews for each of the chosen hotels (800 reviews total). Deceptive reviews are gathered using Amazon Mechanical Turk<sup>2</sup>. In our experiments, we adopt the same 5-fold cross-validation strategy as in Ott et al., using the same data partitions. Words are stemmed using PorterStemmer<sup>3</sup>.

### 3.2 Baselines

We employ a number of techniques as baselines:

**TopicTD:** A topic-modeling approach that only considers two topics: deceptive and truthful. Words in *deceptive train* are all generated from the deceptive topic and words in *truthful train* are generated from the truthful topic. Test documents are presented with a mixture of the deceptive and truthful topics.

**TopicTDB:** A topic-modeling approach that only considers background, deceptive and truthful information.

**SVM-Unigram:** Using SVMlight(Joachims, 1999) to train linear SVM models on unigram features.

**SVM-Bigram:** Using SVMlight(Joachims, 1999) to train linear SVM models on bigram features.

**SVM-Unigram-Removal1:** In SVM-Unigram-Removal, we first train TopicSpam. Then words generated from hotel-specific topics are removed. We use the remaining words as features in SVM-light.

**SVM-Unigram-Removal2:** Same as SVM-Unigram-removal-1 but removing all background words and hotel-specific words.

Experimental results are shown in Table 1<sup>4</sup>. As we can see, the accuracy of TopicSpam is 0.948, outperforming TopicTD by 6.4%. This illustrates the effectiveness of modeling background and hotel-specific information for the opinion spam detection problem. We also see that TopicSpam slightly outperforms TopicTDB, which

<sup>2</sup><https://www.mturk.com/mturk/>.

<sup>3</sup><http://tartarus.org/martin/PorterStemmer/>

<sup>4</sup>Reviews with  $N_r^D = N_r^T$  are regarded as incorrectly classified by TopicSpam.

| Approach             | Accuracy | T-P   | T-R   | T-F   | D-P   | D-R   | D-F   |
|----------------------|----------|-------|-------|-------|-------|-------|-------|
| TopicSpam            | 0.948    | 0.954 | 0.942 | 0.944 | 0.941 | 0.952 | 0.946 |
| TopicTD              | 0.888    | 0.901 | 0.878 | 0.889 | 0.875 | 0.897 | 0.886 |
| TopicTDB             | 0.931    | 0.938 | 0.926 | 0.932 | 0.925 | 0.937 | 0.930 |
| SVM-Unigram          | 0.884    | 0.899 | 0.865 | 0.882 | 0.870 | 0.903 | 0.886 |
| SVM-Bigram           | 0.896    | 0.901 | 0.890 | 0.896 | 0.891 | 0.903 | 0.897 |
| SVM-Unigram-Removal1 | 0.895    | 0.906 | 0.889 | 0.898 | 0.887 | 0.907 | 0.898 |
| SVM-Unigram-Removal2 | 0.822    | 0.852 | 0.806 | 0.829 | 0.793 | 0.840 | 0.817 |

Table 1: Performance for different approaches based on nested 5-fold cross-validation experiments.

neglects hotel-specific information. By checking the results of Gibbs sampling, we find that this is because only a small number of words are generated by the hotel-specific topics. TopicTD and SVM-Unigram get comparative accuracy rates. This can be explained by the fact that both models use unigram frequency as features for the classifier or topic distribution training. SVM-Unigram-Removal1 is also slightly better than SVM-Unigram. In SVM-Unigram-removal1, hotel-specific words are removed for classifier training. So the first-step LDA model can be viewed as a feature selection process for the SVM, giving rise to better results. We can also see that the performance of SVM-Unigram-removal2 is worse than other baselines. This can be explained as follows: for example, word "my" has large probability to be generated from the background topic. However it can also be generated by deceptive topic occasionally but can hardly be generated from the truthful topic. So the removal of these words results in the loss of useful information, and leads to low accuracy rate.

Our topic-modeling approach uses word frequency as features and does not involve any feature selection process. Here we present the results of the sample reviews from Section 1. Stop words are labeled in black, background topics (B) in blue, hotel specific topics (H) in orange, deceptive topics (D) in red and truthful topic (T) in green.

1. *My husband and I stayed for two nights at the Hilton Chicago. We were very pleased with the accommodations and enjoyed the service every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free wifi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.*  
[B,H,D,T]=[41,6,10,1] p(D)=0.909 P(T)=0.091

2. *We stayed at the Sheraton by Navy Pier the first weekend of November. The view from both rooms was spec-*

*tacular (as you can tell from the picture attached). They also left a plate of cookies and treats in the kids room upon check-in made us all feel very special. The hotel is central to both Navy Pier and Michigan Ave. so we walked, trolleyed, and cabbied all around the area. We ate the breakfast buffet both mornings and thought it was pretty good. The eggs were a little runny. Our six year old ate free and our two eleven year old were \$14 ( instead of the adult \$20) The rooms were clean, the concierge and reception staff were both friendly and helpful...we will definitely visit this Sheraton again when we're in Chicago next time.*

[B,H,D,T]=[80,15,3,18] p(D)=0.143 P(T)=0.857

|            |            |          |            |
|------------|------------|----------|------------|
| background | deceptive  | truthful | Hilton     |
| hotel      | hotel      | room     | Hilton     |
| stay       | my         | )        | palmer     |
| we         | chicago    | (        | millennium |
| room       | will       | but      | lockwood   |
| !          | room       | \$       | park       |
| Chicago    | very       | bathroom | lobby      |
| my         | visit      | location | line       |
| great      | husband    | night    | valet      |
| I          | city       | walk     | shampoo    |
| very       | experience | park     | dog        |
| Omni       | Amalfi     | Sheraton | James      |
| Omni       | Amalfi     | tower    | James      |
| pool       | breakfast  | Sheraton | service    |
| plasma     | view       | pool     | spa        |
| sundeck    | floor      | river    | bar        |
| chocolate  | bathroom   | lake     | upgrade    |
| indoor     | cocktail   | navy     | primehouse |
| request    | morning    | indoor   | design     |
| pillow     | wine       | shower   | overlook   |
| suitable   | great      | kid      | romantic   |
| area       | room       | theater  | home       |

Table 2: Top words in different topics from Topic-Spam

## 4 Conclusion

In this paper, we propose a novel topic model for deceptive opinion spam detection. Our model achieves an accuracy of 94.8%, demonstrating its effectiveness on the task.

## 5 Acknowledgements

We thank Myle Ott for his insightful comments and suggestions. This work was supported in part by NSF Grant BCS-0904822, a DARPA Deft grant, and by a gift from Google.

## References

- David Blei, Andrew Ng and Micheal Jordan. Latent Dirichlet allocation. 2003. In *Journal of Machine Learning Research*.
- Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. In *Proceedings of annual international ACM SIGIR conference on Research and development in information retrieval, 2006*.
- Chaltanya Chemudugunta, Padhraic Smyth and Mark Steyers. Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model.. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*.
- Paul-Alexandru Chirita, Jorg Diederich, and Wolfgang Nejdl. MailRank: using ranking for spam detection. In *Proceedings of ACM international conference on Information and knowledge management. 2005*.
- Harris Drucke, Donghui Wu, and Vladimir Vapnik. 2002. Support vector machines for spam categorization. In *Neural Networks*.
- Qiming Diao, Jing Jiang, Feida Zhu and Ee-Peng Lim. In *Proceeding of the 50th Annual Meeting of the Association for Computational Linguistics. 2012*
- Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods*.
- Jack Jansen. 2010. Online product research. In *Pew Internet and American Life Project Report*.
- Nitin Jindal, and Bing Liu. Opinion spam and analysis. 2008. In *Proceedings of the international conference on Web search and web data mining*
- Nitin Jindal, Bing Liu, and Ee-Peng Lim. Finding Unusual Review Patterns Using Unexpected Rules. 2010. In *Proceedings of the 19th ACM international conference on Information and knowledge management*
- Pranam Kolari, Akshay Java, Tim Finin, Tim Oates and Anupam Joshi. Detecting Spam Blogs: A Machine Learning Approach. In *Proceedings of Association for the Advancement of Artificial Intelligence. 2006*.
- Peng Li, Jing Jiang and Yinglin Wang. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.
- Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review Spam. 2011. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*.
- Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting Product Review Spammers Using Rating Behavior. 2010. In *Proceedings of the 19th ACM international conference on Information and knowledge management*.
- Stephen Litvina, Ronald Goldsmithb and Bing Pana. 2008. Electronic word-of-mouth in hospitality and tourism management. *Tourism management*, 29(3):458468.
- Juan Martinez-Romo and Lourdes Araujo. Web Spam Identification Through Language Model Analysis. In *AIRWeb. 2009*.
- Arjun Mukherjee, Bing Liu and Natalie Glance. Spotting Fake Reviewer Groups in Consumer Reviews. In *Proceedings of the 18th international conference on World wide web, 2012*.
- Alexandros Ntoulas, Marc Najork, Mark Manasse and Dennis Fetterly. Detecting Spam Web Pages through Content Analysis. In *Proceedings of international conference on World Wide Web 2006*
- Myle Ott, Yejin Choi, Claire Cardie and Jeffrey Hancock. Finding deceptive opinion spam by any stretch of the imagination. 2011. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*
- Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. In *Found. Trends Inf. Retr.*
- Daniel Ramage, David Hall, Ramesh Nallapati and Christopher D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. 2009. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing 2009*.
- Michal Rosen-zvi, Thomas Griffith, Mark Steyvers and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*.
- Guan Wang, Sihong Xie, Bing Liu and Philip Yu. Review Graph based Online Store Review Spammer Detection. 2011. In *Proceedings of 11th International Conference of Data Mining*.
- Baoning Wu, Vinay Goel and Brian Davison. Topical TrustRank: using topicality to combat Web spam. In *Proceedings of international conference on World Wide Web 2006* .
- Kyang Yoo and Ulrike Gretzel. 2009. Comparison of Deceptive and Truthful Travel Reviews. In *Information and Communication Technologies in Tourism 2009*.

# Semantic Neighborhoods as Hypergraphs

Chris Quirk and Pallavi Choudhury

Microsoft Research

One Microsoft Way

Redmond, WA 98052, USA

{chrisq,pallavic}@microsoft.com

## Abstract

Ambiguity preserving representations such as lattices are very useful in a number of NLP tasks, including paraphrase generation, paraphrase recognition, and machine translation evaluation. Lattices compactly represent lexical variation, but word order variation leads to a combinatorial explosion of states. We advocate hypergraphs as compact representations for sets of utterances describing the same event or object. We present a method to construct hypergraphs from sets of utterances, and evaluate this method on a simple recognition task. Given a set of utterances that describe a single object or event, we construct such a hypergraph, and demonstrate that it can recognize novel descriptions of the same event with high accuracy.

## 1 Introduction

Humans can construct a broad range of descriptions for almost any object or event. In this paper, we will refer to such objects or events as groundings, in the sense of grounded semantics. Examples of groundings include pictures (Rashtchian et al., 2010), videos (Chen and Dolan, 2011), translations of a sentence from another language (Dreyer and Marcu, 2012), or even paraphrases of the same sentence (Barzilay and Lee, 2003).

One crucial problem is recognizing whether novel utterances are relevant descriptions of those groundings. In the case of machine translation, this is the evaluation problem; for images and videos, this is recognition and retrieval. Generating descriptions of events is also often an interesting task: we might like to find a novel paraphrase for a given sentence, or generate a description of a grounding that meets certain criteria (*e.g.*, brevity, use of a restricted vocabulary).

Much prior work has used lattices to compactly represent a range of lexical choices (Pang et al., 2003). However, lattices cannot compactly represent alternate word orders, a common occurrence in linguistic descriptions. Consider the following excerpts from a video description corpus (Chen and Dolan, 2011):

- A man is sliding a cat on the floor.
- A boy is cleaning the floor with the cat.
- A cat is being pushed across the floor by a man.

Ideally we would like to recognize that the following utterance is also a valid description of that event: *A cat is being pushed across the floor by a boy*. That is difficult with lattice representations.

Consider the following context free grammar:

$$\begin{aligned} S &\rightarrow X_0 X_1 \\ &\quad | X_2 X_3 \\ X_0 &\rightarrow a\ man \mid a\ boy \\ X_1 &\rightarrow is\ sliding\ X_2\ on\ X_4 \\ &\quad | is\ cleaning\ X_4\ with\ X_2 \\ X_2 &\rightarrow a\ cat \mid the\ cat \\ X_3 &\rightarrow is\ being\ pushed\ across\ X_4\ by\ X_0 \\ X_4 &\rightarrow the\ floor \end{aligned}$$

This grammar compactly captures many lexical and syntactic variants of the input set. Note how the labels act as a kind of multiple-sequence-alignment allowing reordering: spans of tokens covered by the same label are, in a sense, aligned. This hypergraph or grammar represents a *semantic neighborhood*: a set of utterances that describe the same entity in a semantic space.

Semantic neighborhoods are defined in terms of a grounding. Two utterances are neighbors with respect to some grounding (semantic event) if they are both descriptions of that grounding. Paraphrases, in contrast, may be defined over all possible groundings. That is, two words or phrases

are considered paraphrases if there exists some grounding that they both describe. The paraphrase relation is more permissive than the semantic neighbor relation in that regard. We believe that it is much easier to define and evaluate semantic neighbors. Human annotators may have difficulty separating paraphrases from unrelated or merely related utterances, and this line may not be consistent between judges. Annotating whether an utterance clearly describes a grounding is a much easier task.

This paper describes a simple method for constructing hypergraph-shaped Semantic Neighborhoods from sets of expressions describing the same grounding. The method is evaluated in a paraphrase recognition task, inspired by a CAPTCHA task (Von Ahn et al., 2003).

## 2 Inducing neighborhoods

Constructing a hypergraph to capture a set of utterances is a variant of grammar induction. Given a sample of positive examples, we infer a compact and accurate description of the underlying language. Conventional grammar induction attempts to define the set of grammatical sentences in the language. Here, we search for a grammar over the fluent and adequate descriptions of a particular input. Many of the same techniques still apply.

Rather than starting from scratch, we bootstrap from an existing English parser. We begin by parsing the set of input utterances. This parsed set of utterances acts as a sort of treebank. Reading off a grammar from this treebank produces a grammar that can generate not only the seed sentences, but also a broad range of nearby sentences. In the case above with *cat*, *man*, and *boy*, we would be able to generate cases legitimate variants where *man* was replaced by *boy* as well as undesired variants where *man* is replaced by *cat* or *floor*. This initial grammar captures a large neighborhood of nearby utterances including many such undesirable ones. Therefore, we refine the grammar.

Refinements have been in common use in syntactic parsing for years now. Inspired by the result that manual annotations of Treebank categories can substantially increase parser accuracy (Klein and Manning, 2003), several approaches have been introduced to automatically induce latent symbols on existing trees. We use the split-merge method commonly used in syntactic parsing (Petrov et al., 2006). In its original setting,

the refinements captured details beyond that of the original Penn Treebank symbols. Here, we capture both syntactic and semantic regularities in the descriptions of a given grounding.

As we perform more rounds of refinement, the grammar becomes tightly constrained to the original sentences. Indeed, if we iterated to a fixed point, the resulting grammar would parse only the original sentences. This is a common dilemma in paraphrase learning: the safest meaning preserving rewrite is to change nothing. We optimize the number of split-merge rounds for task-accuracy; two or three rounds works well in practice. Figure 1 illustrates the process.

### 2.1 Split-merge induction

We begin with a set of utterances that describe a specific grounding. They are parsed with a conventional Penn Treebank parser (Quirk et al., 2012) to produce a type of treebank. Unlike conventional treebanks which are annotated by human experts, the trees here are automatically created and thus are more likely to contain errors. This treebank is the input to the split-merge process.

**Split:** Given an input treebank, we propose refinements of the symbols in hopes of increasing the likelihood of the data. For each original symbol in the grammar such as NP, we consider two latent refinements:  $NP_0$  and  $NP_1$ . Each binary rule then produces 8 possible variants, since the parent, left child, and right child now have two possible refinements. The parameters of this grammar are then optimized using EM. Although we do not know the correct set of latent annotations, we can search for the parameters that optimize the likelihood of the given treebank. We initialize the parameters of this refined grammar with the counts from the original grammar along with a small random number. This randomness prevents EM from starting on a saddle point by breaking symmetries; Petrov et al. describe this in more detail.

**Merge:** After EM has run to completion, we have a new grammar with twice as many symbols and eight times as many rules. Many of these symbols may not be necessary, however. For instance, nouns may require substantial refinement to distinguish a number of different actors and objects, where determiners might not require much refinement at all. Therefore, we discard the splits that led to the least increase in likelihood, and then reestimate the grammar once again.

(a) Input:

- the man plays the piano
- the guy plays the keyboard

(b) Parses:

- (S (NP (DT the) (NN man))  
(VP (VBZ plays)  
(NP (DT the) (NN piano))))
- (S (NP (DT the) (NN guy))  
(VP (VBZ plays)  
(NP (DT the) (NN keyboard))))

(c) Parses with latent annotations:

- (S (NP<sub>0</sub> (DT the) (NN<sub>0</sub> man))  
(VP (VBZ plays)  
(NP<sub>1</sub> (DT the) (NN<sub>1</sub> piano))))
- (S (NP<sub>0</sub> (DT the) (NN<sub>0</sub> guy))  
(VP (VBZ plays)  
(NP<sub>1</sub> (DT the) (NN<sub>1</sub> keyboard))))

(d) Refined grammar:

S → NP<sub>0</sub> VP  
NP<sub>0</sub> → DT NN<sub>0</sub>  
NP<sub>1</sub> → DT NN<sub>1</sub>  
NP → VBZ NP<sub>1</sub>  
DT → *the*  
NN<sub>0</sub> → *man | guy*  
NN<sub>1</sub> → *piano | keyboard*  
VBZ → *plays*

Figure 1: Example of hypergraph induction. First a conventional Treebank parser converts input utterances (a) into parse trees (b). A grammar could be directly read from this small treebank, but it would conflate all phrases of the same type. Instead we induce latent refinements of this small treebank (c). The resulting grammar (d) can match and generate novel variants of these inputs, such as *the man plays the keyboard* and *the guy plays the piano*. While this simplified example suggests a single hard assignment of latent annotations to symbols, in practice we maintain a distribution over these latent annotations and extract a weighted grammar.

**Iteration:** We run this process in series. First the original grammar is split, then some of the least useful splits are discarded. This refined grammar is then split again, with the least useful splits discarded once again. We repeat for a number of iterations based on task accuracy.

**Final grammar estimation:** The EM procedure used during split and merge assigns fractional counts  $c(\cdot \cdot \cdot)$  to each refined symbol  $X_i$  and each production  $X_i \rightarrow Y_j Z_k$ . We estimate the final

grammar using these fractional counts.

$$P(X_i \rightarrow Y_j Z_k) = \frac{c(X_i, Y_j, Z_k)}{c(X_i)}$$

In Petrov et al., these latent refinements are later discarded as the goal is to find the best parse with the original coarse symbols. Here, we retain the latent refinements during parsing, since they distinguish semantically related utterances from unrelated utterances. Note in Figure 1 how NN<sub>0</sub> and NN<sub>1</sub> refer to different objects; were we to ignore that distinction, the parser would recognize semantically different utterances such as *the piano plays the piano*.

## 2.2 Pruning and smoothing

For both speed and accuracy, we may also prune the resulting rules. Pruning low probability rules increases the speed of parsing, and tends to increase the precision of the matching operation at the cost of recall. Here we only use an absolute threshold; we vary this threshold and inspect the impact on task accuracy. Once the fully refined grammar has been trained, we only retain those rules with a probability above some threshold. By varying this threshold  $t$  we can adjust precision and recall: as the low probability rules are removed from the grammar, precision tends to increase and recall tends to decrease.

Another critical issue, especially in these small grammars, is smoothing. When parsing with a grammar obtained from only 20 to 50 sentences, we are very likely to encounter words that have never been seen before. We may reasonably reject such sentences under the assumption that they are describing words not present in the training corpus. However, this may be overly restrictive: we might see additional adjectives, for instance. In this work, we perform a very simple form of smoothing. If the fractional count of a word given a pre-terminal symbol falls below a threshold  $k$ , then we consider that instance rare and reserve a fraction of its probability mass for unseen words. This accounts for lexical variation of the grounding, especially in the least consistently used words.

Substantial speedups could be attained by using finite state approximations of this grammar: matching complexity drops to cubic to linear in the length of the input. A broad range of approximations are available (Nederhof, 2000). Since the small grammars in our evaluation below seldom exhibit self-embedding (latent state identification



tends to remove recursion), these approximations would often be tight.

### 3 Experimental evaluation

We explore a task in description recognition. Given a large set of videos and a number of descriptions for each video (Chen and Dolan, 2011), we build a system that can recognize fluent and accurate descriptions of videos. Such a recognizer has a number of uses. One example currently in evaluation is a novel CAPTCHAs: to differentiate a human from a bot, a video is presented, and the response must be a reasonably accurate and fluent description of this video.

We split the above data into training and test. From the training sets, we build a set of recognizers. Then we present these recognizers with a series of inputs, some of which are from the held out set of correct descriptions of this video, and some of which are from descriptions of other videos. Based on discussions with authors of CAPTCHA systems, a ratio of actual users to spammers of 2:1 seemed reasonable, so we selected one negative example for every two positives. This simulates the accuracy of the system when presented with a simple bot that supplies random, well-formed text as CAPTCHA answers.<sup>1</sup>

As a baseline, we compare against a simple tf-idf approach. In this baseline we first pool all the training descriptions of the video into a single virtual document. We gather term frequencies and inverse document frequencies across the whole corpus. An incoming utterance to be classified is scored by computing the dot product of its counted terms with each document; it is assigned to the document with the highest dot product (cosine similarity).

Table 2 demonstrates that a baseline tf-idf approach is a reasonable starting point. An oracle selection from among the top three is the best performance – clearly this is a reasonable approach. That said, grammar based approach shows improvements over the baseline tf-idf, especially in recall. Recall is crucial in a CAPTCHA style task: if we fail to recognize utterances provided by humans, we risk frustration or abandonment of the service protected by the CAPTCHA. The relative importance of false positives versus false negatives

<sup>1</sup>A bot might perform object recognition on the videos and supply a stream of object names. We might simulate this by classifying utterances consisting of appropriate object words but without appropriate syntax or function words.

|              |              |         |
|--------------|--------------|---------|
| Total videos |              | 2,029   |
| Training     | descriptions | 22,198  |
|              | types        | 5,497   |
|              | tokens       | 159,963 |
| Testing      | descriptions | 15,934  |
|              | types        | 4,075   |
|              | tokens       | 114,399 |

Table 1: Characteristics of the evaluation data. The descriptions from the video description corpus are randomly partitioned into training and test.

(a)

| Algorithm             | $S$ | $k$ | Prec | Rec  | F-0  |
|-----------------------|-----|-----|------|------|------|
| tf-idf                |     |     | 99.9 | 46.6 | 63.6 |
| tf-idf (top 3 oracle) |     |     | 99.9 | 65.3 | 79.0 |
| grammar               | 2   | 1   | 86.6 | 51.5 | 64.6 |
|                       | 2   | 4   | 80.2 | 62.6 | 70.3 |
|                       | 2   | 16  | 74.2 | 74.2 | 74.2 |
|                       | 2   | 32  | 73.5 | 76.4 | 74.9 |
|                       | 3   | 1   | 91.1 | 43.9 | 59.2 |
|                       | 3   | 4   | 83.7 | 54.4 | 65.9 |
|                       | 3   | 16  | 77.3 | 65.7 | 71.1 |
|                       | 3   | 32  | 76.4 | 68.1 | 72.0 |
|                       | 4   | 1   | 94.1 | 39.7 | 55.8 |
|                       | 4   | 4   | 85.5 | 51.1 | 64.0 |
|                       | 4   | 16  | 79.1 | 61.5 | 69.2 |
|                       | 4   | 32  | 78.2 | 63.9 | 70.3 |

(b)

| $t$                       | $S$ | Prec | Rec  | F-0  |
|---------------------------|-----|------|------|------|
| $\geq 4.5 \times 10^{-5}$ | 2   | 74.8 | 73.9 | 74.4 |
| $\geq 4.5 \times 10^{-5}$ | 3   | 79.6 | 60.9 | 69.0 |
| $\geq 4.5 \times 10^{-5}$ | 4   | 82.5 | 53.2 | 64.7 |
| $\geq 3.1 \times 10^{-7}$ | 2   | 74.2 | 75.0 | 74.6 |
| $\geq 3.1 \times 10^{-7}$ | 3   | 78.1 | 64.6 | 70.7 |
| $\geq 3.1 \times 10^{-7}$ | 4   | 80.7 | 58.8 | 68.1 |
| $> 0$                     | 2   | 73.4 | 76.4 | 74.9 |
| $> 0$                     | 3   | 76.4 | 68.1 | 72.0 |
| $> 0$                     | 4   | 78.2 | 63.9 | 70.3 |

Table 2: Experimental results. (a) Comparison of tf-idf baseline against grammar based approach, varying several free parameters. An oracle checks if the correct video is in the top three. For the grammar variants, the number of splits  $S$  and the smoothing threshold  $k$  are varied. (b) Variations on the rule pruning threshold  $t$  and number of split-merge rounds  $S$ .  $> 0$  indicates that all rules are retained. Here the smoothing threshold  $k$  is fixed at 32.

(a) Input descriptions:

- A cat pops a bunch of little balloons that are on the ground.
- A dog attacks a bunch of balloons.
- A dog is biting balloons and popping them.
- A dog is playing balloons.
- A dog is playing with balloons.
- A dog is playing with balls.
- A dog is popping balloons with its teeth.
- A dog is popping balloons.
- A dog is popping balloons.
- A dog plays with a bunch of balloons.
- A small dog is attacking balloons.
- The dog enjoyed popping balloons.
- The dog popped the balloons.

(b) Top ranked yields from the resulting grammar:

- +0.085 A dog is popping balloons.
- +0.062 A dog is playing with balloons.
- +0.038 A dog is playing balloons.
- 0.038 A dog is attacking balloons.
- +0.023 A dog plays with a bunch of balloons.
- +0.023 A dog attacks a bunch of balloons.
- 0.023 A dog pops a bunch of balloons.
- 0.023 A dog popped a bunch of balloons.
- 0.023 A dog enjoyed a bunch of balloons.
- 0.018 The dog is popping balloons.
- 0.015 A dog is biting balloons.
- 0.015 A dog is playing with them.
- 0.015 A dog is playing with its teeth.

Figure 2: Example yields from a small grammar. The descriptions in (a) were parsed as-is (including the typographical error “ground”), and a refined grammar was trained with 4 splits. The top  $k$  yields from this grammar along with the probability of that derivation are listed in (b). A ‘+’ symbol indicates that the yield was in the training set. No smoothing or pruning was performed on this grammar.

may vary depending on the underlying resource. Adjusting the free parameters of this method allows us to achieve different thresholds. We can see that rule pruning does not have a large impact on overall results, though it does allow yet another means of trading off precision vs. recall.

## 4 Conclusions

We have presented a method for automatically constructing compact representations of linguistic variation. Although the initial evaluation only explored a simple recognition task, we feel the underlying approach is relevant to many linguistic tasks including machine translation evaluation, and natural language command and control systems. The induction procedure is rather simple but effective, and addresses some of the reordering limitations associated with prior approaches. (Barzilay and Lee, 2003) In effect, we are performing a multiple sequence alignment that allows reordering operations. The refined symbols of the grammar act as a correspondence between related inputs.

The quality of the input parser is crucial. This method only considers one possible parse of the input. A straightforward extension would be to consider an  $n$ -best list or packed forest of input parses, which would allow the method to move past errors in the first input process. Perhaps also this reliance on symbols from the original Treebank is not ideal. We could merge away some or all of the original distinctions, or explore different parameterizations of the grammar that allow more flexibility in parsing.

The handling of unseen words is very simple. We are investigating means of including additional paraphrase resources into the training to increase the effective lexical knowledge of the system. It is inefficient to learn each grammar independently. By sharing parameters across different groundings, we should be able to identify Semantic Neighborhoods with fewer training instances.

## Acknowledgments

We would like to thank William Dolan and the anonymous reviewers for their valuable feedback.

## References

- Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT*.
- David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Markus Dreyer and Daniel Marcu. 2012. Hyter: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171, Montréal, Canada, June. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo,

- Japan, July. Association for Computational Linguistics.
- Mark-Jan Nederhof. 2000. Practical experiments with regular approximation of context-free languages. *Computational Linguistics*, 26(1):17–44, March.
- Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences.
- 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.
- Chris Quirk, Pallavi Choudhury, Jianfeng Gao, Hisami Suzuki, Kristina Toutanova, Michael Gamon, Wentau Yih, Colin Cherry, and Lucy Vanderwende. 2012. Msr splat, a language analysis toolkit. In *Proceedings of the Demonstration Session at the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 21–24, Montréal, Canada, June. Association for Computational Linguistics.
- Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 139–147, Los Angeles, June. Association for Computational Linguistics.
- Luis Von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. 2003. Captcha: Using hard ai problems for security. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer Berlin Heidelberg.

# Unsupervised joke generation from big data

Saša Petrović

School of Informatics  
University of Edinburgh  
sasa.petrovic@ed.ac.uk

David Matthews

School of Informatics  
University of Edinburgh  
dave.matthews@ed.ac.uk

## Abstract

Humor generation is a very hard problem. It is difficult to say exactly what makes a joke funny, and solving this problem algorithmically is assumed to require deep semantic understanding, as well as cultural and other contextual cues. We depart from previous work that tries to model this knowledge using ad-hoc manually created databases and labeled training examples. Instead we present a model that uses large amounts of unannotated data to generate *I like my X like I like my Y, Z* jokes, where X, Y, and Z are variables to be filled in. This is, to the best of our knowledge, the first fully unsupervised humor generation system. Our model significantly outperforms a competitive baseline and generates funny jokes 16% of the time, compared to 33% for human-generated jokes.

## 1 Introduction

Generating jokes is typically considered to be a very hard natural language problem, as it implies a deep semantic and often cultural understanding of text. We deal with generating a particular type of joke – *I like my X like I like my Y, Z* – where X and Y are nouns and Z is typically an attribute that describes X and Y. An example of such a joke is *I like my men like I like my tea, hot and British* – these jokes are very popular online.

While this particular type of joke is not interesting from a purely generational point of view (the syntactic structure is fixed), the content selection problem is very challenging. Indeed, most of the X, Y, and Z triples, when used in the context of this joke, will not be considered funny. Thus, the main challenge in this work is to “fill in” the slots in the joke template in a way that the whole phrase is considered funny.

Unlike the previous work in humor generation, we do not rely on labeled training data or hand-coded rules, but instead on large quantities of unannotated data. We present a machine learning model that expresses our assumptions about what makes these types of jokes funny and show that by using this fairly simple model and large quantities of data, we are able to generate jokes that are considered funny by human raters in 16% of cases.

The main contribution of this paper is, to the best of our knowledge, the first fully unsupervised joke generation system. We rely only on large quantities of unlabeled data, suggesting that generating jokes does not always require deep semantic understanding, as usually thought.

## 2 Related Work

Related work on computational humor can be divided into two classes: humor recognition and humor generation. Humor recognition includes double entendre identification in the form of *That’s what she said* jokes (Kiddon and Brun, 2011), sarcastic sentence identification (Davidov et al., 2010), and one-liner joke recognition (Mihalcea and Strapparava, 2005). All this previous work uses labeled training data. Kiddon and Brun (2011) use a supervised classifier (SVM) trained on 4,000 labeled examples, while Davidov et al. (2010) and Mihalcea and Strapparava (2005) both use a small amount of training data followed by a bootstrapping step to gather more.

Examples of work on humor generation include dirty joke telling robots (Sjöbergh and Araki, 2008), a generative model of two-liner jokes (Labutov and Lipson, 2012), and a model of punning riddles (Binsted and Ritchie, 1994). Again, all this work uses supervision in some form: Sjöbergh and Araki (2008) use only human jokes collected from various sources, Labutov and Lipson (2012) use a supervised approach to learn feasible circuits that connect two concepts in a semantic network, and

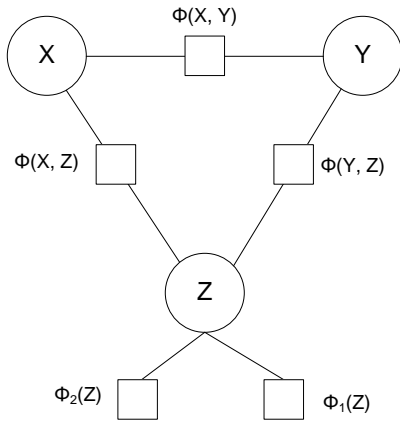


Figure 1: Our model presented as a factor graph.

Binsted and Ritchie (1994) have a set of six hard-coded rules for generating puns.

### 3 Generating jokes

We generate jokes of the form *I like my X like I like my Y, Z*, and we assume that  $X$  and  $Y$  are nouns, and that  $Z$  is an adjective.

#### 3.1 Model

Our model encodes four main assumptions about *I like my* jokes: i) a joke is funnier the more often the attribute is used to describe both nouns, ii) a joke is funnier the less common the attribute is, iii) a joke is funnier the more ambiguous the attribute is, and iv) a joke is funnier the more dissimilar the two nouns are. A graphical representation of our model in the form of a factor graph is shown in Figure 1. Variables, denoted by circles, and factors, denoted by squares, define potential functions involving the variables they are connected to.

Assumption i) is the most straightforward, and is expressed through  $\phi(X, Z)$  and  $\phi(Y, Z)$  factors. Mathematically, this assumption is expressed as:

$$\phi(x, z) = p(x, z) = \frac{f(x, z)}{\sum_{x, z} f(x, z)}, \quad (1)$$

where  $f(x, z)$ <sup>1</sup> is a function that measures the co-occurrence between  $x$  and  $z$ . In this work we simply use frequency of co-occurrence of  $x$  and  $z$  in some large corpus, but other functions, e.g., TF-IDF weighted frequency, could also be used. The same formula is used for  $\phi(Y, Z)$ , only with different variables. Because this factor measures the

<sup>1</sup>We use uppercase to denote random variables, and lowercase to denote random variables taking on a specific value.

similarity between nouns and attributes, we will also refer to it as *noun-attribute similarity*.

Assumption ii) says that jokes are funnier if the attribute used is less common. For example, there are a few attributes that are very common and can be used to describe almost anything (e.g., new, free, good), but using them would probably lead to bad jokes. We posit that the less common the attribute  $Z$  is, the more likely it is to lead to surprisal, which is known to contribute to the funniness of jokes. We express this assumption in the factor  $\phi_1(Z)$ :

$$\phi_1(z) = 1/f(z) \quad (2)$$

where  $f(z)$  is the number of times attribute  $z$  appears in some external corpus. We will refer to this factor as *attribute surprisal*.

Assumption iii) says that more ambiguous attributes lead to funnier jokes. This is based on the observation that the humor often stems from the fact that the attribute is used in one sense when describing noun  $x$ , and in a different sense when describing noun  $y$ . This assumption is expressed in  $\phi_2(Z)$  as:

$$\phi_2(z) = 1/senses(z) \quad (3)$$

where  $senses(z)$  is the number of different senses that attribute  $z$  has. Note that this does not exactly capture the fact that  $z$  should be used in different senses for the different nouns, but it is a reasonable first approximation. We refer to this factor as *attribute ambiguity*.

Finally, assumption iv) says that dissimilar nouns lead to funnier jokes. For example, if the two nouns are *girls* and *boys*, we could easily find many attributes that both nouns share. However, since the two nouns are very similar, the effect of surprisal would diminish as the observer would expect us to find an attribute that can describe both nouns well. We therefore use  $\phi(X, Y)$  to encourage dissimilarity between the two nouns:

$$\phi(x, y) = 1/sim(x, y), \quad (4)$$

where  $sim$  is a similarity function that measures how similar nouns  $x$  and  $y$  are. We call this factor *noun dissimilarity*. There are many similarity functions proposed in the literature, see e.g., Weeds et al. (2004); we use the cosine between the distributional representation of the nouns:

$$sim(x, y) = \frac{\sum_z p(z|x)p(z|y)}{\sqrt{\sum_z p(z|x)^2 * \sum_z p(z|y)^2}} \quad (5)$$

Equation 5 computes the similarity between the nouns by representing them in the space of all attributes used to describe them, and then taking the cosine of the angle between the noun vectors in this representation.

To obtain the joint probability for an  $(x, y, z)$  triple we simply multiply all the factors and normalize over all the triples.

## 4 Data

For estimating  $f(x, y)$  and  $f(z)$ , we use Google n-gram data (Michel et al., 2010), in particular the Google 2-grams. We tag each word in the 2-grams with the part-of-speech (POS) tag that corresponds to the most common POS tag associated with that word in Wordnet (Fellbaum, 1998). Once we have the POS-tagged Google 2-gram data, we extract all (noun, adjective) pairs and use their counts to estimate both  $f(x, z)$  and  $f(y, z)$ . We discard 2-grams whose count in the Google data is less than 1000. After filtering we are left with 2 million (noun, adjective) pairs. We estimate  $f(z)$  by summing the counts of all Google 2-grams that contain that particular  $z$ . We obtain  $senses(z)$  from Wordnet, which contains the number of senses for all common words.

It is important to emphasize here that, while we do use Wordnet in our work, our approach does not crucially rely on it, and we use it to obtain only very shallow information. In particular, we use Wordnet to obtain i) POS tags for Google 2-grams, and ii) number of senses for adjectives. POS tagging could be easily done using any one of the readily available POS taggers, but we chose this approach for its simplicity and speed. The number of different word senses for adjectives is harder to obtain without Wordnet, but this is only one of the four factors in our model, and we do not depend crucially on it.

## 5 Experiments

We evaluate our model in two stages. Firstly, using automatic evaluation with a set of jokes collected from Twitter, and secondly, by comparing our approach to human-generated jokes.

### 5.1 Inference

As the focus of this paper is on the model, not the inference methods, we use exact inference. While this is too expensive for estimating the true probability of any  $(x, y, z)$  triple, it is feasible if we fix

one of the nouns, i.e., if we deal with  $P(Y, Z|X = x)$ . Note that this is only a limitation of our inference procedure, not the model, and future work will look at other ways (e.g., Gibbs sampling) to perform inference. However, generating  $Y$  and  $Z$  given  $X$ , such that the joke is funny, is still a formidable challenge that a lot of humans are not able to perform successfully (cf. performance of human-generated jokes in Table 2).

### 5.2 Automatic evaluation

In the automatic evaluation we measure the effect of the different factors in the model, as laid out in Section 3.1. We use two metrics for this evaluation. The first is similar to log-likelihood, i.e., the log of the probability that our model assigns to a triple. However, because we do not compute it on all the data, just on the data that contains the  $X$ s from our development set, it is not exactly equal to the log-likelihood. It is a local approximation to log-likelihood, and we therefore dub it L<sup>O</sup>cal Log-likelihood, or LOL-likelihood for short. Our second metric computes the rank of the human-generated jokes in the distribution of all possible jokes sorted decreasingly by their LOL-likelihood. This Rank OF Likelihood (ROFL) is computed relative to the number of all possible jokes, and like LOL-likelihood is averaged over all the jokes in our development data. One advantage of ROFL is that it is designed with the way we generate jokes in mind (cf. Section 5.3), and thus more directly measures the quality of generated jokes than LOL-likelihood. For measuring LOL-likelihood and ROFL we use a set of 48 jokes randomly sampled from Twitter that fit the *I like my X like I like my Y, Z* pattern.

Table 1 shows the effect of the different factors on the two metrics. We use a model with only noun-attribute similarity (factors  $\phi(X, Z)$  and  $\phi(Y, Z)$ ) as the baseline. We see that the single biggest improvement comes from the attribute surprisal factor, i.e., from using rarer attributes. The best combination of the factors, according to automatic metrics, is using all factors except for the noun similarity (*Model 1*), while using all the factors is the second best combination (*Model 2*).

### 5.3 Human evaluation

The main evaluation of our model is in terms of human ratings, put simply: do humans find the jokes generated by our model funny? We compare four models: the two best models from Section 5.2

| Model  | LOL-likelihood | ROFL          |
|--|----------------|---------------|
| Baseline                                     | -225.3         | 0.1909        |
| Baseline + $\phi(X, Y)$                      | -227.1         | 0.2431        |
| Baseline + $\phi_1(Z)$                       | -204.9         | 0.1467        |
| Baseline + $\phi_2(Z)$                       | -224.6         | 0.1625        |
| Baseline + $\phi_1(Z) + \phi_2(Z)$ (Model 1) | <b>-198.6</b>  | <b>0.1002</b> |
| All factors (Model 2)                        | -203.7         | 0.1267        |

Table 1: Effect of different factors.

(one that uses all the factors (*Model 2*), and one that uses all factors except for the noun dissimilarity (*Model 1*)), a baseline model that uses only the noun-attribute similarity, and jokes generated by humans, collected from Twitter. We sample a further 32 jokes from Twitter, making sure that there was no overlap with the development set.

To generate a joke for a particular  $x$  we keep the top  $n$  most probable jokes according to the model, renormalize their probabilities so they sum to one, and sample from this reduced distribution. This allows our model to focus on the jokes that it considers “funny”. In our experiments, we use  $n = 30$ , which ensures that we can still generate a variety of jokes for any given  $x$ .

In our experiments we showed five native English speakers the jokes from all the systems in a random, per rater, order. The raters were asked to score each joke on a 3-point Likert scale: 1 (funny), 2 (somewhat funny), and 3 (not funny). Naturally, the raters did not know which approach each joke was coming from. Our model was used to sample  $Y$  and  $Z$  variables, given the same  $X$ s used in the jokes collected from Twitter.

Results are shown in Table 2. The second column shows the inter-rater agreement (Randolph, 2005), and we can see that it is generally good, but that it is lower on the set of human jokes. We inspected the human-generated jokes with high disagreement and found that the disagreement may be partly explained by raters missing cultural references in the jokes (e.g., a *sonic screwdriver* is Doctor Who’s tool of choice, which might be lost on those who are not familiar with the show). We do not explicitly model cultural references, and are thus less likely to generate such jokes, leading to higher agreement. The third column shows the mean joke score (lower is better), and we can see that human-generated jokes were rated the funniest, jokes from the baseline model the least funny, and that the model which uses all the

| Model       | $\kappa$ | Mean | % funny jokes |
|-------------|----------|------|---------------|
| Human jokes | 0.31     | 2.09 | 33.1          |
| Baseline    | 0.58     | 2.78 | 3.7           |
| Model 1     | 0.52     | 2.71 | 6.3           |
| Model 2     | 0.58     | 2.56 | 16.3          |

Table 2: Comparison of different models on the task of generating  $Y$  and  $Z$  given  $X$ .

factors (*Model 2*) outperforms the model that was best according to the automatic evaluation (*Model 1*). Finally, the last column shows the percentage of jokes the raters scored as funny (i.e., the number of *funny* scores divided by the total number of scores). This is a metric that we are ultimately interested in – telling a joke that is somewhat funny is not useful, and we should only reward generating a joke that is found genuinely funny by humans. The last column shows that human-generated jokes are considered funnier than the machine-generated ones, but also that our model with all the factors does much better than the other two models. *Model 2* is significantly better than the baseline at  $p = 0.05$  using a sign test, and human jokes are significantly better than all three models at  $p = 0.05$  (because we were testing multiple hypotheses, we employed Holm-Bonferroni correction (Holm, 1979)). In the end, our best model generated jokes that were found funny by humans in 16% of cases, compared to 33% obtained by human-generated jokes.

Finally, we note that the funny jokes generated by our system are not simply repeats of the human jokes, but entirely new ones that we were not able to find anywhere online. Examples of the funny jokes generated by *Model 2* are shown in Table 3.

## 6 Conclusion

We have presented a fully unsupervised humor generation system for generating jokes of the type

---

I like my relationships like I like my source, open  
I like my coffee like I like my war, cold  
I like my boys like I like my sectors, bad

---

Table 3: Example jokes generated by Model 2.

*I like my X like I like my Y, Z*, where X, Y, and Z are slots to be filled in. To the best of our knowledge, this is the first humor generation system that does not require any labeled data or hard-coded rules. We express our assumptions about what makes a joke funny as a machine learning model and show that by estimating its parameters on large quantities of unlabeled data we can generate jokes that are found funny by humans. While our experiments show that human-generated jokes are funnier more of the time, our model significantly improves upon a non-trivial baseline, and we believe that the fact that humans found jokes generated by our model funny 16% of the time is encouraging.

### Acknowledgements

The authors would like to thank the raters for their help and patience in labeling the (often not so funny) jokes. We would also like to thank Micha Elsner for this helpful comments. Finally, we thank the inhabitants of offices 3.48 and 3.38 for putting up with our sniggering every Friday afternoon.

### References

- Kim Binsted and Graeme Ritchie. 1994. An implemented model of punning riddles. In *Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, AAAI '94, pages 633–638, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116.
- Christiane Fellbaum. 1998. *Wordnet: an electronic lexical database*. MIT Press.
- Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.
- Chloé Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies: short papers - Volume 2*, pages 89–94.
- Igor Labutov and Hod Lipson. 2012. Humor as circuits in semantic networks. In *Proceedings of the 50th Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 150–155, July.
- Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Holberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2010. Quantitative analysis of culture using millions of digitized books. *Science*.
- Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: investigations in automatic humor recognition. In *Proceedings of the conference on Human Language Technology and EMNLP*, pages 531–538.
- Justus J. Randolph. 2005. Free-marginal multirater kappa (multirater free): An alternative to fleiss fixed-marginal multirater kappa. In *Joensuu University Learning and Instruction Symposium*.
- Jonas Sjöbergh and Kenji Araki. 2008. A complete and modestly funny system for generating and performing japanese stand-up comedy. In *Coling 2008: Companion volume: Posters*, pages 111–114, Manchester, UK, August. Coling 2008 Organizing Committee.
- Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*, Stroudsburg, PA, USA. Association for Computational Linguistics.



# Modeling of term-distance and term-occurrence information for improving n-gram language model performance

Tze Yuang Chong<sup>1,2</sup>, Rafael E. Banchs<sup>3</sup>, Eng Siong Chng<sup>1,2</sup>, Haizhou Li<sup>1,2,3</sup>

<sup>1</sup>Temasek Laboratory, Nanyang Technological University, Singapore 639798

<sup>2</sup>School of Computer Engineering, Nanyang Technological University, Singapore 639798

<sup>3</sup>Institute for Infocomm Research, Singapore 138632

tychong@ntu.edu.sg, rembanchs@i2r.a-star.edu.sg,  
aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg

## Abstract

In this paper, we explore the use of distance and co-occurrence information of word-pairs for language modeling. We attempt to extract this information from history-contexts of up to ten words in size, and found it complements well the  $n$ -gram model, which inherently suffers from data scarcity in learning long history-contexts. Evaluated on the WSJ corpus, bigram and trigram model perplexity were reduced up to 23.5% and 14.0%, respectively. Compared to the distant bigram, we show that word-pairs can be more effectively modeled in terms of both distance and occurrence.

## 1 Introduction

Language models have been extensively studied in natural language processing. The role of a language model is to measure how probably a (target) word would occur based on some given evidence extracted from the history-context. The commonly used  $n$ -gram model (Bahl et al. 1983) takes the immediately preceding history-word sequence, of length  $n - 1$ , as the evidence for prediction. Although  $n$ -gram models are simple and effective, modeling long history-contexts lead to severe data scarcity problems. Hence, the context length is commonly limited to as short as three, i.e. the trigram model, and any useful information beyond this window is neglected.

In this work, we explore the possibility of modeling the presence of a history-word in terms of: (1) the distance and (2) the co-occurrence, with a target-word. These two attributes will be exploited and modeled independently from each other, i.e. the distance is described regardless the actual frequency of the history-word, while the co-occurrence is described regardless the actual position of the history-word. We refer to these

two attributes as the term-distance (TD) and the term-occurrence (TO) components, respectively.

The rest of this paper is structured as follows. The following section presents the most relevant related works. Section 3 introduces and motivates our proposed approach. Section 4 presents in detail the derivation of both TD and TO model components. Section 5 presents some perplexity evaluation results. Finally, section 6 presents our conclusions and proposed future work.

## 2 Related Work

The distant bigram model (Huang et.al 1993, Simon et al. 1997, Brun et al. 2007) disassembles the  $n$ -gram into  $(n-1)$  word-pairs, such that each pair is modeled by a distance- $k$  bigram model, where  $1 \leq k \leq n - 1$ . Each distance- $k$  bigram model predicts the target-word based on the occurrence of a history-word located  $k$  positions behind.

Zhou & Lua (1998) enhanced the effectiveness of the model by filtering out those word-pairs exhibiting low correlation, so that only the well associated distant bigrams are retained. This approach is referred to as the distance-dependent trigger model, and is similar to the earlier proposed trigger model (Lau et al. 1993, Rosenfeld 1996) that relies on the bigrams of arbitrary distance, i.e. distance-independent.

Latent-semantic language model approaches (Bellegarda 1998, Coccaro 2005) weight word counts with TFIDF to highlight their semantic importance towards the prediction. In this type of approach, count statistics are accumulated from long contexts, typically beyond ten to twenty words. In order to confine the complexity introduced by such long contexts, word ordering is ignored (i.e. bag-of-words paradigm).

Other approaches such as the class-based language model (Brown 1992, Kneser & Ney 1993)

use POS or POS-like classes of the history-words for prediction. The structured language model (Chelba & Jelinek 2000) determines the “heads” in the history-context by using a parsing tree. There are also works on skipping irrelevant history-words in order to reveal more informative  $n$ -grams (Siu & Ostendorf 2000, Guthrie et al. 2006). Cache language models exploit temporal word frequencies in the history (Kuhn & Mori 1990, Clarkson & Robinson 1997).

### 3 Motivation of the Proposed Approach

The attributes of distance and co-occurrence are exploited and modeled differently in each language modeling approach. In the  $n$ -gram model, for example, these two attributes are jointly taken into account in the ordered word-sequence. Consequently, the  $n$ -gram model can only be effectively implemented within a short history-context (e.g. of size of three or four).

Both, the conventional trigger model and the latent-semantic model capture the co-occurrence information while ignoring the distance information. It is reasonable to assume that distance information at far contexts is less likely to be informative and, hence, can be discarded. However, intermediate distances beyond the  $n$ -gram model limits can be very useful and should not be discarded.

On the other hand, distant-bigram models and distance-dependent trigger models make use of both, distance and co-occurrence, information up to window sizes of ten to twenty. They achieve this by compromising inter-dependencies among history-words (i.e. the context is represented as separated word-pairs). However, similarly to  $n$ -gram models, distance and co-occurrence information are implicitly tied within the word-pairs.

In our proposed approach, we attempt to exploit the TD and TO attributes, separately, to incorporate distant context information into the  $n$ -gram, as a remedy to the data scarcity problem when learning the far context.

### 4 Language Modeling with TD and TO

A language model estimates word probabilities given their history, i.e.  $P(t = w_i | h = w_{i-1}^{i-n+1})$ , where  $t$  denotes the target word and  $h$  denotes its corresponding history. Let the word located at  $i^{\text{th}}$  position,  $w_i$ , be the target-word and its preceding word-sequence  $w_{i-1}^{i-n+1} = (w_{i-n+1} \dots w_{i-2} w_{i-1})$  of length  $n - 1$ , be its history-context. Also, in order to alleviate the data scarcity problem, we assume the occurrences of the history-words to be

independent from each other, conditioned to the occurrence of the target-word  $w_i$ , i.e.  $w_{i-k} \perp w_{i-l} | w_i$ , where  $w_{i-k}, w_{i-l} \in h$ , and  $k \neq l$ . The probability can then be approximated as:

$$P(t = w_i | h = w_{i-1}^{i-n+1}) \approx \frac{P(t = w_i) \prod_{k=1}^{n-1} P(h_k = w_{i-k} | t = w_i)}{Z(h)} \quad (1)$$

where  $Z(h)$  is a normalizing term, and  $h_k = w_{i-k}$  indicates that  $w_{i-k}$  is the word at position  $k^{\text{th}}$ .

#### 4.1 Derivation of the TD-TO Model

In order to define the TD and TO components for language modeling, we express the observation of an arbitrary history-word,  $w_{i-k}$  at the  $k^{\text{th}}$  position behind the target-word, as the joint of two events: i) the word  $w_{i-k}$  occurs within the history-context:  $w_{i-k} \in h$ , and ii) it occurs at distance  $k$  from the target-word:  $\Delta(w_{i-k}) = k$ , ( $\Delta = k$  for brevity); i.e.  $(h_k = w_{i-k}) \equiv (w_{i-k} \in h) \cap (\Delta = k)$ .

Thus, the probability in Eq.1 can be written as:

$$P(t = w_i | h = w_{i-1}^{i-n+1}) \approx \frac{P(t = w_i) \prod_{k=1}^{n-1} P(w_{i-k} \in h, \Delta = k | t = w_i)}{Z(h)} \quad (2)$$

where the likelihood  $P(w_{i-k} \in h, \Delta = k | t = w_i)$  measures how likely the joint event  $(w_{i-k} \in h, \Delta = k)$  would be observed given the target-word  $w_i$ . This can be rewritten in terms of the likelihood function of the distance event (i.e.  $\Delta = k$ ) and the occurrence event (i.e.  $w_{i-k} \in h$ ), where both of them can be modeled and exploited separately, as follows:

$$P(t = w_i | h = w_{i-1}^{i-n+1}) \approx \frac{\left[ \begin{array}{c} P(t = w_i) \\ \prod_{k=1}^{n-1} P(\Delta = k | w_{i-k} \in h, t = w_i) \\ \prod_{k=1}^{n-1} P(w_{i-k} \in h | t = w_i) \end{array} \right]}{Z(h)} \quad (3)$$

The formulation above yields three terms, referred to as the prior, the TD likelihood, and the TO likelihood, respectively.

In Eq.3, we have decoupled the observation of a word-pair into the events of distance and co-occurrence. This allows for independently modeling and exploiting them. In order to control their contributions towards the final prediction of the target-word, we weight these components:

$$P(t = w_i | h = w_{i-1}^{i-n+1}) \approx \frac{\left[ \begin{array}{c} P(t = w_i)^{\beta_n} \\ (\prod_{k=1}^{n-1} P(\Delta = k | w_{i-k} \in h, t = w_i))^{\beta_d} \\ (\prod_{k=1}^{n-1} P(w_{i-k} \in h | t = w_i))^{\beta_o} \end{array} \right]}{Z(h)} \quad (4)$$

where  $\beta_n$ ,  $\beta_d$ , and  $\beta_o$  are the weights for the prior, TD and TO models, respectively.

Notice that the model depicted in Eq.4 is the log-linear interpolation (Klakov 1998) of these models. The prior, which is usually implemented as a unigram model, can be also replaced with a higher order  $n$ -gram model as, for instance, the bigram model:

$$P(t = w_i | h = w_{i-1}^{i-n+1}) \approx \frac{\left[ \begin{array}{c} P(t = w_i | h = w_{i-1})^{\beta_n} \\ (\prod_{k=1}^{n-1} P(\Delta = k | w_{i-k} \in h, t = w_i))^{\beta_d} \\ (\prod_{k=1}^{n-1} P(w_{i-k} \in h | t = w_i))^{\beta_o} \end{array} \right]}{Z(h)} \quad (5)$$

Replacing the unigram model with a higher order  $n$ -gram model is important to compensate the damage incurred by the conditional independence assumption made earlier.

## 4.2 Term-Distance Model Component

Basically, the TD likelihood measures how likely a given word-pair would be separated by a given distance. So, word-pairs possessing consistent separation distances will favor this likelihood. The TD likelihood for a distance  $k$  given the co-occurrence of the word-pair  $(w_{i-k}, w_i)$  can be estimated from counts as follows:

$$\frac{P(\Delta = k | w_{i-k} \in h, t = w_i)}{C(w_{i-k} \in h, t = w_i, \Delta = k)} = \frac{C(w_{i-k} \in h, t = w_i)}{C(w_{i-k} \in h, t = w_i)} \quad (6)$$

The above formulation of the TD likelihood requires smoothing for resolving two problems: i) a word-pair at a particular distance has a zero count, i.e.  $C(w_{i-k} \in h, t = w_i, \Delta = k) = 0$ , which results in a zero probability, and ii) a word-pair is not seen at any distance within the observation window, i.e. zero co-occurrence  $C(w_{i-k} \in h, t = w_i) = 0$ , which results in a division by zero.

For the first problem, we have attempted to redistribute the counts among the word-pairs at different distances (as observed within the window). We assumed that the counts of word-pairs are smooth in the distance domain and that the influence of a word decays as the distance increases. Accordingly, we used a weighted moving-average filter for performing the smoothing. Similar approaches have also been used in other works (Coccaro 2005, Lv & Zhai 2009). Notice, however, that this strategy is different from other conventional smoothing techniques (Chen & Goodman 1996), which rely mainly on the count-of-count statistics for re-estimating and smoothing the original counts.

For the second problem, when a word-pair was not seen at any distance (within the window), we arbitrarily assigned a small probability value,  $P(\Delta = k | w_{i-k} \in h, t = w_i) = 0.01$ , to provide a slight chance for such a word-pair  $(w_{i-k}, w_i)$  to occur at close distances.

## 4.3 Term-Occurrence Model Component

During the decoupling operation (from Eq.2 to Eq.3), the TD model held only the distance information while the count information has been ignored. Notice the normalization of word-pair counts in Eq.6.

As a complement to the TD model, the TO model focuses on co-occurrence, and holds only count information. As the distance information is captured by the TD model, the co-occurrence count captured by the TO model is independent from the given word-pair distance.

In fact, the TO model is closely related to the trigger language model (Rosenfeld 1996), as the prediction of the target-word (the triggered word) is based on the presence of a history-word (the trigger). However, differently from the trigger model, the TO model considers all the word-pairs without filtering out the weak associated ones. Additionally, the TO model takes into account multiple co-occurrences of the same history-word within the window, while the trigger model would count them only once (i.e. considers binary counts).

The word-pairs that frequently co-occur at arbitrary distances (within an observation window) would favor the TO likelihood. It can be estimated from counts as:

$$P(w_{i-k} \in h | t = w_i) = \frac{C(w_{i-k} \in h, t = w_i)}{C(t = w_i)} \quad (7)$$

When a word-pair did not co-occur (within the observation window), we assigned a small probability value,  $P(w_{i-k} \in h | t = w_i) = 0.01$ , to provide a slight chance for the history word to occur within the history-context of the target word.

## 5 Perplexity Evaluation

A perplexity test was run on the BLLIP WSJ corpus (Charniak 2000) with the standard 5K vocabulary. The entire WSJ '87 data (740K sentences 18M words) was used as train-set to train the  $n$ -gram, TD, and TO models. The dev-set and the test-set, each comprising 500 sentences and about 12K terms, were selected randomly from WSJ '88 data. We used them for parameter fine-tuning and performance evaluation.

## 5.1 Capturing Distant Information

In this experiment, we assessed the effectiveness of the TD and TO components in reducing the  $n$ -gram’s perplexity. Following Eq.5, we interpolated  $n$ -gram models (of orders from two to six) with the TD, TO, and the both of them (referred to as TD-TO model).

By using the dev-set, optimal interpolation weights (i.e.  $\beta_n$ ,  $\beta_d$ , and  $\beta_o$ ) for the three combinations ( $n$ -gram with TD, TO, and TD-TO) were computed. The resulting interpolation weights were as follows:  $n$ -gram with TD = (0.85, 0.15),  $n$ -gram with TO = (0.85, 0.15), and  $n$ -gram with TD-TO = (0.80, 0.07, 0.13).

The history-context window sizes were optimized too. Optimal sizes resulted to be 7, 5 and 8 for TD, TO, and TD-TO models, respectively. In fact, we observed that the performance is quite robust with respect to the window’s length. Deviating about two words from the optimum length only worsens the perplexity less than 1%.

Baseline models, in each case, are standard  $n$ -gram models with modified Kneser-Ney interpolation (Chen 1996). The test-set results are depicted in Table 1.

| $N$ | NG    | NG-TD | Red. (%) | NG-TO | Red. (%) | NG-TD-TO | Red. (%) |
|-----|-------|-------|----------|-------|----------|----------|----------|
| 2   | 151.7 | 134.5 | 11.3     | 119.9 | 21.0     | 116.0    | 23.5     |
| 3   | 99.2  | 92.9  | 6.3      | 86.7  | 12.6     | 85.3     | 14.0     |
| 4   | 91.8  | 86.1  | 6.2      | 81.4  | 11.3     | 80.1     | 12.7     |
| 5   | 90.1  | 84.7  | 6.0      | 80.2  | 11.0     | 79.0     | 12.3     |
| 6   | 89.7  | 84.4  | 5.9      | 79.9  | 10.9     | 78.7     | 12.2     |

Table 1. Perplexities of the  $n$ -gram model (NG) of order ( $N$ ) two to six and their combinations with the TD, TO, and TD-TO models.

As seen from the table, for lower order  $n$ -gram models, the complementary information captured by the TD and TO components reduced the perplexity up to 23.5% and 14.0%, for bigram and trigram models, respectively. Higher order  $n$ -gram models, e.g. hexagram, observe history-contexts of similar lengths as the ones observed by the TD, TO, and TD-TO models. Due to the incapability of  $n$ -grams to model long history-contexts, the TD and TO components are still effective in helping to enhance the prediction. Similar results were obtained by using the standard back-off model (Katz 1987) as baseline.

## 5.2 Benefit of Decoupling Distant-Bigram

In this second experiment, we examined whether the proposed decoupling procedure leads to bet-

ter modeling of word-pairs compared to the distant bigram model. Here we compare the perplexity of both, the distance- $k$  bigram model and distance- $k$  TD model (for values of  $k$  ranging from two to ten), when combined with a standard bigram model.

In order to make a fair comparison, without taking into account smoothing effects, we trained both models with raw counts and evaluated their perplexities over the train-set (so that no zero-probability will be encountered). The results are depicted in Table 2.

| $k$ | 2     | 4     | 6     | 8     | 10    |
|-----|-------|-------|-------|-------|-------|
| DBG | 105.7 | 112.5 | 114.4 | 115.9 | 116.8 |
| TD  | 98.5  | 106.6 | 109.1 | 111.0 | 112.2 |

Table 2. Perplexities of the distant bigram (DBG) and TD models when interpolated with a standard bigram model.

The results from Table 2 show that the TD component complements the bigram model better than the distant bigram itself. Firstly, these results suggest that the distance information (as modeled by the TD) offers better cue than the count information (as modeled by the distant bigram) to complement the  $n$ -gram model.

The normalization of distant bigram counts, as indicated in Eq.6, aims at highlighting the information provided by the relative positions of words in the history-context. This has been shown to be an effective manner to exploit the far context. By also considering the results in Table 1, we can deduce that better performance can be obtained when the TO attribute is also involved. Overall, decoupling the word history-context into the TD and TO components offers a good approach to enhance language modeling.

## 6 Conclusions

We have proposed a new approach to compute the  $n$ -gram probabilities, based on the TD and TO model components. Evaluated on the WSJ corpus, the proposed TD and TO models reduced the bigram’s and trigram’s perplexities up to 23.5% and 14.0%, respectively. We have shown the advantages of modeling word-pairs with TD and TO, as compared to the distant bigram.

As future work, we plan to explore the usefulness of the proposed model components in actual natural language processing applications such as machine translation and speech recognition. Additionally, we also plan to develop a more principled framework for dealing with TD smoothing.

## References

- Bahl, L., Jelinek, F. & Mercer, R. 1983. A statistical approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5:179-190.
- Bellegarda, J. R. 1998. A multispans language modeling framework for large vocabulary speech recognition. *IEEE Trans. on Speech and Audio Processing*, 6(5): 456-467.
- Brown, P.F. 1992 Class-based n-gram models of natural language. *Computational Linguistics*, 18: 467-479.
- Brun, A., Langlois, D. & Smaili, K. 2007. Improving language models by using distant information. In *Proc. ISSPA 2007*, pp.1-4.
- Cavnar, W.B. & Trenkle, J.M. 1994. N-gram-based text categorization. *Proc. SDAIR-94*, pp.161-175.
- Charniak, E., et al. 2000. *BLLIP 1987-89 WSJ Corpus Release 1*. Linguistic Data Consortium, Philadelphia.
- Chen, S.F. & Goodman, J. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. ACL '96*, pp. 310-318.
- Chelba, C. & Jelinek, F. 2000. Structured language modeling. *Computer Speech & Language*, 14: 283-332.
- Clarkson, P.R. & Robinson, A.J. 1997. Language model adaptation using mixtures and an exponentially decaying cache. In *Proc. ICASSP-97*, pp.799-802.
- Coccaro, N. 2005. Latent semantic analysis as a tool to improve automatic speech recognition performance. *Doctoral Dissertation*, University of Colorado, Boulder, CO, USA.
- Guthrie, D., Allison, B., Liu, W., Guthrie, L., & Wilks, Y. 2006. A closer look at skip-gram modeling. In *Proc. LREC-2006*, pp.1222-1225.
- Huang, X. et al. 1993. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language*, 2: 137-148.
- Katz, S.M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech, & Signal Processing*, 35:400-401.
- Klakow, D. 1998. Log-linear interpolation of language model. In *Proc. ICSLP 1998*, pp.1-4.
- Kneser, R. & Ney, H. 1993. Improving clustering techniques for class-based statistical language modeling. In *Proc. EUROSPEECH '93*, pp.973-976.
- Kuhn, R. & Mori, R.D. 1990. A cache-based natural language model for speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(6): 570-583.
- Lau, R. et al. 1993. Trigger-based language models: a maximum-entropy approach. In *Proc. ICASSP-94*, pp.45-48.
- Lv Y. & Zhai C. 2009. Positional language models for information retrieval. In *Proc. SIGIR'09*, pp.299-306.
- Rosenfeld, R. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10: 187-228.
- Simons, M., Ney, H. & Martin S.C. 1997. Distant bigram language modelling using maximum entropy. In *Proc. ICASSP-97*, pp.787-790.
- Siu, M. & Ostendorf, M. 2000. Variable n-grams and extensions for conversational speech language modeling. *IEEE Trans. on Speech and Audio Processing*, 8(1): 63-75.
- Zhou G. & Lua K.T. 1998. Word association and MI-trigger-based language modeling. In *Proc. COLING-ACL*, 1465-1471.

# Discriminative Approach to Fill-in-the-Blank Quiz Generation for Language Learners

Keisuke Sakaguchi<sup>1\*</sup> Yuki Arase<sup>2</sup> Mamoru Komachi<sup>1†</sup>

<sup>1</sup>Nara Institute of Science and Technology

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

<sup>2</sup>Microsoft Research Asia

Bldg.2, No. 5 Danling St., Haidian Dist., Beijing, P. R. China

{keisuke-sa, komachi}@is.naist.jp, yukiar@microsoft.com

## Abstract

We propose discriminative methods to generate semantic distractors of fill-in-the-blank quiz for language learners using a large-scale language learners' corpus. Unlike previous studies, the proposed methods aim at satisfying both *reliability* and *validity* of generated distractors; distractors should be exclusive against answers to avoid multiple answers in one quiz, and distractors should discriminate learners' proficiency. Detailed user evaluation with 3 native and 23 non-native speakers of English shows that our methods achieve better reliability and validity than previous methods.

## 1 Introduction

Fill-in-the-blank is a popular style used for evaluating proficiency of language learners, from homework to official tests, such as TOEIC<sup>1</sup> and TOEFL<sup>2</sup>. As shown in Figure 1, a quiz is composed of 4 parts; (1) sentence, (2) blank to fill in, (3) correct answer, and (4) *distractors* (incorrect options). However, it is not easy to come up with appropriate distractors without rich experience in language education. There are two major requirements that distractors should satisfy: *reliability* and *validity* (Alderson et al., 1995). First, distractors should be *reliable*; they are exclusive against the answer and none of distractors can replace the answer to avoid allowing multiple correct answers in one quiz. Second, distractors should be *valid*; they discriminate learners' proficiency adequately.

\* This work has been done when the author was visiting Microsoft Research Asia.

† Now at Tokyo Metropolitan University (Email: komachi@tmu.ac.jp).

<sup>1</sup><http://www.ets.org/toeic>

<sup>2</sup><http://www.ets.org/toefl>

Each side, government and opposition, is \_\_\_\_\_  
the other for the political crisis, and for the violence.

(a) blaming (b) accusing (c) BOTH

Figure 1: Example of a fill-in-the-blank quiz, where (a) *blaming* is the answer and (b) *accusing* is a distractor.

There are previous studies on distractor generation for automatic fill-in-the-blank quiz generation (Mitkov et al., 2006). Hoshino and Nakagawa (2005) randomly selected distractors from words in the same document. Sumita et al. (2005) used an English thesaurus to generate distractors. Liu et al. (2005) collected distractor candidates that are close to the answer in terms of word-frequency, and ranked them by an association/collocation measure between the candidate and surrounding words in a given context. Dahlmeier and Ng (2011) generated candidates for collocation error correction for English as a Second Language (ESL) writing using paraphrasing with native language (L1) pivoting technique. This method takes a sentence containing a collocation error as input and translates it into L1, and then translate it back to English to generate correction candidates. Although the purpose is different, the technique is also applicable for distractor generation. To our best knowledge, there have not been studies that fully employed actual errors made by ESL learners for distractor generation.

In this paper, we propose automated distractor generation methods using a large-scale ESL corpus with a discriminative model. We focus on *semantically confusing* distractors that measure learners' competence to distinguish word-sense and select an appropriate word. We especially target verbs, because verbs are difficult for language learners to use correctly (Leacock et al., 2010).

Our proposed methods use discriminative models

|       |    |       |       |         |       |       |    |
|-------|----|-------|-------|---------|-------|-------|----|
| Orig. | I  | stop  |       | company | on    | today | .  |
| Corr. | I  | quit  | a     | company |       | today | .  |
| Type  | NA | #REP# | #DEL# | NA      | #INS# | NA    | NA |

Figure 2: Example of a sentence correction pair and error tags (Replacement, Deletion and Insertion).

trained on error patterns extracted from an ESL corpus, and can generate exclusive distractors with taking context of a given sentence into consideration.

We conduct human evaluation using 3 native and 23 non-native speakers of English. The result shows that 98.3% of distractors generated by our methods are reliable. Furthermore, the non-native speakers’ performance on quiz generated by our method has about 0.76 of correlation coefficient with their TOEIC scores, which shows that distractors generated by our methods satisfy validity.

Contributions of this paper are twofold; (1) we present methods for generating reliable and valid distractors, (2) we also demonstrate the effectiveness of ESL corpus and discriminative models on distractor generation.

## 2 Proposed Method

To generate distractors, we first need to decide which word to be blanked. We then generate candidates of distractors and rank them based on a certain criterion to select distractors to output.

In this section, we propose our methods for extracting target words from ESL corpus and selecting distractors by a discriminative model that considers long-distance context of a given sentence.

### 2.1 Error-Correction Pair Extraction

We use the Lang-8 Corpus of Learner English<sup>3</sup> as a large-scale ESL corpus, which consists of 1.2M sentence correction pairs. For generating semantic distractors, we regard a correction as a target and the misused word as one of the distractor candidates.

In the Lang-8 corpus, there is no clue to align the original and corrected words. In addition, words may be deleted and inserted in the corrected sentence, which makes the alignment difficult. Therefore, we detect word deletion, insertion, and replacement by dynamic programming<sup>4</sup>. We com-

<sup>3</sup><http://cl.naist.jp/nldata/lang-8/>

<sup>4</sup>The implementation is available at <https://github.com/tkyf/epair>

| Feature      | Example                                  |
|--------------|--|
| Word[i-2]    | ,  |
| Word[i-1]    | is                                       |
| Word[i+1]    | the                                      |
| Word[i+2]    | other                                    |
| Dep[i]_child | nsubj_side, aux_is, dobj_other, prep_for |
| Class        | accuse                                   |

Table 1: Example of features and class label extracted from a sentence: *Each side, government and opposition, is \*accusing/blaming the other for the political crisis, and for the violence.*

pare a corrected sentence against its original sentence, and when word insertion and deletion errors are identified, we put a placeholder (Figure 2). We then extract error-correction (i.e. replacement) pairs by comparing trigrams around the replacement in the original and corrected sentences, for considering surrounding context of the target. These error-correction pairs are a mixture of grammatical mistakes, spelling errors, and semantic confusions. Therefore, we identify pairs due to semantic confusion; we exclude grammatical error corrections by eliminating pairs whose error and correction have different part-of-speech (POS)<sup>5</sup>, and exclude spelling error corrections based on edit-distance. As a result, we extract 689 unique verbs (lemma) and 3,885 correction pairs in total.

Using the error-correction pairs, we calculate conditional probabilities  $P(w_e|w_c)$ , which represent how probable that ESL learners misuse the word  $w_c$  as  $w_e$ . Based on the probabilities, we compute a confusion matrix. The confusion matrix can generate distractors reflecting error patterns of ESL learners. Given a sentence, we identify verbs appearing in the confusion matrix and make them blank, then outputs distractor candidates that have high confusion probability. We rank the candidates by a generative model to consider the surrounding context (e.g. N-gram). We refer to this generative method as Confusion-matrix Method (CFM).

### 2.2 Discriminative Model for Distractor Generation and Selection

To generate distractors that considers long-distance context and reflects detailed syntactic information of the sentence, we train multiple classifiers for each target word using error-correction pairs extracted from ESL corpus. A classifier for

<sup>5</sup>Because the Lang-8 corpus does not have POS tags, we assign POS by the NLTK (<http://nltk.org/>) toolkit.

a target word takes a sentence (in which the target word appears) as an input and outputs a verb as the best distractor given the context using following features: 5-gram ( $\pm 1$  and  $\pm 2$  words of the target) lemmas and dependency type with the target child (lemma). The dependent is normalized when it is a pronoun, date, time, or number (e.g. *he*  $\rightarrow$  #PRP#) to avoid making feature space sparse. Table 1 shows an example of features and a class label for the classifier of a target verb (*blame*).

These classifiers are based on a discriminative model: Support Vector Machine (SVM)<sup>6</sup> (Vapnik, 1995). We propose two methods for training the classifiers.

First, we directly use the corrected sentences in the Lang-8 corpus. As shown in Table 1, we use the 5-gram and dependency features<sup>7</sup>, and use the original word (misused word by ESL learners) as a class. We refer to this method as DiscESL.

Second, we train classifiers with an ESL-simulated native corpus, because (1) the number of sentences containing a certain error-correction pair is still limited in the ESL corpus and (2) corrected sentences are still difficult to parse correctly due to inherent noise in the Lang-8 corpus. Specifically, we use articles collected from *Voice of America (VOA) Learning English*<sup>8</sup>, which consist of 270k sentences. For each target in a given sentence, we artificially change the target into an incorrect word according to the error probabilities obtained from the learners confusion matrix explained in Section 2.2. In order to collect a sufficient amount of training data, we generate 100 samples for each training sentence in which the target word is replaced into an erroneous word. We refer to this method as DiscSimESL<sup>9</sup>.

### 3 Evaluation with Native-Speakers

In this experiment, we evaluate the reliability of generated distractors. The authors asked the help of 3 native speakers of English (1 male and 2 females, majoring computer science) from an author’s graduate school. We provide each participant a gift card of \$30 as a compensation when completing the task.

<sup>6</sup>We use Linear SVM with default settings in the scikit-learn toolkit 0.13.1. <http://scikit-learn.org>

<sup>7</sup>We use the Stanford CoreNLP 1.3.4 <http://nlp.stanford.edu/software/corenlp.shtml>

<sup>8</sup><http://learningenglish.voanews.com/>

<sup>9</sup>The implementation is available at <https://github.com/keisks/disc-sim-esl>

| Method          | Corpus     | Model          |
|-----------------|------------|----------------|
| <i>Proposed</i> |            |                |
| CFM             | ESL        | Generative     |
| DiscESL         | ESL        | Discriminative |
| DiscSimESL      | Pseudo-ESL | Discriminative |
| <i>Baseline</i> |            |                |
| THM             | Native     | Generative     |
| RTM             | Native     | Generative     |

Table 2: Summary of proposed methods (CFM: Confusion Matrix Method, DiscESL: Discriminative model with ESL corpus, DiscSimESL: Discriminative model with simulated ESL corpus) and baseline (THM: Thesaurus Method, RTM: Roundtrip Method).

In order to compare distractors generated by different methods, we ask participants to solve the generated fill-in-the-blank quiz presented in Figure 1. Each quiz has 3 options: (a) only word A is correct, (b) only word B is correct, (c) both are correct. The source sentences to generate a quiz are collected from VOA, which are not included in the training dataset of the DiscSimESL. We generate 50 quizzes using different sentences per each method to avoid showing the same sentence multiple times to participants. We randomly ordered the quizzes generated by different methods for fair comparison.

We compare the proposed methods to two baselines implementing previous studies: Thesaurus-based Method (THM) and Roundtrip Translation Method (RTM). Table 2 shows a summary of each method. The THM is based on (Sumita et al., 2005) and extract distractor candidates from synonyms of the target extracted from WordNet<sup>10</sup>. The RTM is based on (Dahlmeier and Ng, 2011) and extracts distractor candidates from *roundtrip* (pivoting) translation lexicon constructed from the WIT<sup>3</sup> corpus (Cettolo et al., 2012)<sup>11</sup>, which covers a wide variety of topics. We build English-Japanese and Japanese-English word-based translation tables using GIZA++ (IBM Model4). In this dictionary, the target word is translated into Japanese words and they are translated back to English as distractor candidates. To consider (local) context, the candidates generated by the THM, RTM, and CFM are re-ranked by 5-gram language

<sup>10</sup>WordNet 3.0 <http://wordnet.princeton.edu/wordnet/>

<sup>11</sup>Available at <http://wit3.fbk.eu>



| Method          | RAD (%)                   | $\kappa$ |
|-----------------|---------------------------|----------|
| <i>Proposed</i> |                           |          |
| CFM             | 94.5 (93.1 - 96.0)        | 0.55     |
| DiscESL         | 95.0 (93.6 - 96.3)        | 0.73     |
| DiscSimESL      | <b>98.3 (97.5 - 99.1)</b> | 0.69     |
| <i>Baseline</i> |                           |          |
| THM             | 89.3 (87.4 - 91.3)        | 0.57     |
| RTM             | 93.6 (92.1 - 95.1)        | 0.53     |

Table 3: Ratio of appropriate distractors (RAD) with a 95% confidence interval and inter-rater agreement statistics  $\kappa$ .

model score trained on Google 1T Web Corpus (Brants and Franz, 2006) with IRSTLM toolkit<sup>12</sup>.

As an evaluation metric, we compute the ratio of appropriate distractors (*RAD*) by the following equation:  $RAD = N_{AD}/N_{ALL}$ , where  $N_{ALL}$  is the total number of quizzes and  $N_{AD}$  is the number of quizzes on which more than or equal to 2 participants agree by selecting the correct answer. When at least 2 participants select the option (c) (both options are correct), we determine the distractor as inappropriate. We also compute the average of inter-rater agreement  $\kappa$  among all participants for each method.

Table 3 shows the results of the first experiment; RAD with a 95% confidence interval and inter-rater agreement  $\kappa$ . All of our proposed methods outperform baselines regarding RAD with high inter-rater agreement. In particular, DiscSimESL achieves 9.0% and 4.7% higher RAD than THM and RTM, respectively. These results show that the effectiveness of using ESL corpus to generate reliable distractors. With respect to  $\kappa$ , our discriminative models achieve from 0.12 to 0.2 higher agreement than baselines, indicating that the discriminative models can generate sound distractors more effectively than generative models. The lower  $\kappa$  on generative models may be because the distractors are semantically too close to the target (correct answer) as following examples:

The coalition has *\*published/issued* a report saying that ... .

As a result, the quiz from generative models is not reliable since both *published* and *issued* are correct.

#### 4 Evaluation with ESL Learners

In this experiment, we evaluate the validity of generated distractors regarding ESL learners' profi-

<sup>12</sup>The *irstlm* toolkit 5.80 <http://sourceforge.net/projects/irstlm/files/irstlm/>

| Method          | $r$         | Corr | Dist | Both | Std  |
|-----------------|-------------|------|------|------|------|
| <i>Proposed</i> |             |      |      |      |      |
| CFM             | 0.71        | 56.7 | 29.6 | 13.5 | 11.5 |
| DiscESL         | 0.48        | 62.4 | 27.9 | 10.4 | 12.8 |
| DiscSimESL      | <b>0.76</b> | 64.0 | 20.7 | 15.1 | 13.4 |
| <i>Baseline</i> |             |      |      |      |      |
| THM             | 0.68        | 57.2 | 28.1 | 14.6 | 10.7 |
| RTM             | 0.67        | 63.4 | 26.9 | 9.5  | 13.2 |

Table 4: (1) Correlation coefficient  $r$  against participants' TOEIC scores, (2) the average percentage of correct answer (Corr), incorrect answer of distractor (Dist), and incorrect answer that both are correct (Both) chosen by participants, and (3) standard deviation (Std) of Corr.

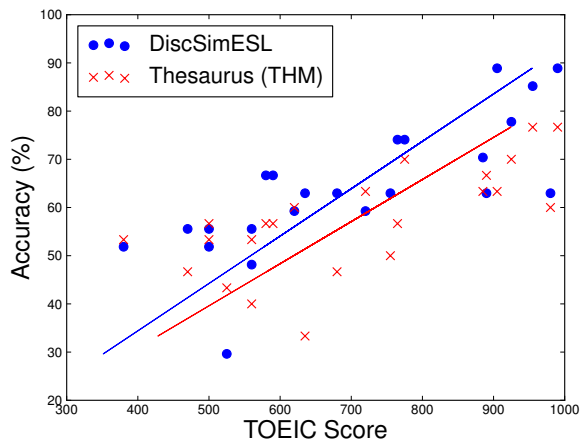


Figure 3: Correlation between the participants' TOEIC scores and accuracy on THM and DiscSimESL.

ciency. Twenty-three Japanese native speakers (15 males and 8 females) are participated. All the participants, who have taken at least 8 years of English education, self-report proficiency levels as the TOEIC scores from 380 to 990<sup>13</sup>. All the participants are graduate students majoring in science related courses. We call for participants by emailing to a graduate school. We provide each participant a gift card of \$10 as a compensation when completing the task. We ask participants to solve 20 quizzes per each method in the same manner as Section 3. To evaluate validity of distractors, we use only reliable quizzes accepted in Section 3. Namely, we exclude quizzes whose options are both correct. We evaluate correlation between learners' accuracy for the generated quizzes and the TOEIC score.

Table 4 represents the results; the highest corre-

<sup>13</sup>The official score range of the TOEIC is from 10 to 990.

lation coefficient  $r$  and standard deviation on DiscSimESL shows that its distractors achieve best validity. Figure 3 depicts the correlations between the participants' TOEIC scores and accuracy (i.e. Corr.) on THM and DiscSimESL. It illustrates that DiscSimESL achieves higher level of positive correlation than THM. Table 4 also shows high percentage of choosing "(c) both are correct" on DiscSimESL, which indicates that distractors generated from DiscSimESL are difficult to distinguish for ESL learners but not for native speakers as a following example:

..., she found herself on stage ...  
\*playing/performing a number one hit.

A relatively lower correlation coefficient on DiscESL may be caused by inherent noise on parsing the Lang-8 corpus and domain difference from quiz sentences (VOA).

## 5 Conclusion

We have presented methods that automatically generate semantic distractors of a fill-in-the-blank quiz for ESL learners. The proposed methods employ discriminative models trained using error patterns extracted from ESL corpus and can generate reliable distractors by taking context of a given sentence into consideration. The human evaluation shows that 98.3% of distractors are reliable when generated by our method (DiscSimESL). The results also demonstrate 0.76 of correlation coefficient to their TOEIC scores, indicating that the distractors have better validity than previous methods. As future work, we plan to extend our methods for other POS, such as adjective and noun. Moreover, we will take ESL learners' proficiency into account for generating distractors of appropriate levels for different learners.

## Acknowledgments

This work was supported by the Microsoft Research Collaborative Research (CORE) Projects. We are grateful to Yangyang Xi for granting permission to use text from Lang-8 and Takuya Fujino for his error pair extraction algorithm. We would also thank anonymous reviewers for valuable comments and suggestions.

## References

- Charles Alderson, Caroline Clapham, and Dianne Wall. 1995. *Language Test Construction and Evaluation*. Cambridge University Press.
- Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Corpus version 1.1. *Technical report, Google Research*.
- Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT<sup>3</sup> : Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trent, Italy, May.
- Daniel Dahlmeier and Hwee Tou Ng. 2011. Correcting semantic collocation errors with 11-induced paraphrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Edinburgh, Scotland, UK., July.
- Ayako Hoshino and Hiroshi Nakagawa. 2005. A Real-Time Multiple-Choice Question Generation for Language Testing — A Preliminary Study —. In *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 17–20, Ann Arbor, June.
- Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel R. Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Chao-Lin Liu, Chun-Hung Wang, Zhao-Ming Gao, and Shang-Ming Huang. 2005. Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items. In *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 1–8, Ann Arbor, June.
- Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. 2006. A Computer-Aided Environment for Generating Multiple-Choice Test Items. *Natural Language Engineering*, 12:177–194, 5.
- Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. Measuring Non-native Speakers' Proficiency of English by Using a Test with Automatically-Generated Fill-in-the-Blank Questions. In *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, pages 61–68, Ann Arbor, June.
- Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

# “Let Everything Turn Well in Your Wife”: Generation of Adult Humor Using Lexical Constraints

**Alessandro Valitutti**

Department of Computer Science  
and HIIT  
University of Helsinki, Finland

**Hannu Toivonen**

Department of Computer Science  
and HIIT  
University of Helsinki, Finland

**Antoine Doucet**

Normandy University – UNICAEN  
GREYC, CNRS UMR–6072  
Caen, France

**Jukka M. Toivanen**

Department of Computer Science  
and HIIT  
University of Helsinki, Finland

## Abstract

We propose a method for automated generation of adult humor by lexical replacement and present empirical evaluation results of the obtained humor. We propose three types of lexical constraints as building blocks of humorous word substitution: constraints concerning the similarity of sounds or spellings of the original word and the substitute, a constraint requiring the substitute to be a taboo word, and constraints concerning the position and context of the replacement. Empirical evidence from extensive user studies indicates that these constraints can increase the effectiveness of humor generation significantly.

## 1 Introduction

Incongruity and taboo meanings are typical ingredients of humor. When used in the proper context, the expression of contrasting or odd meanings can induce surprise, confusion or embarrassment and, thus, make people laugh. While methods from computational linguistics can be used to estimate the capability of words and phrases to induce incongruity or to evoke taboo meanings, computational generation of humorous texts has remained a great challenge.

In this paper we propose a method for automated generation of adult humor by lexical replacement. We consider a setting where a short text is provided to the system, such as an instant message, and the task is to make the text funny by replacing one word in it. Our approach is based

on careful introduction of incongruity and taboo words to induce humor.

We propose three types of lexical constraints as building blocks of humorous word substitution. (1) The *form constraints* turn the text into a pun. The constraints thus concern the similarity of sounds or spellings of the original word and the substitute. (2) The *taboo constraint* requires the substitute to be a taboo word. This is a well-known feature in some jokes. We hypothesize that the effectiveness of humorous lexical replacement can be increased with the introduction of taboo constraints. (3) Finally, the *context constraints* concern the position and context of the replacement.

Our assumption is that a suitably positioned substitution propagates the *tabooness* (defined here as the capability to evoke taboo meanings) to phrase level and amplifies the semantic contrast with the original text. Our second concrete hypothesis is that the context constraints further boost the funniness.

We evaluated the above hypotheses empirically by generating 300 modified versions of SMS messages and having each of them evaluated by 90 subjects using a crowdsourcing platform. The results show a statistically highly significant increase of funniness and agreement with the use of the humorous lexical constraints.

The rest of this paper is structured as follows. In Section 2, we give a short overview of theoretical background and related work on humor generation. In Section 3, we present the three types of constraints for lexical replacement to induce humor. The empirical evaluation is presented in Section 4. Section 5 contains concluding remarks.

## 2 Background

**Humor, Incongruity and Tabooeness** A set of theories known as *incongruity theory* is probably the most influential approach to the study of humor and laughter. The concept of incongruity, first described by Beattie (1971), is related to the perception of incoherence, semantic contrast, or inappropriateness, even though there is no precise and agreed definition. Raskin (1985) formulated the incongruity concept in terms of *script opposition*. This has been developed further, into the *General Theory of Verbal Humor* (Attardo and Raskin, 1991). A cognitive treatment of incongruity in humor is described by Summerfelt et al. (2010).

One specific form of jokes frequently discussed in the literature consists of the so called *forced reinterpretation jokes*. E.g.:

*Alcohol isn't a problem, it's a solution...  
Just ask any chemist.*

In his analysis of forced reinterpretation jokes, Ritchie (2002) emphasises the distinction between three different elements of the joke processing: CONFLICT is the initial perception of incompatibility between punchline and setup according to the initial obvious interpretation; CONTRAST denotes the perception of the contrastive connection between the two interpretations; while INAPPROPRIATENESS refers to the intrinsic oddness or tabooeness characterising the funny interpretation. All three concepts are often connected to the notion of incongruity.

In his integrative approach to humor theories, Martin (2007) discusses the connection between tabooeness and incongruity resolution. In particular, he discusses the *salience hypothesis* (Goldstein et al., 1972; Attardo and Raskin, 1991), according to which “the purpose of aggressive and sexual elements in jokes is to make salient the information needed to resolve the incongruity”.

**Humor Generation** In previous research on computational humor generation, puns are often used as the core of more complex humorous texts, for example as punchlines of simple jokes (Raskin and Attardo, 1994; Levison and Lessard, 1992; Venour, 1999; McKay, 2002). This differs from our setting, where we transform an existing short text into a punning statement.

Only few humor generation systems have been

empirically evaluated. The JAPE program (Binsted et al., 1997) produces specific types of punning riddles. HAHAcronym (Stock and Straparava, 2002) automatically generates humorous versions of existing acronyms, or produces a new funny acronym, starting with concepts provided by the user. The evaluations indicate statistical significance, but the test settings are relatively specific. Below, we will present an approach to evaluation that allows comparison of different systems in the same generation task.

## 3 Lexical Constraints for Humorous Word Substitution

The procedure gets as input a segment of English text (e.g.: “*Let everything turn well in your life!*”). Then it performs a single word substitution (e.g.: ‘*life*’ → ‘*wife*’), and returns the resulting text. To make it funny, the word replacement is performed according to a number of lexical constraints, to be described below. Additionally, the text can be appended with a phrase such as “*I mean ‘life’ not ‘wife’.*” The task of humor generation is thus reduced to a task of lexical selection. The adopted task for humor generation is an extension of the one described by Valitutti (2011).

We define three types of lexical constraints for this task, which will be described next.

### 3.1 Form Constraints

*Form constraints* (FORM) require that the original word and its substitute are similar in form. This turns the text given as input into a kind of *pun*, “text which relies crucially on phonetic similarity for its humorous effect” (Ritchie, 2005).

Obviously, simply replacing a word potentially results in a text that induces “conflict” (and confusion) in the audience. Using a phonetically similar word as a replacement, however, makes the statement pseudo-ambiguous, since the original intended meaning can also be recovered. There then are two “conflicting” and “contrasting” interpretations — the literal one and the original one — increasing the likelihood of humorous incongruity.

Requiring the substitute to share part-of-speech with the original word works in this direction too, and additionally increases the likelihood that the resulting text is a valid English statement.

**Implementation** We adopt an extended definition of punning and also consider orthographically similar or rhyming words as possible substitutes.

Two words are considered *orthographically similar* if one word is obtained with a single character deletion, addition, or replacement from the other one.

We call two words *phonetically similar* if their phonetic transcription is orthographically similar according to the above definition.

Two words *rhyme* if they have same positions of tonic accent, and if they are phonetically identical from the most stressed syllable to the end of the word.

Our implementation of these constraints uses the WordNet lexical database (Fellbaum, 1998) and CMU pronunciation dictionary<sup>1</sup>. The latter also provides a collection of words not normally contained in standard English dictionaries, but commonly used in informal language. This increases the space of potential replacements. We use the TreeTagger<sup>2</sup> POS tagger in order to consider only words with the same part-of-speech of the word to be replaced.

### 3.2 Taboo Constraint

*Taboo constraint* (TABOO) requires that the substitute word is a taboo word or frequently used in taboo expressions, insults, or vulgar expressions. Taboo words “represent a class of emotionally arousing references with respect to body products, body parts, sexual acts, ethnic or racial insults, profanity, vulgarity, slang, and scatology” (Jay et al., 2008), and they directly introduce “inappropriateness” to the text.

**Implementation** We collected a list of 700 taboo words. A first subset contains words manually selected from the domain SEXUALITY of WordNet-Domains (Magnini and Cavaglia, 2000). A second subset was collected from the Web, and contains words commonly used as insults. Finally, a third subset was collected from a website posting examples of funny autocorrection mistakes<sup>3</sup> and includes words that are not directly referring to taboos (e.g.: ‘stimulation’) or often retrieved in

<sup>1</sup>available at <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

<sup>2</sup>available at <http://www.ims.unistuttgart.de/projekte/corplex/TreeTagger>

<sup>3</sup><http://www.damnyouautocorrect.com>

jokes evoking taboo meanings (e.g.: ‘wife’).

### 3.3 Contextual Constraints

*Contextual constraints* (CONT) require that the substitution takes place at the end of the text, and in a locally coherent manner.

By local coherence we mean that the substitute word forms a feasible phrase with its immediate predecessor. If this is *not* the case, then the text is likely to make little sense. On the other hand, if this *is* the case, then the taboo meaning is potentially expanded to the phrase level. This introduces a stronger semantic “contrast” and thus probably contributes to making the text funnier. The semantic contrast is potentially even stronger if the taboo word comes as a surprise in the end of a seemingly innocent text. The humorous effect then is similar to the one of the forced reinterpretation jokes.

**Implementation** Local coherence is implemented using n-grams. In the case of languages that are read from left to right, such as English, expectations will be built by the left-context of the expected word. To estimate the level of expectation triggered by a left-context, we rely on a vast collection of n-grams, the 2012 Google Books n-grams collection<sup>4</sup> (Michel et al., 2011) and compute the cohesion of each n-gram, by comparing their expected frequency (assuming word independence), to their observed number of occurrences. A subsequent Student t-test allows to assign a measure of cohesion to each n-gram (Doucet and Ahonen-Myka, 2006). We use a substitute word only if its cohesion with the previous word is high.

In order to use consistent natural language and avoid time or location-based variations, we focused on contemporary American English. Thus we only used the subsection of Google bigrams for American English, and ignored all the statistics stemming from books published before 1990.

## 4 Evaluation

We evaluated the method empirically using CrowdFlower<sup>5</sup>, a crowdsourcing service. The aim of the evaluation is to measure the potential effect of the three types of constraints on funniness of texts. In particular, we test the potential effect of

<sup>4</sup>available at <http://books.google.com/ngrams>

<sup>5</sup>available at <http://www.crowdfunder.com>

adding the tabooess constraint to the form constraints, and the potential effect of further adding contextual constraints. I.e., we consider three increasingly constrained conditions: (1) substitution according only to the form constraints (FORM), (2) substitution according to both form and taboo constraints (FORM+TABOO), and (3) substitution according to form, taboo and context constraints (FORM+TABOO+CONT).

One of the reasons for the choice of taboo words as lexical constraint is that they allows the system to generate humorous text potentially appreciated by young adults, which are the majority of crowdsourcing users (Ross et al., 2010). We applied the humor generation method on the first 5000 messages of *NUS SMS Corpus*<sup>6</sup>, a corpus of real SMS messages (Chen and Kan, 2012).

We carried out every possible lexical replacement under each of the three conditions mentioned above, one at a time, so that the resulting messages have exactly one word substituted. We then randomly picked 100 such modified messages for each of the conditions. Table 1 shows two example outputs of the humor generator under each of the three experimental conditions. These two examples are the least funny and the funniest message according to the empirical evaluation (see below).

For evaluation, this dataset of 300 messages was randomly divided into groups of 20 messages each. We recruited 208 evaluators using the crowdsourcing service, asking each subject to evaluate one such group of 20 messages. Each message in each group was judged by 90 different participants.

We asked subjects to assess individual messages for their funniness on a scale from 1 to 5. For the analysis of the results, we then measured the effectiveness of the constraints using two derived variables: the *Collective Funniness* (CF) of a message is its mean funniness, while its *Upper Agreement* (UA( $t$ )) is the fraction of funniness scores greater than or equal to a given threshold  $t$ . To rank the generated messages, we take the product of Collective Funniness and Upper Agreement UA(3) and call it the overall *Humor Effectiveness* (HE).

In order to identify and remove potential scammers in the crowdsourcing system, we simply asked subjects to select the last word in the mes-

sage. If a subject failed to answer correctly more than three times all her judgements were removed. As a result, 2% of judgments were discarded as untrusted. From the experiment, we then have a total of 26 534 trusted assessments of messages, 8 400 under FORM condition, 8 551 under FORM+TABOO condition, and 8 633 under FORM+TABOO+CONT condition.

The Collective Funniness of messages increases, on average, from 2.29 under condition FORM to 2.98 when the taboo constraint is added (FORM+TABOO), and further to 3.20 when the contextual constraints are added (FORM+TABOO+CONT) (Table 2). The Upper Agreement UA(4) increases from 0.18 to 0.36 and to 0.43, respectively.

We analyzed the distributions of Collective Funniness values of messages, as well as the distributions of their Upper Agreements (for all values from UA(2) to UA(5)) under the three conditions. According to the one-sided Wilcoxon rank-sum test, both Collective Funniness and all Upper Agreements increase from FORM to FORM+TABOO and from FORM+TABOO to FORM+TABOO+CONT statistically significantly (in all cases  $p < .002$ ). Table 3 shows  $p$ -values associated with all pairwise comparisons.

## 5 Conclusions

We have proposed a new approach for the study of computational humor generation by lexical replacement. The generation task is based on a simple form of punning, where a given text is modified by replacing one word with a similar one.

We proved empirically that, in this setting, humor generation is more effective when using a list of taboo words. The other strong empirical result regards the context of substitutions: using bigrams to model people's expectations, and constraining the position of word replacement to the end of the text, increases funniness significantly. This is likely because of the form of surprise they induce. At best of our knowledge, this is the first time that these aspects of humor generation have been successfully evaluated with a crowdsourcing system and, thus, in a relatively quick and economical way.

The statistical significance is particularly high, even though there were several limitations in the experimental setting. For example, as explained in Section 3.2, the employed word list was built

<sup>6</sup>available at <http://wing.comp.nus.edu.sg/SMSCorpus>

| Experimental Condition | Text Generated by the System  | CF   | UA(3) | HE   |
|------------------------|---|------|-------|------|
| FORM                   | Oh oh...Den muz change plat liao...Go back have yan jiu again...<br>Not 'plat'...'plan'.                              | 1.68 | 0.26  | 0.43 |
| FORM                   | Jos ask if u wana melt up? 'meet' not 'melt'!   | 2.96 | 0.74  | 2.19 |
| FORM+TABOO             | Got caught in the rain.Waited half n hour in the buss stop.<br>Not 'buss'...'bus'!                                    | 2.06 | 0.31  | 0.64 |
| BASE+TABOO             | Hey pple... \$ 700 or \$ 900 for 5 nights...Excellent masturbation<br>wif breakfast hamper!!! Sorry I mean 'location' | 3.98 | 0.85  | 3.39 |
| FORM+TABOO+CONT        | Nope...Juz off from berk... Sorry I mean 'work'   | 2.25 | 0.39  | 0.87 |
| FORM+TABOO+CONT        | I've sent you my fart.. I mean 'part' not 'fart'...   | 4.09 | 0.90  | 3.66 |

Table 1: Examples of outputs of the system. CF: Collective Funniness; UA(3): Upper Agreement; HE: Humor Effectiveness.

|       | Experimental Conditions |                 |                 |
|-------|-------------------------|-----------------|-----------------|
|       | FORM                    | FORM+TABOO      | FORM+TABOO+CONT |
| CF    | $2.29 \pm 0.19$         | $2.98 \pm 0.43$ | $3.20 \pm 0.40$ |
| UA(2) | $0.58 \pm 0.09$         | $0.78 \pm 0.11$ | $0.83 \pm 0.09$ |
| UA(3) | $0.41 \pm 0.07$         | $0.62 \pm 0.13$ | $0.69 \pm 0.12$ |
| UA(4) | $0.18 \pm 0.04$         | $0.36 \pm 0.13$ | $0.43 \pm 0.13$ |
| UA(5) | $0.12 \pm 0.02$         | $0.22 \pm 0.09$ | $0.26 \pm 0.09$ |

Table 2: Mean Collective Funniness (CF) and Upper Agreements (UA(·)) under the three experimental conditions and their standard deviations.

|       | Hypotheses                    |  |
|-------|-------------------------------|--|
|       | FORM $\rightarrow$ FORM+TABOO | FORM+TABOO $\rightarrow$ FORM+TABOO+CONT |
| CF    | $10^{-15}$                    | $9 \times 10^{-5}$                       |
| UA(2) | $10^{-15}$                    | $1 \times 10^{-15}$                      |
| UA(3) | $10^{-15}$                    | $7 \times 10^{-5}$                       |
| UA(4) | $10^{-15}$                    | $2 \times 10^{-4}$                       |
| UA(5) | $10^{-15}$                    | $2 \times 10^{-3}$                       |

Table 3: P-values resulting from the application of one-sided Wilcoxon rank-sum test.

from different sources and contains words not directly referring to taboo meanings and, thus, not widely recognizable as “taboo words”. Furthermore, the possible presence of crowd-working scammers (only partially filtered by the gold standard questions) could have reduced the statistical power of our analysis. Finally, the adopted humor generation task (based on a single word substitution) is extremely simple and the constraints might have not been sufficiently capable to produce a detectable increase of humor appreciation.

The statistically strong results that we obtained can make this evaluation approach attractive for related tasks. In our methodology, we focused attention to the correlation between the parameters of the system (in our case, the constraints used in lexical selection) and the performance of humor generation. We used a multi-dimensional measure of humorous effect (in terms of funniness and agreement) to measure subtly different aspects of the humorous response. We then adopted a comparative setting, where we can measure improve-

ments in the performance across different systems or variants.

In the future, it would be interesting to use a similar setting to empirically investigate more subtle ways to generate humor, potentially with weaker effects but still recognizable in this setting. For instance, we would like to investigate the use of other word lists besides taboo domains and the extent to which the semantic relatedness itself could contribute to the humorous effect.

The current techniques can be improved, too, in various ways. In particular, we plan to extend the use of n-grams to larger contexts and consider more fine-grained tuning of other constraints, too. One goal is to apply the proposed methodology to isolate, on one hand, parameters for inducing incongruity and, on the other hand, parameters for making the incongruity funny.

Finally, we are interested in estimating the probability to induce a humor response by using different constraints. This would offer a novel way to intentionally control the humorous effect.

## References

- S. Attardo and V. Raskin. 1991. Script theory revis(it)ed: joke similarity and joke representation model. *Humour*, 4(3):293–347.
- J. Beattie. 1971. An essay on laughter, and ludicrous composition. In *Essays*. William Creech, Edinburgh, 1776. Reprinted by Garland, New York.
- K. Binsted, H. Pain, and G. Ritchie. 1997. Children’s evaluation of computer-generated punning riddles. *Pragmatics and Cognition*, 2(5):305–354.
- T. Chen and M.-Y. Kan. 2012. Creating a live, public short message service corpus: The nus sms corpus. *Language Resources and Evaluation*, August. published online.
- A. Doucet and H. Ahonen-Myka. 2006. Probability and expected document frequency of discontinued word sequences, an efficient method for their exact computation. *Traitement Automatique des Langues (TAL)*, 46(2):13–37.
- C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.
- J. H. Goldstein, J. M. Suls, and S. Anthony. 1972. Enjoyment of specific types of humor content: Motivation or salience? In J. H. Goldstein and P. E. McGhee, editors, *The psychology of humor: Theoretical perspectives and empirical issues*, pages 159–171. Academic Press, New York.
- T. Jay, C. Caldwell-Harris, and K. King. 2008. Recalling taboo and nontaboo words. *American Journal of Psychology*, 121(1):83–103, Spring.
- M. Levison and G. Lessard. 1992. A system for natural language generation. *Computers and the Humanities*, 26:43–58.
- B. Magnini and G. Cavaglià. 2000. Integrating subject field codes into wordnet. In *Proc. of the 2<sup>nd</sup> International Conference on Language Resources and Evaluation (LREC2000)*, Athens, Greece.
- R. A. Martin. 2007. *The Psychology of Humor: An Integrative Approach*. Elsevier.
- J. McKay. 2002. Generation of idiom-based witticisms to aid second language learning. In *(Stock et al., 2002)*.
- J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, The Google Books Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.
- V. Raskin and S. Attardo. 1994. Non-literality and non-bona-fide in language: approaches to formal and computational treatments of humor. *Pragmatics and Cognition*, 2(1):31–69.
- V. Raskin. 1985. *Semantic Mechanisms of Humor*. Dordrecht/Boston/Lancaster.
- G. Ritchie. 2002. The structure of forced interpretation jokes. In *(Stock et al., 2002)*.
- G. Ritchie. 2005. Computational mechanisms for pun generation. In *Proceedings of the 10th European Natural Language Generation Workshop*, Aberdeen, August.
- J. Ross, I. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson. 2010. Who are the crowdworkers?: Shifting demographics in amazon mechanical turk. In *Proc. of the ACM CHI Conference*.
- O. Stock and C. Strapparava. 2002. HAHAcronym: Humorous agents for humorous acronyms. In *(Stock et al., 2002)*.
- O. Stock, C. Strapparava, and A. Nijholt, editors. 2002. *Proceedings of the The April Fools Day Workshop on Computational Humour (TWLT20)*, Trento.
- H. Summerfelt, L. Lippman, and I. E. Hyman Jr. 2010. The effect of humor on memory: Constrained by the pun. *The Journal of General Psychology*, 137(4):376–394.
- A. Valitutti. 2011. How many jokes are really funny? towards a new approach to the evaluation of computational humour generators. In *Proc. of 8th International Workshop on Natural Language Processing and Cognitive Science*, Copenhagen.
- C. Venour. 1999. The computational generation of a class of puns. Master’s thesis, Queen’s University, Kingston, Ontario.



# Random Walk Factoid Annotation for Collective Discourse

**Ben King**     **Rahul Jha**  
Department of EECS  
University of Michigan  
Ann Arbor, MI  
benking@umich.edu  
rahuljha@umich.edu

**Dragomir R. Radev**  
Department of EECS  
School of Information  
University of Michigan  
Ann Arbor, MI  
radev@umich.edu

**Robert Mankoff** \*  
The New Yorker Magazine  
New York, NY  
bob.mankoff  
@newyorker.com

## Abstract

In this paper, we study the problem of automatically annotating the factoids present in collective discourse. Factoids are information units that are shared between instances of collective discourse and may have many different ways of being realized in words. Our approach divides this problem into two steps, using a graph-based approach for each step: (1) factoid discovery, finding groups of words that correspond to the same factoid, and (2) factoid assignment, using these groups of words to mark collective discourse units that contain the respective factoids. We study this on two novel data sets: the New Yorker caption contest data set, and the crossword clues data set.

## 1 Introduction

Collective discourse tends to contain relatively few *factoids*, or information units about which the author speaks, but many *nuggets*, different ways to speak about or refer to a factoid (Qazvinian and Radev, 2011). Many natural language applications could be improved with good factoid annotation.

Our approach in this paper divides this problem into two subtasks: discovery of factoids, and assignment of factoids. We take a graph-based approach to the problem, clustering a word graph to discover factoids and using random walks to assign factoids to discourse units.

We also introduce two new datasets in this paper, covered in more detail in section 3. The New Yorker cartoon caption dataset, provided by Robert Mankoff, the cartoon editor at The New Yorker magazine, is composed of reader-submitted captions for a cartoon published in the magazine. The crossword clue dataset consists

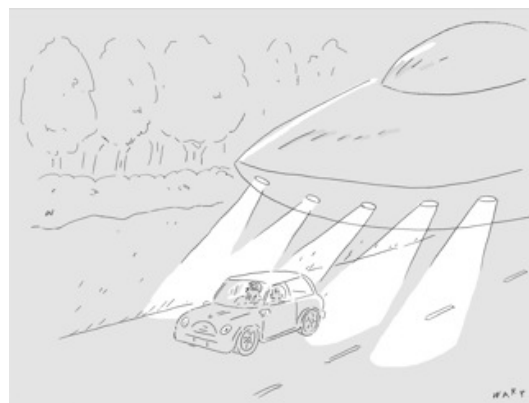


Figure 1: The cartoon used for the New Yorker caption contest #331.

of word-clue pairs used in major American crossword puzzles, with most words having several hundred different clues published for it.

The term “factoid” is used as in (Van Halteren and Teufel, 2003), but in a slightly more abstract sense in this paper, denoting a set of related words that should ideally refer to a real-world entity, but may not for some of the less coherent factoids. The factoids discovered using this method don’t necessarily correspond to the factoids that might be chosen by annotators.

For example, given two user-submitted cartoon captions

- “When they said, ‘Take us to your leader,’ I don’t think they meant your mother’s house,”
- and “You’d better call your mother and tell her to set a few extra place settings,”

a human may say that they share the factoid called “mother.” The automatic methods however, might say that these captions share *factoid3*, which is identified by the words “mother,” “in-laws,” “family,” “house,” etc.

The layout of this paper is as follows: we review related work in section 2, we introduce the datasets

\* Cartoon Editor, The New Yorker magazine

in detail in section 3, we describe our methods in section 4, and report results in section 5.

## 2 Related Work

The distribution of factoids present in text collections is important for several NLP tasks such as summarization. The Pyramid Evaluation method (Nenkova and Passonneau, 2004) for automatic summary evaluation depends on finding and annotating factoids in input sentences. Qazvinian and Radev (2011) also studied the properties of factoids present in collective human datasets and used it to create a summarization system. Hennig et al. (2010) describe an approach for automatically learning factoids for pyramid evaluation using a topic modeling approach.

Our random-walk annotation technique is similar to the one used in (Hassan and Radev, 2010) to identify the semantic polarity of words. Das and Petrov (2011) also introduced a graph-based method for part-of-speech tagging in which edge weights are based on feature vectors similarity, which is like the corpus-based lexical similarity graph that we construct.

## 3 Data Sets

We introduce two new data sets in this paper, the New Yorker caption contest data set, and the crossword clues data set. Though these two data sets are quite different, they share a few important characteristics. First, the discourse units tend to be short, approximately ten words for cartoon captions and approximately three words for crossword clues. Second, though the authors act independently, they tend to produce surprisingly similar text, making the same sorts of jokes, or referring to words in the same sorts of ways. Thirdly, the authors often try to be non-obvious: obvious jokes are often not funny, and obvious crossword clues make a puzzle less challenging.

### 3.1 New Yorker Caption Contest Data Set

The New Yorker magazine holds a weekly contest<sup>1</sup> in which they publish a cartoon without a caption and solicit caption suggestions from their readers. The three funniest captions are selected by the editor and published in the following weeks. Figure 1 shows an example of such a cartoon, while Table 1 shows examples of captions, including its winning captions. As part of

<sup>1</sup><http://www.newyorker.com/humor/caption>

---

|  |
|--|
| <i>I don't care what planet they are from, they can pass on the left like everyone else.</i>     |
| I don't care what planet they're from, they should have the common courtesy to dim their lights. |
| I don't care where he's from, you pass on the left.  |
| If he wants to pass, he can use the right lane like everyone else.                               |
| <i>When they said, 'Take us to your leader,' I don't think they meant your mother's house.</i>   |
| They may be disappointed when they learn that "our leader" is your mother.                       |
| You'd better call your mother and tell her to set a few extra place settings.                    |
| If they ask for our leader, is it Obama or your mother?  |
| <i>Which finger do I use for aliens?</i>   |
| I guess the middle finger means the same thing to them.  |
| I sense somehow that flipping the bird was lost on them.   |
| What's the Klingon gesture for "Go around us, jerk?"   |

---

Table 1: Captions for contest #331. Finalists are listed in italics.

this research project, we have acquired five cartoons along with all of the captions submitted in the corresponding contest.

While the task of automatically identifying the funny captions would be quite useful, it is well beyond the current state of the art in NLP. A much more manageable task, and one that is quite important for the contest's editor is to annotate captions according to their factoids. This allows the organizers of the contest to find the most frequently mentioned factoids and select representative captions for each factoid.

On average, each cartoon has 5,400 submitted captions, but for each of five cartoons, we sampled 500 captions for annotation. The annotators were instructed to mark factoids by identifying and grouping events, objects, and themes present in the captions, creating a unique name for each factoid, and marking the captions that contain each factoid. One caption could be given many different labels. For example, in cartoon #331, such factoids may be "bad directions", "police", "take me to your leader", "racism", or "headlights". After annotating, each set of captions contained about 60 factoids on average. On average a caption was annotated with 0.90 factoids, with approximately 80% of the discourse units having at least one factoid, 20% having at least two, and only 2% having more than two. Inter-annotator agreement was moderate, with an F1-score (described more in section 5) of 0.6 between annotators.

As van Halteren and Teufel (2003) also found

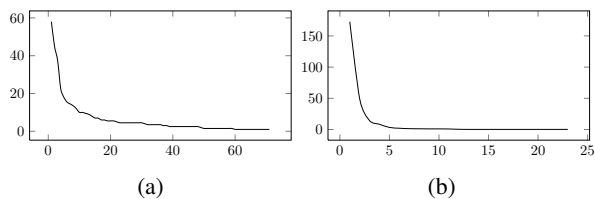


Figure 2: Average factoid frequency distributions for cartoon captions (a) and crossword clues (b).

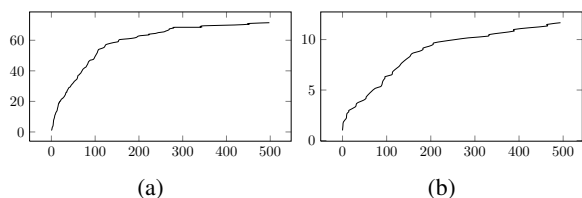


Figure 3: Growth of the number of unique factoids as the size of the corpus grows for cartoon captions (a) and crossword clues (b).

when examining factoid distributions in human-produced summaries, we found that the distribution of factoids in the caption set for each cartoon seems to follow a power law. Figure 2 shows the average frequencies of factoids, when ordered from most- to least-frequent. We also found a Heap’s law-type effect in the number of unique factoids compared to the size of the corpus, as in Figure 3.

### 3.2 Crossword Clues Data Set

Clues in crossword puzzles are typically obscure, requiring the reader to recognize double meanings or puns, which leads to a great deal of diversity. These clues can also refer to one or more of many different senses of the word. Table 2 shows examples of many different clues for the word “tea”. This table clearly illustrates the difference between factoids (the senses being referred to) and nuggets (the realization of the factoids).

The website `crosswordtracker.com` collects a large number of clues that appear in different published crossword puzzles and aggregates them according to their answer. From this site, we collected 200 sets of clues for common crossword answers.

We manually annotated 20 sets of crossword clues according to their factoids in the same fashion as described in section 3.1. On average each set of clues contains 283 clues and 15 different factoids. Inter-annotator agreement on this dataset was quite high with an F1-score of 0.96.

| Clue                     | Sense              |
|--------------------------|--------------------|
| Major Indian export      | drink              |
| Leaves for a break?      | drink              |
| Darjeeling, e.g.         | drink              |
| Afternoon social         | event              |
| 4:00 gathering           | event              |
| Sympathy partner         | film               |
| Mythical Irish queen     | person             |
| ----- Party movement     | political movement |
| Word with rose or garden | plant and place    |

Table 2: Examples of crossword clues and their different senses for the word “tea”.

## 4 Methods

### 4.1 Random Walk Method

We take a graph-based approach to the discovery of factoids, clustering a word similarity graph and taking the resulting clusters to be the factoids. Two different graphs, a word co-occurrence graph and a lexical similarity graph learned from the corpus, are compared. We also compare the graph-based methods against baselines of clustering and topic modeling.

#### 4.1.1 Word Co-occurrence Graph

To create the word co-occurrence graph, we create a link between every pair of words with an edge weight proportional to the number of times they both occur in the same discourse unit.

#### 4.1.2 Corpus-based Lexical Similarity Graph

To build the lexical similarity graph, a lexical similarity function is learned from the corpus, that is, from one set of captions or clues. We do this by computing feature vectors for each lemma and using the cosine similarity between these feature vectors as a lexical similarity function. We construct a word graph with edge weights proportional to the learned similarity of the respective word pairs.

We use three types of features in these feature vectors: context word features, context part-of-speech features, and spelling features. Context features are the presence of each word in a window of five words (two words on each side plus the word in question). Context part-of-speech features are the part-of-speech labels given by the Stanford POS tagger (Toutanova et al., 2003) within the same window. Spelling features are the counts of all character trigrams present in the word.

Table 3 shows examples of similar word pairs from the set of crossword clues for “tea”. From

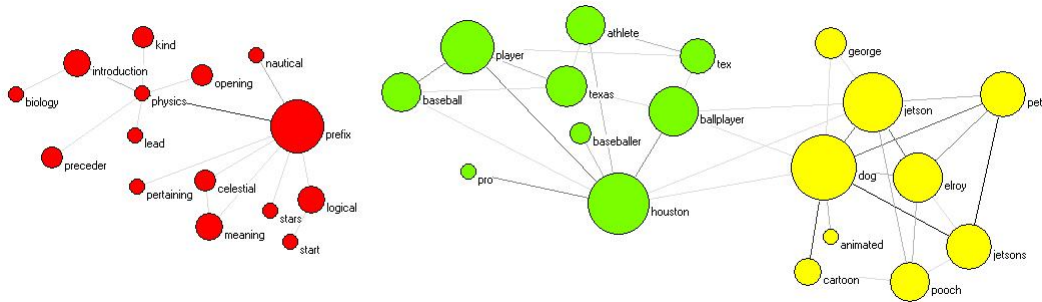


Figure 4: Example of natural clusters in a subsection of the word co-occurrence graph for the crossword clue “astro”.

| Word pair                     | Sim. |
|-------------------------------|------|
| (white-gloves, white-glove)   | 0.74 |
| (may, can)                    | 0.57 |
| (midafternoon, mid-afternoon) | 0.55 |
| (company, co.)                | 0.46 |
| (supermarket, market)         | 0.53 |
| (pick-me-up, perk-me-up)      | 0.44 |
| (green, black)                | 0.44 |
| (lady, earl)                  | 0.39 |
| (kenyan, indian)              | 0.38 |

Table 3: Examples of similar pairs of words as calculated on the set of crossword clues for “tea”.

this table, we can see that this method is able to successfully identify several similar word pairs that would be missed by most lexical databases: minor lexical variations, such as “pick-me-up” vs. “perk-me-up”; abbreviations, such as “company” and “co.”; and words that are similar only in this context, such as “lady” and “earl” (referring to Lady Grey and Earl Grey tea).

### 4.1.3 Graph Clustering

To cluster the word similarity graph, we use the Louvain graph clustering method (Blondel et al., 2008), a hierarchical method that optimizes graph modularity. This method produces several hierarchical cluster levels. We use the highest level, corresponding to the fewest number of clusters.

Figure 4 shows an example of clusters found in the word graph for the crossword clue “astro”. There are three obvious clusters, one for the Houston Astros baseball team, one for the dog in the Jetsons cartoon, and one for the lexical prefix “astro-”. In this example, two of the clusters are connected by a clue that mentions multiple senses, “Houston ballplayer or Jetson dog”.

### 4.1.4 Random Walk Factoid Assignment

After discovering factoids, the remaining task is to annotate captions according to the factoids they contain. We approach this problem by taking random walks on the word graph constructed in the previous sections, starting the random walks from words in the caption and measuring the hitting times to different clusters.

For each discourse unit, we repeatedly sample words from it and take Markov random walks starting from the nodes corresponding to the selected and lasting 10 steps (which is enough to ensure that every node in the graph can be reached). After 1000 random walks, we measure the average hitting time to each cluster, where a cluster is considered to be reached by the random walk the first time a node in that cluster is reached. Heuristically, 1000 random walks was more than enough to ensure that the factoid distribution had stabilized in development data.

The labels that are applied to a caption are the labels of the clusters that have a sufficiently low hitting time. We perform five-fold cross validation on each caption or set of clues and tune the threshold on the hitting time such that the average number of labels per unit produced matches the average number of labels per unit in the gold annotation of the held-out portion.

For example, a certain caption may have the following hitting times to the different factoid clusters:

|                 |      |
|-----------------|------|
| <i>factoid1</i> | 0.11 |
| <i>factoid2</i> | 0.75 |
| <i>factoid3</i> | 1.14 |
| <i>factoid4</i> | 2.41 |

If the held-out portion has 1.2 factoids per caption, it may be determined that the optimal thresh-

old on the hitting times is 0.8, that is, a threshold of 0.8 produces 1.2 factoids per caption in the test-set on average. In this case *factoid1* and *factoid2* would be marked for this caption, since the hitting times fall below the threshold.

## 4.2 Clustering

A simple baseline that can act as a surrogate for factoid annotation is clustering of discourse units, which is equivalent to assigning exactly one factoid (the name of its cluster) to each discourse unit. As our clustering method, we use C-Lexrank (Qazvinian and Radev, 2008), a method that has been well-tested on collective discourse.

## 4.3 Topic Model

Topic modeling is a natural way to approach the problem of factoid annotation, if we consider the topics to be factoids. We use the Mallet (McCallum, 2002) implementation of Latent Dirichlet Allocation (LDA) (Blei et al., 2003). As with the random walk method, we perform five-fold cross validation, tuning the threshold for the average number of labels per discourse unit to match the average number of labels in the held-out portion. Because LDA needs to know the number of topics *a priori*, we set the number of topics to be equal to the true number of factoids. We also use the average number of unique factoids in the held-out portion as the number of LDA topics.

## 5 Evaluation and Results

We evaluate this task in a way similar to pairwise clustering evaluation methods, where every pair of discourse units that should share at least one factoid and does is a true positive instance, every pair that should share a factoid and does not is a false negative, etc. From this we are able to calculate precision, recall, and F1-score. This is a reasonable evaluation method, since the average number of factoids per discourse unit is close to one. Because the factoids discovered by this method don't necessarily match the factoids chosen by the annotators, it doesn't make sense to try to measure whether two discourse units share the "correct" factoid.

Tables 4 and 5 show the results of the various methods on the cartoon captions and crossword clues datasets, respectively. On the crossword clues datasets, the random-walk-based methods are clearly superior to the other methods tested, whereas simple clustering is more effective on the

| Method                   | Prec. | Rec.  | F1    |
|--------------------------|-------|-------|-------|
| LDA                      | 0.318 | 0.070 | 0.115 |
| C-Lexrank                | 0.131 | 0.347 | 0.183 |
| Word co-occurrence graph | 0.115 | 0.348 | 0.166 |
| Word similarity graph    | 0.093 | 0.669 | 0.162 |

Table 4: Performance of various methods annotating factoids for cartoon captions.

| Method                   | Prec. | Rec.  | F1    |
|--------------------------|-------|-------|-------|
| LDA                      | 0.315 | 0.067 | 0.106 |
| C-Lexrank                | 0.702 | 0.251 | 0.336 |
| Word co-occurrence graph | 0.649 | 0.257 | 0.347 |
| Word similarity graph    | 0.575 | 0.397 | 0.447 |

Table 5: Performance of various methods annotating factoids for crossword clues.

cartoon captions dataset.

In some sense, the two datasets in this paper both represent difficult domains, ones in which authors are intentionally obscure. The good results achieved on the crossword clues dataset indicate that this obscurity can be overcome when discourse units are short. Future work in this vein includes applying these methods to domains, such as newswire, that are more typical for summarization, and if necessary, investigating how these methods can best be applied to domains with longer sentences.

## References

- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.
- Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.
- 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.
- Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics.
- Leonhard Hennig, Ernesto William De Luca, and Sahin Albayrak. 2010. Learning summary content units with topic modeling. In *Proceedings of the 23rd*

*International Conference on Computational Linguistics: Posters*, COLING '10, pages 391–399, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method.

Vahed Qazvinian and Dragomir R Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 689–696. Association for Computational Linguistics.

Vahed Qazvinian and Dragomir R Radev. 2011. Learning from collective human behavior to introduce diversity in lexical choice. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1098–1108.

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. Association for Computational Linguistics.

Hans Van Halteren and Simone Teufel. 2003. Examining the consensus between human summaries: initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 57–64. Association for Computational Linguistics.

# Identifying English and Hungarian Light Verb Constructions: A Contrastive Approach

Veronika Vincze<sup>1,2</sup>, István Nagy T.<sup>2</sup> and Richárd Farkas<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  
{nistvan, rfarkas}@inf.u-szeged.hu

## Abstract

Here, we introduce a machine learning-based approach that allows us to identify light verb constructions (LVCs) in Hungarian and English free texts. We also present the results of our experiments on the SzegedParalellFX English–Hungarian parallel corpus where LVCs were manually annotated in both languages. With our approach, we were able to contrast the performance of our method and define language-specific features for these typologically different languages. Our presented method proved to be sufficiently robust as it achieved approximately the same scores on the two typologically different languages.

## 1 Introduction

In natural language processing (NLP), a significant part of research is carried out on the English language. However, the investigation of languages that are typologically different from English is also essential since it can lead to innovations that might be usefully integrated into systems developed for English. Comparative approaches may also highlight some important differences among languages and the usefulness of techniques that are applied.

In this paper, we focus on the task of identifying light verb constructions (LVCs) in English and Hungarian free texts. Thus, the same task will be carried out for English and a morphologically rich language. We compare whether the same set of features can be used for both languages, we investigate the benefits of integrating language specific features into the systems and we explore how the systems could be further improved. For this purpose, we make use of the English–Hungarian parallel corpus SzegedParalellFX (Vincze, 2012), where LVCs have been manually annotated.

## 2 Light Verb Constructions

Light verb constructions (e.g. *to give advice*) are a subtype of multiword expressions (Sag et al., 2002). They consist of a nominal and a verbal component where the verb functions as the syntactic head, but the semantic head is the noun. The verbal component (also called a light verb) usually loses its original sense to some extent. Although it is the noun that conveys most of the meaning of the construction, the verb itself cannot be viewed as semantically bleached (Apresjan, 2004; Alonso Ramos, 2004; Sanromán Vilas, 2009) since it also adds important aspects to the meaning of the construction (for instance, the beginning of an action, such as *set on fire*, see Mel'čuk (2004)). The meaning of LVCs can be only partially computed on the basis of the meanings of their parts and the way they are related to each other, hence it is important to treat them in a special way in many NLP applications.

LVCs are usually distinguished from productive or literal verb + noun constructions on the one hand and idiomatic verb + noun expressions on the other (Fazly and Stevenson, 2007). Variativity and omitting the verb play the most significant role in distinguishing LVCs from productive constructions and idioms (Vincze, 2011). Variativity reflects the fact that LVCs can be often substituted by a verb derived from the same root as the nominal component within the construction: productive constructions and idioms can be rarely substituted by a single verb (like *make a decision – decide*). Omitting the verb exploits the fact that it is the nominal component that mostly bears the semantic content of the LVC, hence the event denoted by the construction can be determined even without the verb in most cases. Furthermore, the very same noun + verb combination 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), hence it is important to identify them in context.

In theoretical linguistics, Kearns (2002) distinguishes between two subtypes of light verb constructions. True light verb constructions such as *to give a wipe* or *to have a laugh* and vague action verbs such as *to make an agreement* or *to do the ironing* differ in some syntactic and semantic features and can be separated by various tests, e.g. passivization, WH-movement, pronominalization etc. This distinction also manifests in natural language processing as several authors pay attention to the identification of just true light verb constructions, e.g. Tu and Roth (2011). However, here we do not make such a distinction and aim to identify all types of light verb constructions both in English and in Hungarian, in accordance with the annotation principles of SZPFX.

The canonical form of a Hungarian light verb construction is a bare noun + third person singular verb. However, they may occur in non-canonical versions as well: the verb may precede the noun, or the noun and the verb may be not adjacent due to the free word order. Moreover, as Hungarian is a morphologically rich language, the verb may occur in different surface forms inflected for tense, mood, person and number. These features will be paid attention to when implementing our system for detecting Hungarian LVCs.

### 3 Related Work

Recently, LVCs have received special interest in the NLP research community. They have been automatically identified in several languages such as English (Cook et al., 2007; Bannard, 2007; Vincze et al., 2011a; Tu and Roth, 2011), Dutch (Van de Cruys and Moirón, 2007), Basque (Gurrutxaga and Alegria, 2011) and German (Evert and Kermeš, 2003).

Parallel corpora are of high importance in the automatic identification of multiword expressions: it is usually one-to-many correspondence that is exploited when designing methods for detecting multiword expressions. Caseli et al. (2010) developed an alignment-based method for extracting multiword expressions from Portuguese–English parallel corpora. Samardžić and Merlo (2010) analyzed English and German light verb constructions in parallel corpora: they pay special attention to their manual and automatic alignment. Zariëb

and Kuhn (2009) argued that multiword expressions can be reliably detected in parallel corpora by using dependency-parsed, word-aligned sentences. Sinha (2009) detected Hindi complex predicates (i.e. a combination of a light verb and a noun, a verb or an adjective) in a Hindi–English parallel corpus by identifying a mismatch of the Hindi light verb meaning in the aligned English sentence. Many-to-one correspondences were also exploited by Attia et al. (2010) when identifying Arabic multiword expressions relying on asymmetries between parallel entry titles of Wikipedia. Tsvetkov and Wintner (2010) identified Hebrew multiword expressions by searching for misalignments in an English–Hebrew parallel corpus.

To the best of our knowledge, parallel corpora have not been used for testing the efficiency of an MWE-detecting method for two languages at the same time. Here, we investigate the performance of our base LVC-detector on English and Hungarian and pay special attention to the added value of language-specific features.

## 4 Experiments

In our investigations we made use of the Szeged-ParalellFX English-Hungarian parallel corpus, which consists of 14,000 sentences and contains about 1370 LVCs for each language. In addition, we are aware of two other corpora – the Szeged Treebank (Vincze and Csirik, 2010) and Wiki50 (Vincze et al., 2011b) –, which were manually annotated for LVCs on the basis of similar principles as SZPFX, so we exploited these corpora when defining our features.

To automatically identify LVCs in running texts, a machine learning based approach was applied. This method first parsed each sentence and extracted potential LVCs. Afterwards, a binary classification method was utilized, which can automatically classify potential LVCs as an LVC or not. This binary classifier was based on a rich feature set described below.

The candidate extraction method investigated the dependency relation among the verbs and nouns. Verb-object, verb-subject, verb-prepositional object, verb-other argument (in the case of Hungarian) and noun-modifier pairs were collected from the texts. The dependency labels were provided by the Bohnet parser (Bohnet, 2010) for English and by *magyarlan* 2.0 (Zsibrita et al., 2013) for Hungarian.



The features used by the binary classifier can be categorised as follows:

**Morphological features:** As the nominal component of LVCs is typically derived from a verbal stem (*make a decision*) or coincides with a verb (*have a walk*), the **VerbalStem** binary feature focuses on the stem of the noun; if it had a verbal nature, the candidates were marked as *true*. The **POS-pattern** feature investigates the POS-tag sequence of the potential LVC. If it matched one pattern typical of LVCs (e.g. verb + noun) the candidate was marked as *true*; otherwise as *false*. The English **auxiliary** verbs, *do* and *have* often occur as light verbs, hence we defined a feature for the two verbs to denote whether or not they were auxiliary verbs in a given sentence. The POS code of the next word of LVC candidate was also applied as a feature. As Hungarian is a morphologically rich language, we were able to define various morphology-based features like the case of the noun or its number etc. Nouns which were historically derived from verbs but were not treated as derivation by the Hungarian morphological parser were also added as a feature.

**Semantic features:** This feature also exploited the fact that the nominal component is usually derived from verbs. Consequently, the *activity* or *event* semantic senses were looked for among the upper level hyperonyms of the head of the noun phrase in English WordNet 3.1<sup>1</sup> and in the Hungarian WordNet (Miháltz et al., 2008).

**Orthographic features:** The **suffix** feature is also based on the fact that many nominal components in LVCs are derived from verbs. This feature checks whether the lemma of the noun ended in a given character bi- or trigram. The **number of words** of the candidate LVC was also noted and applied as a feature.

**Statistical features:** Potential English LVCs and their **occurrences** were collected from 10,000 English Wikipedia pages by the candidate extraction method. The number of occurrences was used as a feature when the candidate was one of the syntactic phrases collected.

**Lexical features:** We exploit the fact that the **most common verbs** are typically light verbs. Therefore, fifteen typical light verbs were selected from the list of the most frequent verbs taken from the Wiki50 (Vincze et al., 2011b) in the case of English and from the Szeged Treebank (Vincze and

Csirik, 2010) in the case of Hungarian. Then, we investigated whether the lemmatised verbal component of the candidate was one of these fifteen verbs. The **lemma of the noun** was also applied as a lexical feature. The nouns found in LVCs were collected from the above-mentioned corpora. Afterwards, we constructed **lists of lemmatised LVCs** got from the other corpora.

**Syntactic features:** As the candidate extraction methods basically depended on the **dependency relation** between the noun and the verb, they could also be utilised in identifying LVCs. Though the *dobj*, *prep*, *rcmod*, *partmod* or *nsubjpass* dependency labels were used in candidate extraction in the case of English, these syntactic relations were defined as features, while the *att*, *obj*, *obl*, *subj* dependency relations were used in the case of Hungarian. When the noun had a **determiner** in the candidate LVC, it was also encoded as another syntactic feature.

Our feature set includes language-independent and language-specific features as well. Language-independent features seek to acquire general features of LVCs while language-specific features can be applied due to the different grammatical characteristics of the two languages or due to the availability of different resources. Table 1 shows which features were applied for which language.

We experimented with several learning algorithms and decision trees have been proven performing best. This is probably due to the fact that our feature set consists of compact – i.e. high-level – features. We trained the J48 classifier of the WEKA package (Hall et al., 2009). This machine learning approach implements the decision trees algorithm C4.5 (Quinlan, 1993). The J48 classifier was trained with the above-mentioned features and we evaluated it in a 10-fold cross validation.

The potential LVCs which are extracted by the candidate extraction method but not marked as positive in the gold standard were classed as negative. As just the positive LVCs were annotated on the SZPFX corpus, the  $F_{\beta=1}$  score interpreted on the positive class was employed as an evaluation metric. The candidate extraction methods could not detect all LVCs in the corpus data, so some positive elements in the corpora were not covered. Hence, we regarded the omitted LVCs as false negatives in our evaluation.

<sup>1</sup><http://wordnet.princeton.edu>

| Features              | Base | English | Hungarian |
|-----------------------|------|---------|-----------|
| Orthographical        | •    | –       | –         |
| VerbalStem            | •    | –       | –         |
| POS pattern           | •    | –       | –         |
| LVC list              | •    | –       | –         |
| Light verb list       | •    | –       | –         |
| Semantic features     | •    | –       | –         |
| Syntactic features    | •    | –       | –         |
| Auxiliary verb        | –    | •       | –         |
| Determiner            | –    | •       | –         |
| Noun list             | –    | •       | –         |
| POS After             | –    | •       | –         |
| LVC freq. stat.       | –    | •       | –         |
| Agglutinative morph.  | –    | –       | •         |
| Historical derivation | –    | –       | •         |

Table 1: The basic feature set and language-specific features.

|    | English           | Hungarian         |
|----|-------------------|-------------------|
| ML | 63.29/56.91/59.93 | 66.1/50.04/56.96  |
| DM | 73.71/29.22/41.67 | 63.24/34.46/44.59 |

Table 2: Results obtained in terms of precision, recall and F-score. ML: machine learning approach DM: dictionary matching method.

## 5 Results

As a baseline, a context free dictionary matching method was applied. For this, the gold-standard LVC lemmas were gathered from Wiki50 and the Szeged Treebank. Texts were lemmatized and if an item on the list was found in the text, it was treated as an LVC.

Table 2 lists the results got on the two different parts of SZPFX using the machine learning-based approach and the baseline dictionary matching. The dictionary matching approach yielded the highest precision on the English part of SZPFX, namely 73.71%. However, the machine learning-based approach proved to be the most successful as it achieved an F-score that was 18.26 higher than that with dictionary matching. Hence, this method turned out to be more effective regarding recall. At the same time, the machine learning and dictionary matching methods got roughly the same precision score on the Hungarian part of SZPFX, but again the machine learning-based approach achieved the best F-score. While in the case of English the dictionary matching method got a higher precision score, the machine learning approach proved to be more effective.

An ablation analysis was carried out to examine the effectiveness of each individual feature of the machine learning-based candidate classifica-

| Feature           | English | Hungarian |
|-------------------|---------|-----------|
| All               | 59.93   | 56.96     |
| Lexical           | -19.11  | -14.05    |
| Morphological     | -1.68   | -1.75     |
| Orthographic      | -0.43   | -3.31     |
| Syntactic         | -1.84   | -1.28     |
| Semantic          | -2.17   | -0.34     |
| Statistical       | -2.23   | –         |
| Language-specific | -1.83   | -1.05     |

Table 3: The usefulness of individual features in terms of F-score using the SZPFX corpus.

tion. For each feature type, a J48 classifier was trained with all of the features except that one. We also investigated how language-specific features improved the performance compared to the base feature set. We then compared the performance to that got with all the features. Table 3 shows the contribution of each individual feature type on the SZPFX corpus. For each of the two languages, each type of feature contributed to the overall performance. Lexical features were very effective in both languages.

## 6 Discussion

According to the results, our base system is robust enough to achieve approximately the same results on two typologically different languages. Language-specific features further contribute to the performance as shown by the ablation analysis. It should be also mentioned that some of the base features (e.g. POS-patterns, which we thought would be useful for English due to the fixed word order) were originally inspired by one of the languages and later expanded to the other one (i.e. they were included in the base feature set) since it was also effective in the case of the other language. Thus, a multilingual approach may be also beneficial in the case of monolingual applications as well.

The most obvious difference between the performances on the two languages is the recall scores (the difference being 6.87 percentage points between the two languages). This may be related to the fact that the distribution of light verbs is quite different in the two languages. While the top 15 verbs covers more than 80% of the English LVCs, in Hungarian, this number is only 63% (and in order to reach the same coverage, 38 verbs should be included). Another difference is that there are 102

different verbs in English, which follow the Zipf distribution, on the other hand, there are 157 Hungarian verbs with a more balanced distributional pattern. Thus, fewer verbs cover a greater part of LVCs in English than in Hungarian and this also explains why lexical features contribute more to the overall performance in English. This fact also indicates that if verb lists are further extended, still better recall scores may be achieved for both languages.

As for the effectiveness of morphological and syntactic features, morphological features perform better on a language with a rich morphological representation (Hungarian). However, syntax plays a more important role in LVC detection in English: the added value of syntax is higher for the English corpora than for the Hungarian one, where syntactic features are also encoded in suffixes, i.e. morphological information.

We carried out an error analysis in order to see how our system could be further improved and the errors reduced. We concluded that there were some general and language-specific errors as well. Among the general errors, we found that LVCs with a rare light verb were difficult to recognize (e.g. *to utter a lie*). In other cases, an originally deverbal noun was used in a lexicalised sense together with a typical light verb ((e.g. *buildings are given (something)*) and these candidates were falsely classed as LVCs. Also, some errors in POS-tagging or dependency parsing also led to some erroneous predictions.

As for language-specific errors, English verb-particle combinations (VPCs) followed by a noun were often labeled as LVCs such as *make up his mind* or *give in his notice*. In Hungarian, verb + proper noun constructions (*Hamletet játsszák* (Hamlet-ACC play-3PL.DEF) “they are playing Hamlet”) were sometimes regarded as LVCs since the morphological analysis does not make a distinction between proper and common nouns. These language-specific errors may be eliminated by integrating a VPC detector and a named entity recognition system into the English and Hungarian systems, respectively.

Although there has been a considerable amount of literature on English LVC identification (see Section 3), our results are not directly comparable to them. This may be explained by the fact that different authors aimed to identify a different scope of linguistic phenomena and thus interpreted the

concept of “light verb construction” slightly differently. For instance, Tu and Roth (2011) and Tan et al. (2006) focused only on true light verb constructions while only object–verb pairs are considered in other studies (Stevenson et al., 2004; Tan et al., 2006; Fazly and Stevenson, 2007; Cook et al., 2007; Bannard, 2007; Tu and Roth, 2011). Several other studies report results only on light verb constructions formed with certain light verbs (Stevenson et al., 2004; Tan et al., 2006; Tu and Roth, 2011). In contrast, we aimed to identify all kinds of LVCs, i.e. we did not apply any restrictions on the nature of LVCs to be detected. In other words, our task was somewhat more difficult than those found in earlier literature. Although our results are somewhat lower on English LVC detection than those attained by previous studies, we think that despite the difficulty of the task, our method could offer promising results for identifying all types of LVCs both in English and in Hungarian.

## 7 Conclusions

In this paper, we introduced our machine learning-based approach for identifying LVCs in Hungarian and English free texts. The method proved to be sufficiently robust as it achieved approximately the same scores on two typologically different languages. The language-specific features further contributed to the performance in both languages. In addition, some language-independent features were inspired by one of the languages, so a multilingual approach proved to be fruitful in the case of monolingual LVC detection as well.

In the future, we would like to improve our system by conducting a detailed analysis of the effect of each feature on the results. Later, we also plan to adapt the tool to other types of multiword expressions and conduct further experiments on languages other than English and Hungarian, the results of which may further lead to a more robust, general LVC system. Moreover, we can improve the method applied in each language by implementing other language-specific features 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

- Margarita Alonso Ramos. 2004. *Las construcciones con verbo de apoyo*. Visor Libros, Madrid.
- Jurij D. Apresjan. 2004. O semantičeskoj nepustote i motivirovannosti glagol'nyx leksičeskix funkcij. *Voprosy jazykoznanija*, (4):3–18.
- Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina, and Josef van Genabith. 2010. Automatic Extraction of Arabic Multiword Expressions. In *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*, pages 19–27, Beijing, China, August. Coling 2010 Organizing Committee.
- Colin Bannard. 2007. A measure of syntactic flexibility for automatically identifying multiword expressions in corpora. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*, MWE '07, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.
- Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97.
- Helena de Medeiros Caseli, Carlos Ramisch, Maria das Graças Volpe Nunes, and Aline Villavicencio. 2010. Alignment-based extraction of multiword expressions. *Language Resources and Evaluation*, 44(1-2):59–77.
- 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 the Workshop on a Broader Perspective on Multiword Expressions*, MWE '07, pages 41–48, Morristown, NJ, USA. Association for Computational Linguistics.
- Stefan Evert and Hannah Kermes. 2003. Experiments on candidate data for collocation extraction. In *Proceedings of EACL 2003*, pages 83–86.
- Afsaneh Fazly and Suzanne Stevenson. 2007. Distinguishing Subtypes of Multiword Expressions Using Linguistically-Motivated Statistical Measures. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.
- Antton Gurrutxaga and Iñaki Alegria. 2011. Automatic Extraction of NV Expressions in Basque: Basic Issues on Cooccurrence Techniques. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 2–7, Portland, Oregon, USA, June. Association for Computational Linguistics.
- 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.
- Igor Mel'čuk. 2004. Verbes supports sans peine. *Linguisticae Investigationes*, 27(2):203–217.
- Márton Miháلتz, Csaba Hatvani, Judit Kuti, György Szarvas, János Csirik, Gábor Prószéky, and Tamás Váradi. 2008. Methods and Results of the Hungarian WordNet Project. In Attila Tanács, Dóra Csendes, Veronika Vincze, Christiane Fellbaum, and Piek Vossen, editors, *Proceedings of the Fourth Global WordNet Conference (GWC 2008)*, pages 311–320, Szeged. University of Szeged.
- 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 the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (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. Association for Computational Linguistics.
- 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 David Beck, Kim Gerdes, Jasmina Miličević, and Alain Polguère, editors, *Proceedings of the Fourth International Conference on Meaning-Text Theory – MTT'09*, pages 327–337, Montreal, Canada. Université de Montréal.
- R. Mahesh K. Sinha. 2009. Mining Complex Predicates In Hindi Using A Parallel Hindi-English Corpus. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 40–46, Singapore, August. Association for Computational Linguistics.
- Suzanne Stevenson, Afsaneh Fazly, and Ryan North. 2004. Statistical Measures of the Semi-Productivity of Light Verb Constructions. In Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors, *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pages 1–8, Barcelona, Spain, July. Association for Computational Linguistics.
- 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 the EACL Workshop on*

- Multi-Word Expressions in a Multilingual Contexts*, pages 49–56, Trento, Italy, April. Association for Computational Linguistics.
- Yulia Tsvetkov and Shuly Wintner. 2010. Extraction of multi-word expressions from small parallel corpora. In *Coling 2010: Posters*, pages 1256–1264, Beijing, China, August. Coling 2010 Organizing Committee.
- Yuancheng Tu and Dan Roth. 2011. Learning English Light Verb Constructions: Contextual or Statistical. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 31–39, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Tim Van de Cruys and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions, MWE '07*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.
- Veronika Vincze and János Csirik. 2010. Hungarian corpus of light verb constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1110–1118, Beijing, China, August. Coling 2010 Organizing Committee.
- Veronika Vincze, István Nagy T., and Gábor Berend. 2011a. Detecting Noun Compounds and Light Verb Constructions: a Contrastive Study. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 116–121, Portland, Oregon, USA, June. ACL.
- Veronika Vincze, István Nagy T., and Gábor Berend. 2011b. Multiword expressions and named entities in the Wiki50 corpus. In *Proceedings of RANLP 2011*, Hissar, Bulgaria.
- Veronika Vincze. 2011. *Semi-Compositional Noun + Verb Constructions: Theoretical Questions and Computational Linguistic Analyses*. Ph.D. thesis, University of Szeged, Szeged, Hungary.
- Veronika Vincze. 2012. Light Verb Constructions in the SzegedParalellFX English–Hungarian Parallel Corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May. European Language Resources Association (ELRA).
- Sina Zarrieß and Jonas Kuhn. 2009. Exploiting Translational Correspondences for Pattern-Independent MWE Identification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 23–30, Singapore, August. Association for Computational Linguistics.
- János Zsibrita, Veronika Vincze, and Richárd Farkas. 2013. magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés [magyarlanc 2.0: Syntactic parsing and accelerated POS-tagging]. In Attila Tanács and Veronika Vincze, editors, *MSzNy 2013 – IX. Magyar Számítógépes Nyelvészeti Konferencia*, pages 368–374, Szeged. Szegedi Tudományegyetem.

# English→Russian MT evaluation campaign

**Pavel Braslavski**  
Kontur Labs /  
Ural Federal  
University, Russia  
pbras@yandex.ru

**Alexander Beloborodov**  
Ural Federal University  
Russia  
xander-beloborodov  
@yandex.ru

**Maxim Khalilov**  
TAUS Labs  
The Netherlands  
maxim  
@tauslabs.com

**Serge Sharoff**  
University of Leeds  
UK  
s.sharoff  
@leeds.ac.uk

## Abstract

This paper presents the settings and the results of the ROMIP 2013 MT shared task for the English→Russian language direction. The quality of generated translations was assessed using automatic metrics and human evaluation. We also discuss ways to reduce human evaluation efforts using pairwise sentence comparisons by human judges to simulate sort operations.

## 1 Introduction

Machine Translation (MT) between English and Russian was one of the first translation directions tested at the dawn of MT research in the 1950s (Hutchins, 2000). Since then the MT paradigms changed many times, many systems for this language pair appeared (and disappeared), but as far as we know there was no systematic quantitative evaluation of a range of systems, analogous to DARPA'94 (White et al., 1994) and later evaluation campaigns. The Workshop on Statistical MT (WMT) in 2013 has announced a Russian evaluation track for the first time.<sup>1</sup> However, this evaluation is currently ongoing, it should include new methods for building statistical MT (SMT) systems for Russian from the data provided in this track, but it will not cover the performance of existing systems, especially rule-based (RBMT) or hybrid ones.

Evaluation campaigns play an important role in promotion of the progress for MT technologies. Recently, there have been a number of MT shared tasks for combinations of several European, Asian and Semitic languages (Callison-Burch et al., 2011; Callison-Burch et al., 2012; Federico et al., 2012), which we took into account in designing the campaign for the English-Russian direction. The evaluation has been held in the

<sup>1</sup><http://www.statmt.org/wmt13/>

context of ROMIP,<sup>2</sup> which stands for Russian Information Retrieval Evaluation Seminar and is a TREC-like<sup>3</sup> Russian initiative started in 2002.

One of the main challenges in developing MT systems for Russian and for evaluating them is the need to deal with its free word order and complex morphology. Long-distance dependencies are common, and this creates problems for both RBMT and SMT systems (especially for phrase-based ones). Complex morphology also leads to considerable sparseness for word alignment in SMT.

The language direction was chosen to be English→Russian, first because of the availability of native speakers for evaluation, second because the systems taking part in this evaluation are mostly used in translation of English texts for the Russian readers.

## 2 Corpus preparation

In designing the set of texts for evaluation, we had two issues in mind. First, it is known that the domain and genre can influence MT performance (Langlais, 2002; Babych et al., 2007), so we wanted to control the set of genres. Second, we were aiming at using sources allowing distribution of texts under a Creative Commons licence. In the end two genres were used coming from two sources. The newswire texts were collected from the English Wikinews website.<sup>4</sup> The second genre was represented by 'regulations' (laws, contracts, rules, etc), which were collected from the Web using a genre classification method described in (Sharoff, 2010). The method provided a sufficient accuracy (74%) for the initial selection of texts under the category of 'regulations,' which was followed by a manual check to reject texts clearly outside of this genre category.

<sup>2</sup><http://romip.ru/en/>

<sup>3</sup><http://trec.nist.gov/>

<sup>4</sup><http://en.wikinews.org/>

The initial corpus consists of 8,356 original English texts that make up 148,864 sentences. We chose to retain the entire texts in the corpus rather than individual sentences, since some MT systems may use information beyond isolated sentences. 100,889 sentences originated from Wikinews; 47,975 sentences came from the ‘regulations’ corpus. The first 1,002 sentences were published in advance to allow potential participants time to adjust their systems to the corpus format. The remaining 147,862 sentences were the corpus for testing translation into Russian. Two examples of texts in the corpus:

90237 *Ambassadors from the United States of America, Australia and Britain have all met with Fijian military officers to seek assurances that there wasn't going to be a coup.*

102835 *If you are given a discount for booking more than one person onto the same date and you later wish to transfer some of the delegates to another event, the fees will be recalculated and you will be asked to pay additional fees due as well as any administrative charge.*

For automatic evaluation we randomly selected 947 ‘clean’ sentences, i.e. those with clear sentence boundaries, no HTML markup remains, etc. (such flaws sometimes occur in corpora collected from the Web). 759 sentences originated from the ‘news’ part of the corpus, the remaining 188 came from the ‘regulations’ part. The sentences came from sources without published translations into Russian, so that some of the participating systems do not get unfair advantage by using them for training. These sentences were translated by professional translators. For manual evaluation, we randomly selected 330 sentences out of 947 used for automatic evaluation, specifically, 190 from the ‘news’ part and 140 from the ‘regulations’ part.

The organisers also provided participants with access to the following additional resources:

- 1 million sentences from the English-Russian parallel corpus released by Yandex (the same as used in WMT13)<sup>5</sup>;
- 119 thousand sentences from the English-Russian parallel corpus from the TAUS Data Repository.<sup>6</sup>

These resources are not related to the test corpus of the evaluation campaign. Their purpose was

<sup>5</sup><https://translate.yandex.ru/corpus?lang=en>

<sup>6</sup><https://www.tausdata.org>

to make it easier to participate in the shared task for teams without sufficient data for this language pair.

### 3 Evaluation methodology

The main idea of manual evaluation was (1) to make the assessment as simple as possible for a human judge and (2) to make the results of evaluation unambiguous. We opted for pairwise comparison of MT outputs. This is different from simultaneous *ranking* of several MT outputs, as commonly used in WMT evaluation campaigns. In case of a large number of participating systems each assessor ranks only a subset of MT outputs. However, a fair overall ranking cannot be always derived from such partial rankings (Callison-Burch et al., 2012). The pairwise comparisons we used can be directly converted into unambiguous overall rankings. This task is also much simpler for human judges to complete. On the other hand, pairwise comparisons require a larger number of evaluation decisions, which is feasible only for few participants (and we indeed had relatively few submissions in this campaign). Below we also discuss how to reduce the amount of human efforts for evaluation.

In our case the assessors were asked to make a pairwise comparison of two sentences translated by two different MT systems against a gold standard translation. The question for them was to judge translation adequacy, i.e., which MT output conveys information from the reference translation better. The source English sentence was not presented to the assessors, because we think that we can have more trust in understanding of the source text by a professional translator. The translator also had access to the entire text, while the assessors could only see a single sentence.

For human evaluation we employed the multi-functional TAUS DQF tool<sup>7</sup> in the ‘Quick Comparison’ mode.

Assessors’ judgements resulted in rankings for each sentence in the test set. In case of ties the ranks were averaged, e.g. when the ranks of the systems in positions 2-4 and 7-8 were tied, their ranks became: 1 3 3 3 5 6 7.5 7.5. To produce the final ranking, the sentence-level ranks were averaged over all sentences.

Pairwise comparisons are time-consuming:  $n$

<sup>7</sup><https://tauslabs.com/dynamic-quality/dqf-tools-mt>

| Metric                                 | OS1          | OS2   | OS3   | OS4   | P1           | P2    | P3    | P4    | P5    | P6    | P7    |
|--|--------------|-------|-------|-------|--------------|-------|-------|-------|-------|-------|-------|
| Automatic metrics ALL (947 sentences)  |              |       |       |       |              |       |       |       |       |       |       |
| BLEU                                   | 0.150        | 0.141 | 0.133 | 0.124 | <b>0.157</b> | 0.112 | 0.105 | 0.073 | 0.094 | 0.071 | 0.073 |
| NIST                                   | <b>5.12</b>  | 4.94  | 4.80  | 4.67  | 5.00         | 4.46  | 4.11  | 2.38  | 4.16  | 3.362 | 3.38  |
| Meteor                                 | <b>0.258</b> | 0.240 | 0.231 | 0.240 | 0.251        | 0.207 | 0.169 | 0.133 | 0.178 | 0.136 | 0.149 |
| TER                                    | <b>0.755</b> | 0.766 | 0.764 | 0.758 | 0.758        | 0.796 | 0.901 | 0.931 | 0.826 | 0.934 | 0.830 |
| GTM                                    | <b>0.351</b> | 0.338 | 0.332 | 0.336 | 0.349        | 0.303 | 0.246 | 0.207 | 0.275 | 0.208 | 0.230 |
| Automatic metrics NEWS (759 sentences) |              |       |       |       |              |       |       |       |       |       |       |
| BLEU                                   | 0.137        | 0.131 | 0.123 | 0.114 | <b>0.153</b> | 0.103 | 0.096 | 0.070 | 0.083 | 0.066 | 0.067 |
| NIST                                   | <b>4.86</b>  | 4.72  | 4.55  | 4.35  | 4.79         | 4.26  | 3.83  | 2.47  | 3.90  | 3.20  | 3.19  |
| Meteor                                 | 0.241        | 0.224 | 0.214 | 0.222 | <b>0.242</b> | 0.192 | 0.156 | 0.127 | 0.161 | 0.126 | 0.136 |
| TER                                    | 0.772        | 0.776 | 0.784 | 0.777 | <b>0.768</b> | 0.809 | 0.908 | 0.936 | 0.844 | 0.938 | 0.839 |
| GTM                                    | 0.335        | 0.324 | 0.317 | 0.320 | <b>0.339</b> | 0.290 | 0.233 | 0.201 | 0.257 | 0.199 | 0.217 |

Table 1: Automatic evaluation results

cases require  $\frac{n(n-1)}{2}$  pairwise decisions. In this study we also simulated a ‘human-assisted’ insertion sort algorithm and its variant with binary search. The idea is to run a standard sort algorithm and ask a human judge each time a comparison operation is required. This assumes that human perception of quality is transitive: if we know that  $A < B$  and  $B < C$ , we can spare evaluation of  $A$  and  $C$ . This approach also implies that sentence pairs to judge are generated and presented to assessors on the fly; each decision contributes to selection of the pairs to be judged in the next step. If the systems are pre-sorted in a reasonable way (e.g. by an MT metric, under assumption that automatic pre-ranking is closer to the ‘ideal’ ranking than a random one), then we can potentially save even more pairwise comparison operations. Pre-sorting makes ranking somewhat biased in favour of the order established by an MT metric. For example, if it favours one system against another, while in human judgement they are equal, the final ranking will preserve the initial order. Insertion sort of  $n$  sentences requires  $n - 1$  comparisons in the best case of already sorted data and  $\frac{n(n-1)}{2}$  in the worst case (reversely ordered data). Insertion sort with binary search requires  $\sim n \log n$  comparisons regardless of the initial order. For this study we ran exhaustive pairwise evaluation and used its results to simulate human-assisted sorting.

In addition to human evaluation, we also ran system-level automatic evaluations using BLEU (Papineni et al., 2001), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2009), and GTM (Turian et al., 2003). We also wanted to estimate the correla-

tions of these metrics with human judgements for the English→Russian pair on the corpus level and on the level of individual sentences.

## 4 Results

We received results from five teams, two teams submitted two runs each, which totals seven participants’ runs (referred to as P1..P7 in the paper). The participants represent SMT, RBMT, and hybrid approaches. They included established groups from academia and industry, as well as new research teams. The evaluation runs also included the translations of the 947 test sentences produced by four free online systems in their default modes (referred to as OS1..OS4). For 11 runs automatic evaluation measures were calculated; eight runs underwent manual evaluation (four online systems plus four participants’ runs; no manual evaluation was done by agreement with the participants for the runs P3, P6, and P7 to reduce the workload).

| ID   | Name and information              |
|------|-----------------------------------|
| OS1  | Phrase-based SMT                  |
| OS2  | Phrase-based SMT                  |
| OS3  | Hybrid (RBMT+statistical PE)      |
| OS4  | Dependency-based SMT              |
| P1   | Compreno, Hybrid, ABBYY Corp      |
| P2   | Pharaon, Moses, Yandex&TAUS data  |
| P3,4 | Balagur, Moses, Yandex&news data  |
| P5   | ETAP-3, RBMT, (Boguslavsky, 1995) |
| P6,7 | Pereved, Moses, Internet data     |

OS3 is a hybrid system based on RBMT with SMT post-editing (PE). P1 is a hybrid system with analysis and generation driven by statistical evaluation of hypotheses.



| All (330 sentences)                               |       |       |       |       |       |       |             |
|---|-------|-------|-------|-------|-------|-------|-------------|
| OS3 (highest)                                     | P1    | OS1   | OS2   | OS4   | P5    | P2    | P4 (lowest) |
| 3.159   | 3.350 | 3.530 | 3.961 | 4.082 | 5.447 | 5.998 | 6.473       |
| News (190 sentences)                              |       |       |       |       |       |       |             |
| OS3 (highest)                                     | P1    | OS1   | OS2   | OS4   | P5    | P2    | P4 (lowest) |
| 2.947   | 3.450 | 3.482 | 4.084 | 4.242 | 5.474 | 5,968 | 6,353       |
| Regulations (140 sentences)                       |       |       |       |       |       |       |             |
| P1 (highest)                                      | OS3   | OS1   | OS2   | OS4   | P5    | P2    | P4 (lowest) |
| 3.214   | 3.446 | 3.596 | 3.793 | 3.864 | 5.411 | 6.039 | 6.636       |
| Simulated dynamic ranking (insertion sort)        |       |       |       |       |       |       |             |
| P1 (highest)                                      | OS1   | OS3   | OS2   | OS4   | P5    | P4    | P2 (lowest) |
| 3.318   | 3.327 | 3.588 | 4.221 | 4.300 | 5.227 | 5.900 | 6.118       |
| Simulated dynamic ranking (binary insertion sort) |       |       |       |       |       |       |             |
| OS1 (highest)                                     | P1    | OS3   | OS2   | OS4   | P5    | P2    | P4 (lowest) |
| 2.924   | 3.045 | 3.303 | 3.812 | 4.267 | 5.833 | 5.903 | 6.882       |

Table 2: Human evaluation results

Table 1 gives the automatic scores for each of participating runs and four online systems. OS1 usually has the highest overall score (except BLEU), it also has the highest scores for ‘regulations’ (more formal texts), P1 scores are better for the news documents.

14 assessors were recruited for evaluation (participating team members and volunteers); the total volume of evaluation is 10,920 pairwise sentence comparisons. Table 2 presents the rankings of the participating systems using averaged ranks from the human evaluation. There is no statistically significant difference (using Welch’s t-test at  $p \leq 0.05$ ) in the overall ranks within the following groups: (OS1, OS3, P1) < (OS2, OS4) < P5 < (P2, P4). OS3 (mostly RBMT) belongs to the troika of leaders in human evaluation contrary to the results of its automatic scores (Table 1). Similarly, P5 is consistently ranked higher than P2 by the assessors, while the automatic scores suggest the opposite. This observation confirms the well-known fact that the automatic scores underestimate RBMT systems, e.g., (Béchar et al., 2012).

To investigate applicability of the automatic measures to the English-Russian language direction, we computed Spearman’s  $\rho$  correlation between the ranks given by the evaluators and by the respective measures. Because of the amount of variation for each measure on the sentence level, robust estimates, such as the median and the trimmed mean, are more informative than the mean, since they discard the outliers (Huber, 1996). The results are listed in Table 3. All mea-

asures exhibit reasonable correlation on the corpus level (330 sentences), but the sentence-level results are less impressive. While TER and GTM are known to provide better correlation with post-editing efforts for English (O’Brien, 2011), free word order and greater data sparseness on the sentence level makes TER much less reliable for Russian. METEOR (with its built-in Russian lemmatisation) and GTM offer the best correlation with human judgements.

The lower part of Table 2 also reports the results of simulated dynamic ranking (using the NIST rankings as the initial order for the sort operation). It resulted in a slightly different final ranking of the systems since we did not account for ties and ‘averaged ranks’. However, the ranking is practically the same up to the statistically significant rank differences in reference ranking (see above). The advantage is that it requires a significantly lower number of pairwise comparisons. Insertion sort yielded 5,131 comparisons (15.5 per sentence; 56% of exhaustive comparisons for 330 sentences and 8 systems); binary insertion sort yielded 4,327 comparisons (13.1 per sentence; 47% of exhaustive comparisons).

Out of the original set of 330 sentences for human evaluation, 60 sentences were evaluated by two annotators (which resulted in  $60 \cdot 28 = 1680$  pairwise comparisons), so we were able to calculate the standard Kohen’s  $\kappa$  and Krippendorff’s  $\alpha$  scores (Artstein and Poesio, 2008). The results of inter-annotator agreement are: percentage agreement 0.56,  $\kappa = 0.34$ ,  $\alpha = 0.48$ , which is simi-

| Metric | Sentence level |       |         | Corpus level |
|--------|----------------|-------|---------|--------------|
|        | Median         | Mean  | Trimmed |              |
| BLEU   | 0.357          | 0.298 | 0.348   | 0.833        |
| NIST   | 0.357          | 0.291 | 0.347   | 0.810        |
| Meteor | 0.429          | 0.348 | 0.393   | 0.714        |
| TER    | 0.214          | 0.186 | 0.204   | 0.619        |
| GTM    | 0.429          | 0.340 | 0.392   | 0.714        |

Table 3: Correlation to human judgements

lar to sentence ranking reported in other evaluation campaigns (Callison-Burch et al., 2012; Callison-Burch et al., 2011). It was interesting to see the agreement results distinguishing the top three systems against the rest, i.e. by ignoring the assessments for the pairs within each group,  $\alpha = 0.53$ , which indicates that the judges agree on the difference in quality between the top three systems and the rest. On the other hand, the agreement results within the top three systems are low:  $\kappa = 0.23$ ,  $\alpha = 0.33$ , which is again in line with the results for similar evaluations between closely performing systems (Callison-Burch et al., 2011).

## 5 Conclusions and future plans

This was the first attempt at making proper quantitative and qualitative evaluation of the English→Russian MT systems. In the future editions, we will be aiming at developing a new test corpus with a wider genre palette. We will probably complement the campaign with Russian→English translation direction. We hope to attract more participants, including international ones and plan to prepare a ‘light version’ for students and young researchers. We will also address the problem of tailoring automatic evaluation measures to Russian — accounting for complex morphology and free word order. To this end we will re-use human evaluation data gathered within the 2013 campaign. While the campaign was based exclusively on data in one language direction, the correlation results for automatic MT quality measures should be applicable to other languages with free word order and complex morphology.

We have made the corpus comprising the source sentences, their human translations, translations by participating MT systems and the human evaluation data publicly available.<sup>8</sup>

<sup>8</sup><http://romip.ru/mteval/>

## Acknowledgements

We would like to thank the translators, assessors, as well as Anna Tsygankova, Maxim Gubin, and Marina Nekrestyanova for project coordination and organisational help. Research on corpus preparation methods was supported by EU FP7 funding, contract No 251534 (HyghTra). Our special gratitude goes to Yandex and ABBYY who partially covered the expenses incurred on corpus translation. We’re also grateful to the anonymous reviewers for their useful comments.

## References

- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.
- Bogdan Babych, Anthony Hartley, Serge Sharoff, and Olga Mudraya. 2007. Assisting translators in indirect lexical transfer. In *Proc. of 45<sup>th</sup> ACL*, pages 739–746, Prague.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June.
- Hanna Béchar, Raphaël Rubino, Yifan He, Yanjun Ma, and Josef van Genabith. 2012. An evaluation of statistical post-editing systems applied to RBMT and SMT systems. In *Proceedings of COLING’12*, Mumbai.
- Igor Boguslavsky. 1995. A bi-directional Russian-to-English machine translation system (ETAP-3). In *Proceedings of the Machine Translation Summit V*, Luxembourg.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar F Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64. Association for Computational Linguistics.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology*, pages 138–145, San Diego, CA.

- Marcelo Federico, Mauro Cettolo, Luisa Bentivogli, Michael Paul, and Sebastian Stuker. 2012. Overview of the IWSLT 2012 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 12–34, Hong Kong, December.
- Peter J. Huber. 1996. *Robust Statistical Procedures*. Society for Industrial and Applied Mathematics.
- John Hutchins, editor. 2000. *Early years in machine translation: Memoirs and biographies of pioneers*. John Benjamins, Amsterdam, Philadelphia. <http://www.hutchinsweb.me.uk/EarlyYears-2000-TOC.htm>.
- Philippe Langlais. 2002. Improving a general-purpose statistical translation engine by terminological lexicons. In *Proceedings of Second international workshop on computational terminology (COMPUTERM 2002)*, pages 1–7, Taipei, Taiwan. <http://acl.ldc.upenn.edu/W/W02/W02-1405.pdf>.
- Sharon O’Brien. 2011. Towards predicting post-editing productivity. *Machine translation*, 25(3):197–215.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Thomas J. Watson Research Center.
- Serge Sharoff. 2010. In the garden and in the jungle: Comparing genres in the BNC and Internet. In Alexander Mehler, Serge Sharoff, and Marina Santini, editors, *Genres on the Web: Computational Models and Empirical Studies*, pages 149–166. Springer, Berlin/New York.
- Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268, Athens, Greece, March.
- Joseph Turian, Luke Shen, and I. Dan Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of Machine Translation Summit IX*, New Orleans, LA, USA, September.
- John S. White, Theresa O’Connell, and Francis O’Mara. 1994. The ARPA MT evaluation methodologies: Evolution, lessons, and further approaches. In *Proceedings of AMTA’94*, pages 193–205.

# IndoNet: A Multilingual Lexical Knowledge Network for Indian Languages

Brijesh Bhatt Lahari Poddar Pushpak Bhattacharyya

Center for Indian Language Technology

Indian Institute of Technology Bombay

Mumbai, India

{ brijesh, lahari, pb } @cse.iitb.ac.in

## Abstract

We present IndoNet, a multilingual lexical knowledge base for Indian languages. It is a linked structure of wordnets of 18 different Indian languages, Universal Word dictionary and the Suggested Upper Merged Ontology (SUMO). We discuss various benefits of the network and challenges involved in the development. The system is encoded in Lexical Markup Framework (LMF) and we propose modifications in LMF to accommodate Universal Word Dictionary and SUMO. This standardized version of lexical knowledge base of Indian Languages can now easily be linked to similar global resources.

## 1 Introduction

Lexical resources play an important role in natural language processing tasks. Past couple of decades have shown an immense growth in the development of lexical resources such as wordnet, Wikipedia, ontologies etc. These resources vary significantly in structure and representation formalism.

In order to develop applications that can make use of different resources, it is essential to link these heterogeneous resources and develop a common representation framework. However, the differences in encoding of knowledge and multilinguality are the major road blocks in development of such a framework. Particularly, in a multilingual country like India, information is available in many different languages. In order to exchange information across cultures and languages, it is essential to create an architecture to share various lexical resources across languages.

In this paper we present IndoNet, a lexical resource created by merging wordnets of 18 dif-

ferent Indian languages<sup>1</sup>, Universal Word Dictionary (Uchida et al., 1999) and an upper ontology, SUMO (Niles and Pease, 2001).

Universal Word (UW), defined by a headword and a set of restrictions which give an unambiguous representation of the concept, forms the vocabulary of Universal Networking Language. Suggested Upper Merged Ontology (SUMO) is the largest freely available ontology which is linked to the entire English WordNet (Niles and Pease, 2003). Though UNL is a graph based representation and SUMO is a formal ontology, both provide language independent conceptualization. This makes them suitable candidates for interlingua.

IndoNet is encoded in Lexical Markup Framework (LMF), an ISO standard (ISO-24613) for encoding lexical resources (Francopoulo et al., 2009).

The contribution of this work is twofold,

1. We propose an architecture to link lexical resources of Indian languages.
2. We propose modifications in Lexical Markup Framework to create a linked structure of multilingual lexical resources and ontology.

## 2 Related Work

Over the years wordnet has emerged as the most widely used lexical resource. Though most of the wordnets are built by following the standards laid by English Wordnet (Fellbaum, 1998), their conceptualizations differ because of the differences in lexicalization of concepts across languages. ‘Not

<sup>1</sup>Wordnets for Indian languages are developed in IndoWordNet project. Wordnets are available in following Indian languages: Assamese, Bodo, Bengali, English, Gujarati, Hindi, Kashmiri, Konkani, Kannada, Malayalam, Manipuri, Marathi, Nepali, Punjabi, Sanskrit, Tamil, Telugu and Urdu. These languages covers 3 different language families, Indo Aryan, Sino-Tibetan and Dravidian. <http://www.cfilt.iitb.ac.in/indowordnet>

only that, there exist lexical gaps where a word in one language has no correspondence in another language, but there are differences in the ways languages structure their words and concepts'. (Pease and Fellbaum, 2010).

The challenge of constructing a unified multilingual resource was first addressed in EuroWordNet (Vossen, 1998). EuroWordNet linked wordnets of 8 different European languages through a common interlingual index (ILI). ILI consists of English synsets and serves as a pivot to link other wordnets. While ILI allows each language wordnet to preserve its semantic structure, it has two basic drawbacks as described in Fellbaum and Vossen (2012),

1. An ILI tied to one specific language clearly reflects only the inventory of the language it is based on, and gaps show up when lexicons of different languages are mapped to it.
2. The semantic space covered by a word in one language often overlaps only partially with a similar word in another language, resulting in less than perfect mappings.

Subsequently in KYOTO project<sup>2</sup>, ontologies are preferred over ILI for linking of concepts of different languages. Ontologies provide language independent conceptualization, hence the linking remains unbiased to a particular language. Top level ontology SUMO is used to link common base concepts across languages. Because of the small size of the top level ontology, only a few wordnet synsets can be linked directly to the ontological concept and most of the synsets get linked through subsumption relation. This leads to a significant amount of information loss.

KYOTO project used Lexical Markup Framework (LMF) (Francopoulo et al., 2009) as a representation language. 'LMF provides a common model for the creation and use of lexical resources, to manage the exchange of data among these resources, and to enable the merging of a large number of individual electronic resources to form extensive global electronic resources' (Francopoulo et al., 2009). Soria et al. (2009) proposed WordNet-LMF to represent wordnets in LMF format. Henrich and Hinrichs (2010) have further modified Wordnet-LMF to accommodate lexical

relations. LMF also provides extensions for multilingual lexicons and for linking external resources, such as ontology. However, LMF does not explicitly define standards to share a common ontology among multilingual lexicons.

Our work falls in line with EuroWordNet and Kyoto except for the following key differences,

- Instead of using ILI, we use a 'common concept hierarchy' as a backbone to link lexicons of different languages.
- In addition to an upper ontology, a concept in common concept hierarchy is also linked to Universal Word Dictionary. Universal Word dictionary provides additional semantic information regarding argument types of verbs, that can be used to provide clues for selectional preference of a verb.
- We refine LMF to link external resources (e.g. ontologies) with multilingual lexicon and to represent Universal Word Dictionary.

### 3 IndoNet

IndoNet uses a common concept hierarchy to link various heterogeneous lexical resources. As shown in figure 1, concepts of different wordnets, Universal Word Dictionary and Upper Ontology are merged to form the common concept hierarchy. Figure 1 shows how concepts of English WordNet (EWN), Hindi Wordnet (HWN), upper ontology (SUMO) and Universal Word Dictionary (UWD) are linked through common concept hierarchy (CCH).

This section provides details of Common Concept Hierarchy and LMF encoding for different resources.

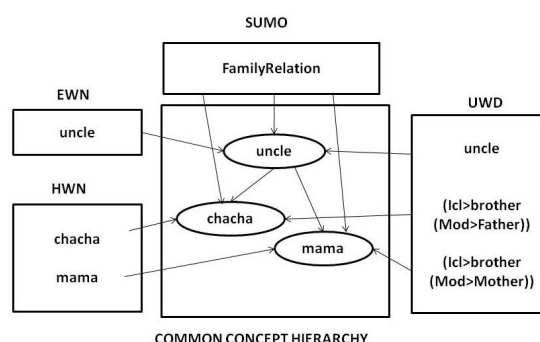


Figure 1: An Example of Indonet Structure

<sup>2</sup><http://kyoto-project.eu/xmlgroup.iit.cnr.it/kyoto/index.html>

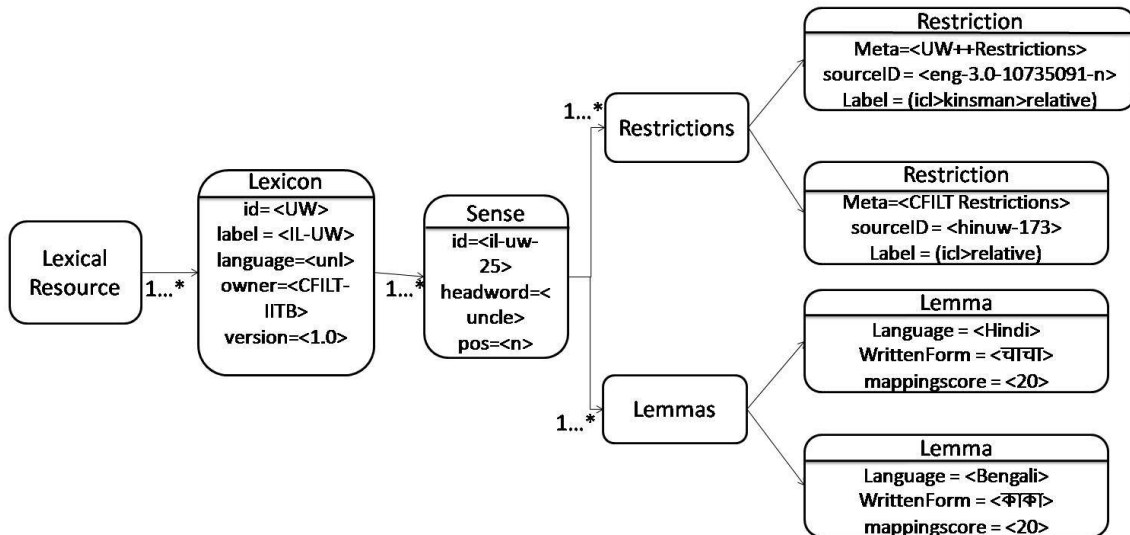


Figure 2: LMF representation for Universal Word Dictionary

### 3.1 Common Concept Hierarchy (CCH)

The common concept hierarchy is an abstract pivot index to link lexical resources of all languages. An element of a common concept hierarchy is defined as  $\langle \text{sinid}_1, \text{sinid}_2, \dots, \text{uwid}, \text{sumoid} \rangle$  where,  $\text{sinid}_i$  is synset id of  $i^{\text{th}}$  wordnet,  $\text{uw\_id}$  is universal word id, and  $\text{sumo\_id}$  is SUMO term id of the concept. Unlike ILI, the hypernymy-hyponymy relations from different wordnets are merged to construct the concept hierarchy. Each synset of wordnet is directly linked to a concept in ‘common concept hierarchy’.

### 3.2 LMF for Wordnet

We have adapted the Wordnet-LMF, as specified in Soria et al. (2009). However IndoWordnet encodes more lexical relations compared to EuroWordnet. We enhanced the Wordnet-LMF to accommodate the following relations: *antonym*, *gradation*, *hypernymy*, *meronymy*, *troponymy*, *entailment* and cross part of speech links for *ability* and *capability*.

### 3.3 LMF for Universal Word Dictionary

A Universal Word is composed of a headword and a list of restrictions, that provide unique meaning of the UW. In our architecture we allow each sense of a headword to have more than one set of restrictions (defined by different UW dictionaries) and be linked to lemmas of multiple languages with a confidence score. This allows us to merge multiple

UW dictionaries and represent it in LMF format. We introduce four new LMF classes; *Restrictions*, *Restriction*, *Lemmas* and *Lemma* and add new attributes; *headword* and *mapping score* to existing LMF classes.

Figure 2 shows an example of LMF representation of UW Dictionary. At present, the dictionary is created by merging two dictionaries, UW++ (Boguslavsky et al., 2007) and CFILT Hin-UW<sup>3</sup>. Lemmas from different languages are mapped to universal words and stored under the *Lemmas* class.

### 3.4 LMF to link ontology with Common Concept Hierarchy

Figure 3 shows an example LMF representation of CCH. The interlingual pivot is represented through *SenseAxis*. Concepts in different resources are linked to the *SenseAxis* in such a way that concepts linked to same *SenseAxis* convey the same *Sense*.

Using LMF class *MonolingualExternalRefs*, ontology can be integrated with a monolingual lexicon. In order to share an ontology among multilingual resources, we modify the original core package of LMF.

As shown in figure 3, a SUMO term is shared across multiple lexicons via the *SenseAxis*. SUMO is linked with concept hierarchy using the follow-

<sup>3</sup>[http://www.cfilt.iitb.ac.in/~hdict/webinterface\\_user/](http://www.cfilt.iitb.ac.in/~hdict/webinterface_user/)

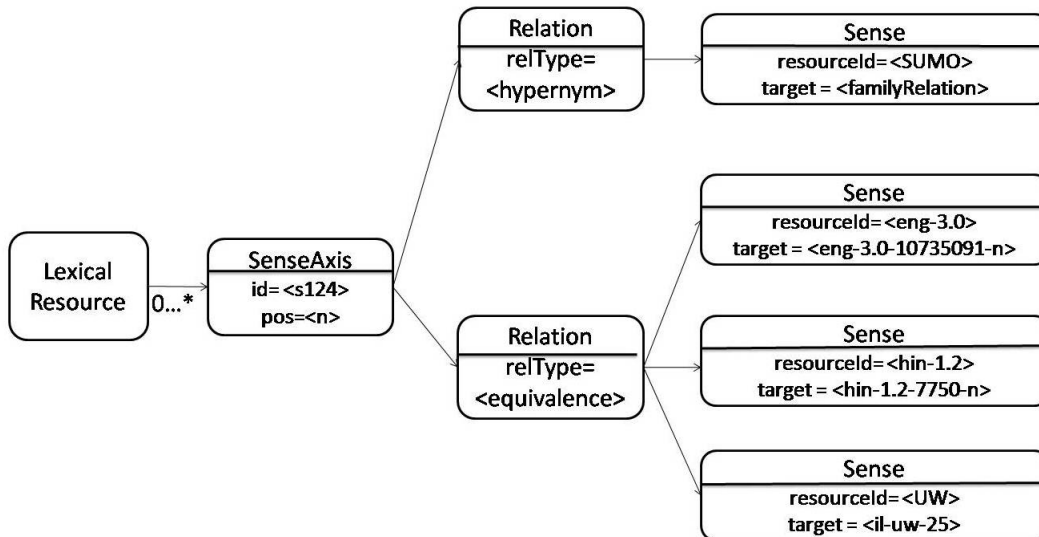


Figure 3: LMF representation for Common Concept Hierarchy

ing relations: *antonym*, *hypernym*, *instance* and *equivalent*. In order to support these relations, *Reltype* attribute is added to the interlingual *Sense* class.

#### 4 Observation

Table 1 shows *part of speech* wise status of linked concepts<sup>4</sup>. The concept hierarchy contains 53848 concepts which are shared among wordnets of Indian languages, SUMO and Universal Word Dictionary. Out of the total 53848 concepts, 21984 are linked to SUMO, 34114 are linked to HWN and 44119 are linked to UW. Among these, 12,254 are common between UW and SUMO and 21984 are common between wordnet and SUMO.

| POS       | HWN   | UW    | SUMO  | CCH   |
|-----------|-------|-------|-------|-------|
| adjective | 5532  | 2865  | 3140  | 5193  |
| adverb    | 380   | 2697  | 249   | 2813  |
| noun      | 25721 | 32831 | 16889 | 39620 |
| verb      | 2481  | 5726  | 1706  | 6222  |
| total     | 34114 | 44119 | 21984 | 53848 |

Table 1: Details of the concepts linked

This creates a multilingual semantic lexicon that captures semantic relations between concepts of different languages. Figure 1 demonstrates this with an example of ‘kinship relation’. As

<sup>4</sup>Table 1 shows data for Hindi Wordnet. Statistics for other wordnets can be found at [http://www.cfilt.iitb.ac.in/wordnet/webhwn/iwn\\_stats.php](http://www.cfilt.iitb.ac.in/wordnet/webhwn/iwn_stats.php)

shown in Figure 1, ‘uncle’ is an English language concept defined as ‘the brother of your father or mother’. Hindi has no concept equivalent to ‘uncle’ but there are two more specific concepts ‘kaka’, ‘brother of father.’ and ‘mama’, ‘brother of mother.’

The lexical gap is captured when these concepts are linked to CCH. Through CCH, these concepts are linked to SUMO term ‘FamilyRelation’ which shows relation between these concepts. Universal Word Dictionary captures exact relation between these concepts by applying restrictions [*chacha*] *uncle(icl>brother (mod>father))* and [*mama*] *uncle(icl>brother (mod>mother))*. This makes it possible to link concepts across languages.

#### 5 Conclusion

We have presented a multilingual lexical resource for Indian languages. The proposed architecture handles the ‘lexical gap’ and ‘structural divergence’ among languages, by building a common concept hierarchy. In order to encode this resource in LMF, we developed standards to represent UW in LMF.

IndoNet is emerging as the largest multilingual resource covering 18 languages of 3 different language families and it is possible to link or merge other standardized lexical resources with it.

Since Universal Word dictionary is an integral part of the system, it can be used for UNL based

Machine Translation tasks. Ontological structure of the system can be used for multilingual information retrieval and extraction.

In future, we aim to address ontological issues of the common concept hierarchy and integrate domain ontologies with the system. We are also aiming to develop standards to evaluate such multilingual resources and to validate axiomatic foundation of the same. We plan to make this resource freely available to researchers.

## Acknowledgements

We acknowledge the support of the Department of Information Technology (DIT), Ministry of Communication and Information Technology, Government of India and also of Ministry of Human Resource Development. We are also grateful to Study Group for Machine Translation and Automated Processing of Languages and Speech (GETALP) of the Laboratory of Informatics of Grenoble (LIG) for assisting us in building the Universal Word dictionary.

## References

- I. Boguslavsky, J. Bekios, J. Cardenosa, and C. Gallardo. 2007. Using Wordnet for Building an Interlingual Dictionary. In *Fifth International Conference Information Research and Applications*, (TECH 2007).
- Christiane Fellbaum and Piek Vossen. 2012. Challenges for a multilingual wordnet. *Language Resources and Evaluation*, 46(2):313–326, june.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Gil Francopoulo, Nuria Bel, Monte George, Nicoletta Calzolari, Monica Monachini, Mandy Pet, and Claudia Soria. 2009. Multilingual resources for NLP in the lexical markup framework (LMF). *Language Resources and Evaluation*.
- Verena Henrich and Erhard Hinrichs. 2010. Standardizing wordnets in the ISO standard LMF: Wordnet-LMF for GermaNet. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 456–464, Stroudsburg, PA, USA.
- Ian Niles and Adam Pease. 2001. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9, New York NY USA. ACM.
- Ian Niles and Adam Pease. 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings Of The 2003 International Conference On Information And Knowledge Engineering (Ike 03)*, Las Vegas, pages 412–416.
- Adam Pease and Christiane Fellbaum. 2010. Formal ontology as interlingua: The SUMO and WordNet linking project and global wordnet. In *Ontology and Lexicon, A Natural Language Processing perspective*, pages 25–35. Cambridge University Press.
- Claudia Soria, Monica Monachini, and Piek Vossen. 2009. Wordnet-LMF: fleshing out a standardized format for wordnet interoperability. In *Proceedings of the 2009 international workshop on Intercultural collaboration*, IWIC '09, pages 139–146, New York, NY, USA. ACM.
- H. Uchida, M. Zhu, and T. Della Senta. 1999. *The UNL- a Gift for the Millenium*. United Nations University Press, Tokyo.
- Piek Vossen, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Norwell, MA, USA.



# Building Japanese Textual Entailment Specialized Data Sets for Inference of Basic Sentence Relations

Kimi Kaneko<sup>†</sup> Yusuke Miyao<sup>‡</sup> Daisuke Bekki<sup>†</sup>

<sup>†</sup> Ochanomizu University, Tokyo, Japan

<sup>‡</sup> National Institute of Informatics, Tokyo, Japan

<sup>†</sup> {kaneko.kimi | bekki}@is.ocha.ac.jp

<sup>‡</sup> yusuke@nii.ac.jp

## Abstract

This paper proposes a methodology for generating specialized Japanese data sets for textual entailment, which consists of pairs decomposed into basic sentence relations. We experimented with our methodology over a number of pairs taken from the RITE-2 data set. We compared our methodology with existing studies in terms of agreement, frequencies and times, and we evaluated its validity by investigating recognition accuracy.

## 1 Introduction

In recognizing textual entailment (RTE), automated systems assess whether a human reader would consider that, given a snippet of text  $t_1$  and some unspecified (but restricted) world knowledge, a second snippet of text  $t_2$  is true. An example is given below.

Ex. 1) Example of a sentence pair for RTE

- Label: Y
- $t_1$ : Shakespeare wrote *Hamlet* and *Macbeth*.
- $t_2$ : Shakespeare is the author of *Hamlet*.

“Label” on line 1 shows whether textual entailment (TE) holds between  $t_1$  and  $t_2$ . The pair is labeled ‘Y’ if the pair exhibits TE and ‘N’ otherwise.

It is difficult for computers to make such assessments because pairs have multiple interrelated basic sentence relations (BSRs, for detailed information on BSRs, see section 3). Recognizing each BSRs in pairs exactly is difficult for computers. Therefore, we should generate specialized data sets consisting of  $t_1$ - $t_2$  pairs decomposed into BSRs and a methodology for generating such data sets since such data and methodologies for Japanese are unavailable at present.

This paper proposes a methodology for generating specialized Japanese data sets for TE that

consist of *monothematic*  $t_1$ - $t_2$  pairs (i.e., pairs in which only one BSR relevant to the entailment relation is highlighted and isolated). In addition, we compare our methodology with existing studies and analyze its validity.

## 2 Existing Studies

Sammons *et al.*(2010) point out that it is necessary to establish a methodology for decomposing pairs into chains of BSRs, and that establishing such methodology will enable understanding of how other existing studies can be combined to solve problems in natural language processing and identification of currently unsolvable problems. Sammons *et al.* experimented with their methodology over the RTE-5 data set and showed that the recognition accuracy of a system trained with their specialized data set was higher than that of the system trained with the original data set. In addition, Bentivogli *et al.*(2010) proposed a methodology for classifying more details than was possible in the study by Sammons *et al.*.

However, these studies were based on only English data sets. In this regard, the word-order rules and the grammar of many languages (such as Japanese) are different from those of English. We thus cannot assess the validity of methodologies for any Japanese data set because each language has different usages. Therefore, it is necessary to assess the validity of such methodologies with specialized Japanese data sets.

Kotani *et al.* (2008) generated specialized Japanese data sets for RTE that were designed such that each pair included only one BSR. However, in that approach the data set is generated artificially, and BSRs between pairs of real world texts cannot be analyzed.

We develop our methodology by generating specialized data sets from a collection of pairs from RITE-2<sup>1</sup> binary class (BC) subtask data sets containing sentences from Wikipedia. RITE-2 is

an evaluation-based workshop focusing on RTE. Four subtasks are available in RITE-2, one of which is the BC subtask whereby systems assess whether there is TE between t1 and t2. The reason why we apply our methodology to part of the RITE-2 BC subtask data set is that we can consider the validity of the methodology in view of the recognition accuracy by using the data sets generated in RITE-2 tasks, and that we can analyze BSRs in real texts by using sentence pairs extracted from Wikipedia.

### 3 Methodology

In this study, we extended and refined the methodology defined in Bentivogli *et al.*(2010) and developed a methodology for generating Japanese data sets broken down into BSRs and non-BSRs as defined below.

#### Basic sentence relations (BSRs):

- *Lexical*: Synonymy, Hypernymy, Entailment, Meronymy;
  - *Phrasal*: Synonymy, Hypernymy, Entailment, Meronymy, Nominalization, Corference;
  - *Syntactic*: Scrambling, Case alteration, Modifier, Transparent head, Clause, List, Apposition, Relative clause;
  - *Reasoning*: Temporal, Spatial, Quantity, Implicit relation, Inference;
- #### Non-basic sentence relations (non-BSRs) :
- *Disagreement*: Lexical, Phrasal, Modal, Modifier, Temporal, Spatial, Quantity;

Mainly, we used relations defined in Bentivogli *et al.*(2010) and divided **Synonymy**, **Hypernymy**, **Entailment** and **Meronymy** into *Lexical* and *Phrasal*. The differences between our study and Bentivogli *et al.*(2010) are as follows. **Demonymy** and **Statements** in Bentivogli *et al.*(2010) were not considered in our study because they were not necessary for Japanese data sets. In addition, **Scrambling**, **Entailment**, **Disagreement: temporal**, **Disagreement: spatial** and **Disagreement: quantity** were newly added in our study. **Scrambling** is a rule for changing the order of phrases and clauses. **Entailment** is a rule whereby the latter sentence is true whenever the former is true (e.g., “divorce” → “marry”). **Entailment** is a rule different from **Synonymy**, **Hypernymy** and **Meronymy**.

The rules for decomposition are schematized as follows:

<sup>1</sup><http://www.cl.ecei.tohoku.ac.jp/rite2/doku.php>

- Break down pairs into BSRs in order to bring t1 close to t2 gradually, as the interpretation of the converted sentence becomes wider
- Label each pair of BSRs or non-BSRs such that each pair is decomposed to ensure that there are not multiple BSRs

An example is shown below, where the underlined parts represent the revised points.

|                 |   |
|-----------------|---|
| t1 :            | シェイクスピアは <u>ハムレット</u> や <u>マクベス</u> を <u>書いた</u> 。  |
|                 | Shakespeare <sub>nom</sub> Hamlet <sub>com</sub> Macbeth <sub>acc</sub> write <sub>past</sub>       |
|                 | ‘Shakespeare wrote <u>Hamlet and Macbeth</u> .’   |
| [List]          | シェイクスピアは <u>ハムレット</u> を <u>書いた</u> 。  |
|                 | Shakespeare <sub>nom</sub> Hamlet <sub>acc</sub> write <sub>past</sub>                              |
|                 | ‘Shakespeare wrote <u>Hamlet</u> .’   |
| t2 : [Synonymy] | シェイクスピアは <u>ハムレットの</u> <u>作者</u> である。   |
|                 | : phrasal Shakespeare <sub>nom</sub> Hamlet <sub>gen</sub> author <sub>comp</sub> be <sub>cop</sub> |
|                 | ‘Shakespeare is <u>the author of Hamlet</u> .’  |

Table 1: Example of a pair with TE

An example of a pair without TE is shown below.

|                     |   |
|---------------------|---|
| t1 :                | ブルガリアは <u>ユーラシア大陸</u> に <u>ある</u> 。   |
|                     | Bulgaria <sub>nom</sub> Eurasia.continent <sub>dat</sub> be <sub>cop</sub>            |
|                     | ‘Bulgaria <u>is on the Eurasian continent</u> .’                                      |
| [Entailment]        | ブルガリアは <u>大陸国家</u> である。   |
|                     | : phrasal Bulgaria <sub>nom</sub> continental.state <sub>comp</sub> be <sub>cop</sub> |
|                     | ‘Bulgaria is a <u>continental state</u> .’  |
| t2 : [Disagreement] | ブルガリアは <u>島国</u> である。   |
|                     | : lexical Bulgaria <sub>nom</sub> island.country <sub>comp</sub> be <sub>cop</sub>    |
|                     | ‘Bulgaria is <u>an island country</u> .’  |

Table 2: Example of a pair without TE (Part 1)

To facilitate TE assessments like Table 3, non-BSR labels were used in decomposing pairs. In addition, we allowed labels to be used several times when some BSRs in a pair are related to ‘N’ assessments.

|                 |  |
|-----------------|--|
| t1 :            | ブルガリアは <u>ユーラシア大陸</u> に <u>ある</u> 。  |
|                 | Bulgaria <sub>nom</sub> Eurasia.continent <sub>dat</sub> be <sub>cop</sub>             |
|                 | ‘Bulgaria <u>is on the Eurasian continent</u> .’                                       |
| [Disagreement]  | ブルガリアは <u>ユーラシア大陸</u> に <u>ない</u> 。  |
|                 | : modal Bulgaria <sub>nom</sub> Eurasia.continent <sub>dat</sub> be <sub>cop-neg</sub> |
|                 | ‘Bulgaria is <u>not on the Eurasian continent</u> .’                                   |
| t2 : [Synonymy] | ブルガリアは <u>ヨーロッパ</u> に <u>属さない</u> 。  |
|                 | : lexical Bulgaria <sub>nom</sub> Europe <sub>dat</sub> belong <sub>cop-neg</sub>      |
|                 | ‘Bulgaria <u>does not belong to Europe</u> .’  |

Table 3: Example of a pair without TE (Part 2)

As mentioned above, the idea here is to decompose pairs in order to bring t1 closer to t2, the latter of which in principle has a wider semantic scope. We prohibited the conversion of t2 because it was possible to decompose the pairs such that they could be true even if there was no TE. Nevertheless, since it is sometimes easier to convert t2,

we allowed the conversion of t2 in only the case that t1 contradicted t2 and the scope of t2 did not overlap with that of t1 even if t2 was converted and TE would be unchanged. An example in case that we allowed to convert t2 is shown below. Bold-faced types in Table 4 shows that it becomes easy to compare t1 with t2 by converting to t2.

|                |      |   |
|----------------|------|---|
|                | t1 : | トムは 今日、朝食を 食べなかった。<br>Tom <sub>nom</sub> today breakfast <sub>acc</sub> eat <sub>past-neg</sub><br>'Tom didn't eat breakfast today.'          |
| [Scrambling]   |      | 今日、 トムは 朝食を 食べなかった。<br>today Tom <sub>nom</sub> breakfast <sub>acc</sub> eat <sub>past-neg</sub><br>'Today, Tom <b>didn't eat</b> breakfast.' |
|                | t2 : | 今朝、 トムは パンを 食べた。<br>this.morning Tom <sub>nom</sub> bread <sub>acc</sub> eat <sub>past</sub><br>'This morning, Tom ate bread and salad.'      |
| [Entailment]   |      | 今日、 トムは 朝食を 食べた。<br>: phrasal today Tom <sub>nom</sub> breakfast <sub>acc</sub> eat <sub>past</sub><br>'Today, Tom ate <b>breakfast</b> .'    |
| [Disagreement] |      | 今日、 トムは朝食を 食べた。<br>: modal 'Today, Tom <b>ate</b> breakfast.'   |

Table 4: Example of conversion of t2

## 4 Results

### 4.1 Comparison with Existing Studies

We applied our methodology to 173 pairs from the RITE-2 BC subtask data set. The pairs were decomposed by one annotator, and the decomposed pairs were assigned labels by two annotators. During labeling, we used the labels presented in Section 3 and “unknown” in cases where pairs could not be labeled. Our methodology was developed based on 112 pairs, and by using the other 61 pairs, we evaluated the inter-annotator agreement as well as the frequencies and times of decomposition.

The agreement for 241 monothematic pairs generated from 61 pairs amounted to 0.83 and was computed as follows. The kappa coefficient for them amounted 0.81.

$$Agreement = \frac{\text{“Agreed” labels}}{Total}^2$$

Bentivogli *et al.* (2010) reported an agreement rate of 0.78, although they computed the agreement by using the Dice coefficient (Dice, 1945), and therefore the results are not directly comparable to ours. Nevertheless, the close values suggest

<sup>2</sup>Because the “Agreed” pairs were clear to be classified as “Agreed”, where “Total” is the number of pairs labeled “Agreed” subtracted from the number of labeled pairs. “Agreed” labels is the number of pairs labeled “Agreed” subtract from the number of pairs with the same label assigned by the two annotators.

that our methodology is comparable to that in Bentivogli’s study in terms of agreement.

Table 5 shows the distribution of monothematic pairs with respect to original Y/N pairs.

| Original pairs | Monothematic pairs |    |       |
|----------------|--------------------|----|-------|
|                | Y                  | N  | Total |
| Y (32)         | 116                | –  | 116   |
| N (29)         | 96                 | 29 | 125   |
| Total (61)     | 212                | 29 | 241   |

Table 5: Distribution of monothematic pairs with respect to original Y/N pairs

When the methodology was applied to 61 pairs, a total of 241 and an average of 3.95 monothematic pairs were derived. The average was slightly greater than the 2.98 reported in (Bentivogli *et al.*, 2010). For pairs originally labeled ‘Y’ and ‘N’, an average of 3.62 and 3.31 monothematic pairs were derived, respectively. Both average values were slightly higher than the values of 3.03 and 2.80 reported in (Bentivogli *et al.*, 2010). On the basis of the small differences between the average values in our study and those in (Bentivogli *et al.*, 2010), we are justified in saying that our methodology is valid.

Table 6<sup>3</sup> shows the distribution of BSRs in t1-t2 pairs in an existing study and the present study. We can see from Table 6 that **Confidence** was seen more frequently in Bentivogli’s study than in our study, while **Entailment** and **Scrambling** were seen more frequently in our study. This demonstrates that differences between languages are relevant to the distribution and classification of BSRs.

An average of 5 and 4 original pairs were decomposed per hour in our study and Bentivogli’s study, respectively. This indicates that the complexity of our methodology is not much different from that in Bentivogli *et al.*(2010).

### 4.2 Evaluation of Accuracy in BSR

In the RITE-2 formal run<sup>4</sup>, 15 teams used our specialized data set for the evaluation of their systems. Table 7 shows the average of  $F_1$  scores<sup>5</sup> for each BSR.

**Scrambling** and **Modifier** yielded high scores (close to 90%). The score of **List** was also

<sup>3</sup>Because “lexical” and “phrasal” are classified together in Bentivogli *et al.*(2010), they are not shown separately in Table 6.

<sup>4</sup>In RITE-2, data generated by our methodology were released as “unit test data”.

<sup>5</sup>The traditional  $F_1$  score is the harmonic mean of precision and recall.

| BSR                           | Monothematic pairs       |     |    |               |     |    |
|-------------------------------|--------------------------|-----|----|---------------|-----|----|
|                               | Bentivogli <i>et al.</i> |     |    | Present study |     |    |
|                               | Total                    | Y   | N  | Total         | Y   | N  |
| Synonymy                      | 25                       | 22  | 3  | 45            | 45  | 0  |
| Hypernymy                     | 5                        | 3   | 2  | 5             | 5   | 0  |
| Entailment                    | -                        | -   | -  | 44            | 44  | 0  |
| Meronymy                      | 7                        | 4   | 3  | 1             | 1   | 0  |
| Nominalization                | 9                        | 9   | 0  | 1             | 1   | 0  |
| Corference                    | 49                       | 48  | 1  | 3             | 3   | 0  |
| Scrambling                    | -                        | -   | -  | 15            | 15  | 0  |
| Case alteration               | 7                        | 5   | 2  | 7             | 7   | 0  |
| Modifier                      | 25                       | 15  | 10 | 42            | 42  | 0  |
| Transparent head              | 6                        | 6   | 0  | 1             | 1   | 0  |
| Clause                        | 5                        | 4   | 1  | 14            | 14  | 0  |
| List                          | 1                        | 1   | 0  | 3             | 3   | 0  |
| Apposition                    | 3                        | 2   | 1  | 1             | 1   | 0  |
| Relative clause               | 1                        | 1   | 0  | 8             | 8   | 0  |
| Temporal                      | 2                        | 1   | 1  | 1             | 1   | 0  |
| Spatial                       | 1                        | 1   | 0  | 1             | 1   | 0  |
| Quantity                      | 6                        | 0   | 6  | 0             | 0   | 0  |
| Implicit relation             | 7                        | 7   | 0  | 18            | 18  | 0  |
| Inference                     | 40                       | 26  | 14 | 2             | 2   | 0  |
| Disagreement: lexical/phrasal | 3                        | 0   | 3  | 27            | 0   | 27 |
| Disagreement: modal           | 1                        | 0   | 1  | 1             | 0   | 1  |
| Disagreement: temporal        | -                        | -   | -  | 1             | 0   | 1  |
| Disagreement: spatial         | -                        | -   | -  | 0             | 0   | 0  |
| Disagreement: quantity        | -                        | -   | -  | 0             | 0   | 0  |
| Demonymy                      | 1                        | 1   | 0  | -             | -   | -  |
| Statements                    | 1                        | 1   | 0  | -             | -   | -  |
| total                         | 205                      | 157 | 48 | 241           | 212 | 29 |

Table 6: Distribution of BSRs in t1-t2 pairs in an existing study and in the present study using our methodology

| BSR                    | $F_1$ (%) | Monothematic Pairs | Miss |
|------------------------|-----------|--------------------|------|
| Scrambling             | 89.6      | 15                 | 4    |
| Modifier               | 88.8      | 42                 | 0    |
| List                   | 88.6      | 3                  | 0    |
| Temporal               | 85.7      | 1                  | 1    |
| Relative clause        | 85.4      | 8                  | 2    |
| Clause                 | 85.0      | 14                 | 2    |
| Hypernymy: lexical     | 85.0      | 5                  | 1    |
| Disagreement: phrasal  | 80.1      | 25                 | 0    |
| Case alteration        | 79.9      | 7                  | 2    |
| Synonymy: lexical      | 79.7      | 9                  | 6    |
| Transparent head       | 78.6      | 1                  | 2    |
| Implicit relation      | 75.7      | 18                 | 2    |
| Synonymy: phrasal      | 73.6      | 36                 | 9    |
| Corference             | 70.9      | 3                  | 1    |
| Entailment: phrasal    | 70.2      | 44                 | 7    |
| Disagreement: lexical  | 69.0      | 2                  | 0    |
| Meronymy: lexical      | 64.3      | 1                  | 1    |
| Nominalization         | 64.3      | 1                  | 0    |
| Apposition             | 50.0      | 1                  | 1    |
| Spatial                | 50.0      | 1                  | 1    |
| Inference              | 40.5      | 2                  | 2    |
| Disagreement: modal    | 35.7      | 1                  | 0    |
| Disagreement: temporal | 28.6      | 1                  | 1    |
| Total                  | -         | 241                | 41   |

Table 7: Average  $F_1$  scores in BSR and frequencies of misclassifications by annotators

nearly 90%, although the data sets included only 3 instances. These scores were high because pairs with these BSRs are easily recognized in terms of syntactic structure. By contrast, **Disagreement: temporal**, **Disagreement: modal**, **Inference**, **Spatial** and **Apposition** yielded low scores (less than 50%). The scores of **Disagreement: lexical**, **Nominalization** and **Disagreement: Meronymy** were about 50-70%. BSRs that yielded scores of less than 70% occurred less than 3 times, and those that yielded scores of not

more than 70% occurred 3 times or more, except for **Temporal** and **Transparent head**. Therefore, the frequencies of BSRs are related to  $F_1$  scores, and we should consider how to build systems that recognize infrequent BSRs accurately. In addition,  $F_1$  scores in **Synonymy: phrasal** and **Entailment: phrasal** are low, although these are labeled frequently. This is one possible direction of future work.

Table 7 also shows the number of pairs in BSR to which the two annotators assigned different labels. For example, one annotator labeled t2 [**Apposition**] while the other labeled t2 [**Spatial**] in the following pair:

Ex. 2) Example of a pair for RTE

- t1: Tokyo, the capital of Japan, is in Asia.
- t2: The capital of Japan is in Asia.

We can see from Table 7 that the  $F_1$  scores for BSRs, which are often assessed as different by different people, are generally low, except for several labels, such as **Synonymy: lexical** and **Scrambling**. For this reason, we can conjecture that cases in which computers experience difficulty determining the correct labels are correlated with cases in which humans also experience such difficulty.

## 5 Conclusions

This paper presented a methodology for generating Japanese data sets broken down into BSRs and Non-BSRs, and we conducted experiments in which we applied our methodology to 61 pairs extracted from the RITE-2 BC subtask data set. We compared our method with that of Bentivogli *et al.*(2010) in terms of agreement as well as frequencies and times of decomposition, and we obtained similar results. This demonstrated that our methodology is as feasible as Bentivogli *et al.*(2010) and that differences between languages emerge only as the different sets of labels and the different distributions of BSRs. In addition, 241 monothematic pairs were recognized by computers, and we showed that both the frequencies of BSRs and the rate of misclassification by humans are relevant to  $F_1$  scores.

Decomposition patterns were not empirically compared in the present study and will be investigated in future work. We will also develop an RTE inference system by using our specialized data set.

## References

- Bentivogli, L., Cabrio, E., Dagan, I., Giampiccolo, D., Leggio, M. L., Magnini, B. 2010. *Building Textual Entailment Specialized Data Sets: a Methodology for Isolating Linguistic Phenomena Relevant to Inference*. In Proceedings of LREC 2010, Valletta, Malta.
- Dagan, I, Glickman, O., Magnini, B. 2005. *Recognizing Textual Entailment Challenge*. In Proc. of the First PASCAL Challenges Workshop on RTE. Southampton, U.K.
- Kotani, M., Shibata, T., Nakata, T, Kurohashi, S. 2008. *Building Textual Entailment Japanese Data Sets and Recognizing Reasoning Relations Based on Synonymy Acquired Automatically*. In Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing, Tokyo, Japan.
- Magnini, B., Cabrio, E. 2009. *Combining Specialized Entailment Engines*. In Proceedings of LTC '09. Poznan, Poland.
- Dice, L. R. 1945. *Measures of the amount of ecologic association between species*. Ecology, 26(3):297-302.
- Mark Sammons, V.G.Vinod Vydiswaran, Dan Roth. 2010. "Ask not what textual entailment can do for you...". In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 1199-1208.

# Building Comparable Corpora Based on Bilingual LDA Model

Zede Zhu

University of Science and Technology  
of China, Institute of Intelligent Ma-  
chines Chinese Academy of Sciences  
Hefei, China  
zhuzede@mail.ustc.edu.cn

Miao Li, Lei Chen, Zhenxin Yang

Institute of Intelligent Machines Chinese  
Academy of Sciences  
Hefei, China  
mli@iim.ac.cn, alan.cl@163.com,  
xinzyang@mail.ustc.edu.cn

## Abstract

Comparable corpora are important basic resources in cross-language information processing. However, the existing methods of building comparable corpora, which use inter-translate words and relative features, cannot evaluate the topical relation between document pairs. This paper adopts the bilingual LDA model to predict the topical structures of the documents and proposes three algorithms of document similarity in different languages. Experiments show that the novel method can obtain similar documents with consistent topics own better adaptability and stability performance.

## 1 Introduction

Comparable corpora can be mined fine-grained translation equivalents, such as bilingual terminologies, named entities and parallel sentences, to support the bilingual lexicography, statistical machine translation and cross-language information retrieval (AbduI-Rauf et al., 2009). Comparable corpora are defined as pairs of monolingual corpora selected according to the criteria of content similarity but non-direct translation in different languages, which reduces limitation of matching source language and target language documents. Thus comparable corpora have the advantage over parallel corpora in which they are more up-to-date, abundant and accessible (Ji, 2009).

Many works, which focused on the exploitation of building comparable corpora, were proposed in the past years. Tao et al. (2005) acquired comparable corpora based on the truth that terms are inter-translation in different languages if they have similar frequency correlation at the same time periods. Talvensaaari et al. (2007) extracted appropriate keywords from the source language documents and translated them into the target language, which were regarded as the que-

ry words to retrieve similar target documents. Thuy et al. (2009) analyzed document similarity based on the publication dates, linguistic independent units, bilingual dictionaries and word frequency distributions. Otero et al. (2010) took advantage of the translation equivalents inserted in Wikipedia by means of interlanguage links to extract similar articles. Bo et al. (2010) proposed a comparability measure based on the expectation of finding the translation for each word.

The above studies rely on the high coverage of the original bilingual knowledge and a specific data source together with the translation vocabularies, co-occurrence information and language links. However, the severest problem is that they cannot understand semantic information. The new studies seek to match similar documents on topic level to solve the traditional problems. Preiss (2012) transformed the source language topical model to the target language and classified probability distribution of topics in the same language, whose shortcoming is that the effect of model translation seriously hampers the comparable corpora quality. Ni et al. (2009) adapted monolingual topic model to bilingual topic model in which the documents of a concept unit in different languages were assumed to share identical topic distribution. Bilingual topic model is widely adopted to mine translation equivalents from multi-language documents (Mimno et al., 2009; Ivan et al., 2011).

Based on the bilingual topic model, this paper predicts the topical structure of documents in different languages and calculates the similarity of topics over documents to build comparable corpora. The paper concretely includes: 1) Introduce the Bilingual LDA (Latent Dirichlet Allocation) model which builds comparable corpora and improves the efficiency of matching similar documents; 2) Design a novel method of TFIDF (Topic Frequency-Inverse Document Frequency) to enhance the distinguishing ability of topics from different documents; 3) Propose a tailored

method of conditional probability to calculate document similarity; 4) Address a language-independent study which isn't limited to a particular data source in any language.

## 2 Bilingual LDA Model

### 2.1 Standard LDA

LDA model (Blei et al., 2003) represents the latent topic of the document distribution by Dirichlet distribution with a  $K$ -dimensional implicit random variable, which is transformed into a complete generative model when  $\beta$  is exerted to Dirichlet distribution (Griffiths et al., 2004) (Shown in Fig. 1),

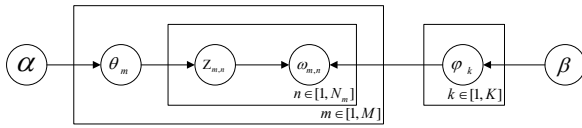


Figure 1: Standard LDA model

where  $\alpha$  and  $\beta$  denote the parameters distributed by Dirichlet;  $K$  denotes the topic numbers;  $\phi_k$  denotes the vocabulary probability distribution in the topic  $k$ ;  $M$  denotes the document number;  $\theta_m$  denotes the topic probability distribution in the document  $m$ ;  $N_m$  denotes the length of  $m$ ;  $Z_{m,n}$  and  $\omega_{m,n}$  denote the topic and the word in  $m$  respectively.

### 2.2 Bilingual LDA

Bilingual LDA is a bilingual extension of a standard LDA model. It takes advantage of the document alignment which shares the same topic distribution  $\theta_m$  and uses different word distributions for each topic (Shown in Fig. 2), where  $S$  and  $T$  denote source language and target language respectively.

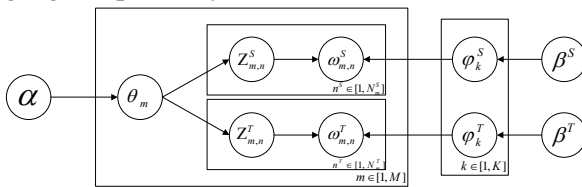


Figure 2: Bilingual LDA model

For each language  $l$  ( $l \in \{S, T\}$ ),  $Z_{m,n}^l$  and  $\omega_{m,n}^l$  are drawn using  $Z_{m,n}^l \sim P(Z_{m,n}^l | \theta_m)$  and  $\omega_{m,n}^l \sim P(\omega_{m,n}^l | Z_{m,n}^l, \phi^l)$ .

Giving the comparable corpora  $M$ , the distribution  $\phi_{k,v}$  can be obtained by sampling a new

token as word  $v$  from a topic  $k$ . For new collection of documents  $\tilde{M}$ , keeping  $\phi_{k,v}$ , the distribution  $\theta_{\tilde{m}^l, k}$  of sampling a topic  $k$  from document  $\tilde{m}$  can be obtained as follows:

$$P(Z_k | \tilde{m}^l) = \theta_{\tilde{m}^l, k} = \frac{n_{\tilde{m}^l}^{(k)} + \alpha_k}{\sum_{k=1}^K (n_{\tilde{m}^l}^{(k)} + \alpha_k)}, \quad (1)$$

where  $n_{\tilde{m}^l}^{(k)}$  denotes the total number of times that the document  $\tilde{m}$  is assigned to the topic  $k$ .

## 3 Building comparable corpora

Based on the bilingual LDA model, building comparable corpora includes several steps to generate the bilingual topic model  $\phi_{k,v}$  from the given bilingual corpora, predict the topic distribution  $\theta_{\tilde{m}^l, k}$  of the new documents, calculate the similarity of documents and select the largest similar document pairs. The key step is that the document similarity is calculated to align the source language document  $\tilde{m}^S$  with relevant target language document  $\tilde{m}^T$ .

As one general way of expressing similarity, the Kullback-Leibler (KL) Divergence is adopted to measure the document similarity by topic distributions  $\theta_{\tilde{m}^S, k}$  and  $\theta_{\tilde{m}^T, k}$  as follows:

$$\begin{aligned} Sim_{KL}(\tilde{m}^S, \tilde{m}^T) &= KL[P(Z | \tilde{m}^S), P(Z | \tilde{m}^T)] \\ &= \sum_{k=1}^K \left[ \theta_{\tilde{m}^S, k} \log \left( \theta_{\tilde{m}^S, k} / \theta_{\tilde{m}^T, k} \right) \right]. \end{aligned} \quad (2)$$

The remainder section focuses on other two methods of calculating document similarity.

### 3.1 Cosine Similarity

The similarity between  $\tilde{m}^S$  and  $\tilde{m}^T$  can be measured by Topic Frequency-Inverse Document Frequency. It gives high weights to the topic which appears frequently in a specific document and rarely appears in other documents. Then the relation between  $TFIDF_{\tilde{m}^S, Z}$  and  $TFIDF_{\tilde{m}^T, Z}$  is measured by Cosine Similarity (CS).

Similar to Term Frequency-Inverse Document Frequency (Manning et al., 1999), Topic Frequency (TF) denoting frequency of topic  $Z$  for the document  $\tilde{m}^l$  is denoted by  $P(Z | \tilde{m}^l)$ . Given a constant value  $\lambda$ , Inverse Document Frequency (IDF) is defined as the total number of documents  $|\tilde{M}|$  divided by the number of documents

$|\tilde{m}^l : P(Z | \tilde{m}^l) > \lambda|$  containing a particular topic, and then taking the logarithm, which is calculated as follows:

$$IDF = \log \frac{|\tilde{M}|}{1 + |\tilde{m}^l : P(Z | \tilde{m}^l) > \lambda|}. \quad (3)$$

The TFIDF is calculated as follows:

$$TFIDF = TF * IDF \\ = P(Z | \tilde{m}^l) \log \frac{|\tilde{M}|}{1 + |\tilde{m}^l : P(Z | \tilde{m}^l) > \lambda|}. \quad (4)$$

Thus, the TFIDF score of the topic  $k$  over document  $\tilde{m}^l$  is given by:

$$TFIDF_{\tilde{m}^l, k} \\ = P(Z_k | \tilde{m}^l) \log \frac{|\tilde{M}|}{1 + |\tilde{m}^l : P(Z_k | \tilde{m}^l) > \lambda|} \\ = \theta_{\tilde{m}^l, k} \log \frac{|\tilde{M}|}{1 + |\tilde{m}^l : \theta_{\tilde{m}^l, k} > \lambda|}. \quad (5)$$

The similarity between  $\tilde{m}^S$  and  $\tilde{m}^T$  is given by:

$$Sim_{CS}(\tilde{m}^S, \tilde{m}^T) = Cos(TFIDF_{\tilde{m}^S, Z}, TFIDF_{\tilde{m}^T, Z}) \\ = \frac{\sum_{k=1}^K TFIDF_{\tilde{m}^S, k} TFIDF_{\tilde{m}^T, k}}{\sqrt{\sum_{k=1}^K TFIDF_{\tilde{m}^S, k}^2} \sqrt{\sum_{k=1}^K TFIDF_{\tilde{m}^T, k}^2}}. \quad (6)$$

### 3.2 Conditional Probability

The similarity between  $\tilde{m}^S$  and  $\tilde{m}^T$  is defined as the Conditional Probability (CP) of documents  $P(\tilde{m}^T | \tilde{m}^S)$  that  $\tilde{m}^T$  will be generated as a response to the cue  $\tilde{m}^S$ .

$P(Z)$  as prior topic distribution is assumed a uniform distribution and satisfied the condition  $P(Z_k) = P(Z)$ . According to the total probability formula, the document  $\tilde{m}^T$  is given as:

$$P(\tilde{m}^T) = \sum_{k=1}^K P(\tilde{m}^T | Z_k) P(Z_k) \\ = P(Z) \sum_{k=1}^K P(\tilde{m}^T | Z_k). \quad (7)$$

Based on the Bayesian formula, the probability that a given topic  $Z$  is assigned to a particular target language document  $\tilde{m}^T$  is expressed:

$$P(\tilde{m}^T | Z) = \frac{P(Z | \tilde{m}^T) P(\tilde{m}^T)}{P(Z)} \\ = P(Z | \tilde{m}^T) \sum_{k=1}^K P(\tilde{m}^T | Z_k). \quad (8)$$

The sum of all probabilities  $\sum_{k=1}^K P(\tilde{m}^T | Z_k)$

that all topics  $Z$  are assigned to a particular document  $\tilde{m}^T$  is a constant  $\Omega$ , thus equation (8) is converted as follows:

$$P(\tilde{m}^T | Z) = \Omega P(Z | \tilde{m}^T). \quad (9)$$

According to the total probability formula, the similarity between  $\tilde{m}^S$  and  $\tilde{m}^T$  is given by:

$$Sim_{CP}(\tilde{m}^S, \tilde{m}^T) = P(\tilde{m}^T | \tilde{m}^S) \\ = \sum_{k=1}^K [P(\tilde{m}^T | Z_k) P(Z_k | \tilde{m}^S)] \\ = \Omega \sum_{k=1}^K [P(Z_k | \tilde{m}^T) P(Z_k | \tilde{m}^S)] \\ = \Omega \sum_{k=1}^K [\theta_{\tilde{m}^S, k} \theta_{\tilde{m}^T, k}]. \quad (10)$$

## 4 Experiments and analysis

### 4.1 Datasets and Evaluation

The experiments are conducted on two sets of Chinese-English comparable corpora. The first dataset is news corpora with 3254 comparable document pairs, from which 200 pairs are randomly selected as the test dataset *News-Test* and the remainder is the training dataset *News-Train*. The second dataset contains 8317 bilingual Wikipedia entry pairs, from which 200 pairs are randomly selected as the test dataset *Wiki-Test* and the remainder is the training dataset *Wiki-Train*. Then *News-Train* and *Wiki-Train* are merged into the training dataset *NW-Train*. And the hand-labeled gold standard namely *NW-Test* is composed of *News-Test* and *Wiki-Test*.

Braschler et al. (1998) used five levels of relevance to assess the alignments as follows: Same Story, Related Story, Shared Aspect, Common Terminology and Unrelated. The paper selects the documents with Same Story and Related Story as comparable corpora. Let  $C_p$  be the comparable corpora in the building result and  $C_l$  be the comparable corpora in the labeled result. The Precision ( $P$ ), Recall ( $R$ ) and F-measure ( $F$ ) are defined as:

$$P = \frac{|C_p \cap C_l|}{|C_p|}, R = \frac{|C_p \cap C_l|}{|C_l|}, F = \frac{2PR}{P + R}. \quad (11)$$

### 4.2 Results and analysis

Two groups of validation experiments are set with sampling frequency of 1000, parameter  $\alpha$



of 50/ $K$ , parameter  $\beta$  of 0.01 and topic number  $K$  of 600.

### Group 1: Different data source

We learn bilingual LDA models by taking different training datasets. The performance of three approaches (KL, CS and CP) is examined on different test datasets. Tab. 1 demonstrates these results with the winners for each algorithm in bold.

| <i>Train</i> | <i>Test</i> | <i>KL</i> |             | <i>CS</i> |             | <i>CP</i> |             |
|--------------|-------------|-----------|-------------|-----------|-------------|-----------|-------------|
|              |             | <i>P</i>  | <i>F</i>    | <i>P</i>  | <i>F</i>    | <i>P</i>  | <i>F</i>    |
| <i>News</i>  | <i>News</i> | 0.62      | 0.52        | 0.73      | 0.59        | 0.69      | 0.56        |
| <i>News</i>  | <i>Wiki</i> | 0.60      | 0.47        | 0.68      | 0.56        | 0.66      | 0.52        |
| <i>Wiki</i>  | <i>News</i> | 0.61      | 0.48        | 0.71      | 0.58        | 0.68      | 0.55        |
| <i>Wiki</i>  | <i>Wiki</i> | 0.63      | 0.50        | 0.75      | 0.60        | 0.71      | 0.59        |
| <i>NW</i>    | <i>NW</i>   | 0.66      | <b>0.55</b> | 0.76      | <b>0.62</b> | 0.73      | <b>0.60</b> |

Table 1: Sensitivity of Data Source

The results indicate the robustness and effectiveness of these algorithms. The performance of algorithms on *Wiki-Train* is much better than *News-Train*. The main reason is that *Wiki-Train* is an extensive snapshot of human knowledge which can cover most topics talked in *News-Train*. The probability of vocabularies among the test dataset which have not appeared in the training data is very low. And then the document topic can effectively concentrate all the vocabularies' expressions. The topic model slightly faces with the problem of knowledge migration issue, so the performance of the topic model trained by *Wiki-Train* shows a slight decline in the experiments on *News-Test*.

CS shows the strongest performance among the three algorithms to recognize the document pairs with similar topics. CP has almost equivalent performance with CS. Comparing the equation (5) and (6) with (10), we can find out that CP is similar to a simplified CS. CP can improve the operating efficiency and decrease the performance. The performance achieved by KL is the weakest and there is a large gap between KL and others. In addition, the shortage of KL is that when the exchange between the source language and the target language documents takes place, different evaluations will occur in the same document pairs.

### Group 2: Existing Methods Comparison

We adopt the *NW-Train* and *NW-Test* as training set and test set respectively, and utilize the CS algorithm to calculate the document similarity to

verify the excellence of methods in the study. Then we compare its performance with the existing representative approaches proposed by Thuy et al. (2009) and Preiss (2012) (Shown in Tab. 2).

| <i>Algorithm</i> | <i>P</i> | <i>R</i> | <i>F</i>    |
|------------------|----------|----------|-------------|
| <i>Thuy</i>      | 0.45     | 0.32     | 0.37        |
| <i>Preiss</i>    | 0.67     | 0.44     | 0.53        |
| <i>CS</i>        | 0.76     | 0.53     | <b>0.62</b> |

Table 2: Existing Methods Comparison

The table shows CS outperforms other algorithms, which indicates that bilingual LDA is valid to construct comparable corpora. Thuy et al. (2009) matches similar documents in the view of inter-translated vocabulary and co-occurrence information features, which cannot understand the content effectively. Preiss (2012) uses monolingual training dataset to generate topic model and translates source language topic model into target language topic model respectively. Yet the translation accuracy constrains the matching effectiveness of similar documents, and the cosine similarity is directly used to calculate document-topic similarity failing to highlight the topic contributions of different documents.

## 5 Conclusion

This study proposes a new method of using bilingual topic to match similar documents. When CS is used to match the documents, TFIDF is proposed to enhance the topic discrepancies among different documents. The method of CP is also addressed to measure document similarity.

Experimental results show that the matching algorithm is superior to the existing algorithms. It can utilize comprehensively large scales of document information in training set to avoid the information deficiency of the document itself and over-reliance on bilingual knowledge. The algorithm makes the document match on the basis of understanding the document. This study does not calculate similar contents existed in the monolingual documents. However, a large number of documents in the same language describe the same event. We intend to incorporate monolingual document similarity into bilingual topics analysis to match multi-documents in different languages perfectly.

### Acknowledgments

The work is supported by the National Natural Science Foundation of China under No. 61070099 and the project of MSR-CNIC Windows Azure Theme.

## References

- AbduI-Rauf S, Schwenk H. On the use of comparable corpora to improve SMT performance[C]//Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009: 16-23.
- Ji H. Mining name translations from comparable corpora by creating bilingual information networks[C] // Proceedings of BUCC 2009. Suntec, Singapore, 2009: 34-37.
- Braschler M, Schauble P. Multilingual Information Retrieval based on document alignment techniques[C] // Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries. Heraklion, Greece. 1998: 183-197.
- Tao Tao, Chengxiang Zhai. Mining comparable bilingual text corpora for cross-language information integration[C] // Proceedings of ACM SIGKDD, Chicago, Illinois, USA. 2005:691-696.
- Talvensaari T, Laurikkala J, Jarvelin K, et al. Creating and Exploiting a Comparable Corpus in Cross-Language Information Retrieval[J]. ACM Transactions on Information Systems. 2007, 25(1): 322-334.
- Thuy Vu, Ai Ti Aw, Min Zhang. Feature-based method for document alignment in comparable news corpora[C] // Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, Greece. 2009: 843-851.
- Otero P G, L'opez I G. Wikipedia as Multilingual Source of Comparable Corpora[C] // Proceedings of the 3rd Workshop on BUCC, LREC2010. Malta. 2010: 21-25.
- Li B, Gaussier E. Improving corpus comparability for bilingual lexicon extraction from comparable corpora[C]//Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010: 644-652.
- Judita Preiss. Identifying Comparable Corpora Using LDA[C]//2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Montreal, Canada, June 3-8, 2012: 558-562.
- Mimno D, Wallach H, Naradowsky J et al. Polylingual topic models[C]//Proceedings of the EMNLP. Singapore, 2009: 880-889.
- Vulic I, De Smet W, Moens M F, et al. Identifying word translations from comparable corpora using latent topic models[C]//Proceedings of ACL. 2011: 479-484.
- Ni X, Sun J T, Hu J, et al. Mining multilingual topics from wikipedia[C]//Proceedings of the 18th international conference on World wide web. ACM, 2009: 1155-1156.
- Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.
- Griffiths T L, Steyvers M. Finding scientific topics[J]. Proceedings of the National academy of Sciences of the United States of America, 2004, 101: 5228-5235.
- Manning C D, Schütze H. Foundations of statistical natural language processing[M]. MIT press, 1999.

# Using Lexical Expansion to Learn Inference Rules from Sparse Data

Oren Melamud<sup>§</sup>, Ido Dagan<sup>§</sup>, Jacob Goldberger<sup>†</sup>, Idan Szpektor<sup>‡</sup>

<sup>§</sup> Computer Science Department, Bar-Ilan University

<sup>†</sup> Faculty of Engineering, Bar-Ilan University

<sup>‡</sup> Yahoo! Research Israel

{melamuo, dagan, goldbej}@{cs, cs, eng}.biu.ac.il

idan@yahoo-inc.com

## Abstract

Automatic acquisition of inference rules for predicates is widely addressed by computing distributional similarity scores between vectors of argument words. In this scheme, prior work typically refrained from learning rules for low frequency predicates associated with very sparse argument vectors due to expected low reliability. To improve the learning of such rules in an unsupervised way, we propose to lexically expand sparse argument word vectors with semantically similar words. Our evaluation shows that lexical expansion significantly improves performance in comparison to state-of-the-art baselines.

## 1 Introduction

The benefit of utilizing template-based inference rules between predicates was demonstrated in NLP tasks such as Question Answering (QA) (Ravichandran and Hovy, 2002) and Information Extraction (IE) (Shinyama and Sekine, 2006). For example, the inference rule ‘ $X \text{ treat } Y \rightarrow X \text{ relieve } Y$ ’, between the templates ‘ $X \text{ treat } Y$ ’ and ‘ $X \text{ relieve } Y$ ’ may be useful to identify the answer to “Which drugs relieve stomach ache?”.

The predominant unsupervised approach for learning inference rules between templates is via distributional similarity (Lin and Pantel, 2001; Ravichandran and Hovy, 2002; Szpektor and Dagan, 2008). Specifically, each argument slot in a template is represented by an argument vector, containing the words (or terms) that instantiate this slot in all of the occurrences of the template in a learning corpus. Two templates are then deemed semantically similar if the argument vectors of their corresponding slots are similar.

Ideally, inference rules should be learned for all templates that occur in the learning corpus.

However, many templates are rare and occur only few times in the corpus. This is a typical NLP phenomenon that can be associated with either a small learning corpus, as in the cases of domain specific corpora and resource-scarce languages, or with templates with rare terms or long multi-word expressions such as ‘ $X \text{ be also a risk factor to } Y$ ’ or ‘ $X \text{ finish second in } Y$ ’, which capture very specific meanings. Due to few occurrences, the slots of rare templates are represented with very sparse argument vectors, which in turn lead to low reliability in distributional similarity scores.

A common practice in prior work for learning predicate inference rules is to simply disregard templates below a minimal frequency threshold (Lin and Pantel, 2001; Kotlerman et al., 2010; Dinu and Lapata, 2010; Ritter et al., 2010). Yet, acquiring rules for rare templates may be beneficial both in terms of coverage, but also in terms of more accurate rule application, since rare templates are less ambiguous than frequent ones.

We propose to improve the learning of rules between infrequent templates by expanding their argument vectors. This is done via a “dual” distributional similarity approach, in which we consider two words to be similar if they instantiate similar sets of templates. We then use these similarities to expand the argument vector of each slot with words that were identified as similar to the original arguments in the vector. Finally, similarities between templates are computed using the expanded vectors, resulting in a ‘smoothed’ version of the original similarity measure.

Evaluations on a rule application task show that our lexical expansion approach significantly improves the performance of the state-of-the-art DIRT algorithm (Lin and Pantel, 2001). In addition, our approach outperforms a similarity measure based on vectors of latent topics instead of word vectors, a common way to avoid sparseness issues by means of dimensionality reduction.

## 2 Technical Background

The distributional similarity score for an inference rule between two predicate templates, *e.g.* ‘*X resign Y → X quit Y*’, is typically computed by measuring the similarity between the argument vectors of the corresponding *X* slots and *Y* slots of the two templates. To this end, first the argument vectors should be constructed and then a similarity measure between two vectors should be provided. We note that we focus here on binary templates with two slots each, but this approach can be applied to any template.

A common starting point is to compute a co-occurrence matrix *M* from a learning corpus. *M*’s rows correspond to the template slots and the columns correspond to the various terms that instantiate the slots. Each entry  $M_{i,j}$ , *e.g.*  $M_{x \text{ quit}, \text{John}}$ , contains a count of the number of times the term *j* instantiated the template slot *i* in the corpus. Thus, each row  $M_{i,*}$  corresponds to an argument vector for slot *i*. Next, some function of the counts is used to assign weights to all  $M_{i,j}$  entries. In this paper we use pointwise mutual information (PMI), which is common in prior work (Lin and Pantel, 2001; Szepkator and Dagan, 2008).

Finally, rules are assessed using some similarity measure between corresponding argument vectors. The state-of-the-art DIRT algorithm (Lin and Pantel, 2001) uses the highly cited *Lin* similarity measures (Lin, 1998) to score rules between binary templates as follows:

$$Lin(v, v') = \frac{\sum_{w \in v \cap v'} [v(w) + v'(w)]}{\sum_{w \in v \cup v'} [v(w) + v'(w)]} \quad (1)$$

$$\begin{aligned} &DIRT(l \rightarrow r) \\ &= \sqrt{Lin(v_{l:x}, v_{r:x}) \cdot Lin(v_{l:y}, v_{r:y})} \quad (2) \end{aligned}$$

where *v* and *v'* are two argument vectors, *l* and *r* are the templates participating in the inference rule and  $v_{l:x}$  corresponds to the argument vector of slot *X* of template *l*, etc. While the original DIRT algorithm utilizes the *Lin* measure, one can replace it with any other vector similarity measure.

A separate line of research for word similarity introduced directional similarity measures that have a bias for identifying generalization/specification relations, *i.e.* relations between predicates with narrow (or specific) semantic meanings to predicates with broader meanings

inferred by them (unlike the symmetric *Lin*). One such example is the *Cover* measure (Weeds and Weir, 2003):

$$Cover(v, v') = \frac{\sum_{w \in v \cap v'} [v(w)]}{\sum_{w \in v \cup v'} [v(w)]} \quad (3)$$

As can be seen, in the core of the *Lin* and *Cover* measures, as well as in many other well known distributional similarity measures such as Jaccard, Dice and Cosine, stand the number of shared arguments vs. the total number of arguments in the two vectors. Therefore, when the argument vectors are sparse, containing very few non-zero features, these scores become unreliable and volatile, changing greatly with every inclusion or exclusion of a single shared argument.

## 3 Lexical Expansion Scheme

We wish to overcome the sparseness issues in rare feature vectors, especially in cases where argument vectors of semantically similar predicates comprise similar but not exactly identical arguments. To this end, we propose a three step scheme. First, we learn lexical expansion sets for argument words, such as the set {*euros, money*} for the word *dollars*. Then we use these sets to expand the argument word vectors of predicate templates. For example, given the template ‘*X can be exchanged for Y*’, with the following argument words instantiating slot *X* {*dollars, gold*}, and the expansion set above, we would expand the argument word vector to include all the following words {*dollars, euros, money, gold*}. Finally, we use the expanded argument word vectors to compute the scores for predicate inference rules with a given similarity measure.

When a template is instantiated with an observed word, we expect it to also be instantiated with semantically similar words such as the ones in the expansion set of the observed word. We “blame” the lack of such template occurrences only on the size of the corpus and the sparseness phenomenon in natural languages. Thus, we utilize our lexical expansion scheme to synthetically add these expected but missing occurrences, effectively smoothing or generalizing over the explicitly observed argument occurrences. Our approach is inspired by query expansion (Voorhees, 1994) in Information Retrieval (IR), as well as by the recent lexical expansion framework proposed in (Biemann and Riedl, 2013), and the work by

Miller et al. (2012) on word sense disambiguation. Yet, to the best of our knowledge, this is the first work that applies lexical expansion to distributional similarity feature vectors. We next describe our scheme in detail.

### 3.1 Learning Lexical Expansions

We start by constructing the co-occurrence matrix  $M$  (Section 2), where each entry  $M_{t:s,w}$  indicates the number of times that word  $w$  instantiates slot  $s$  of template  $t$  in the learning corpus, denoted by ' $t:s$ ', where  $s$  can be either X or Y.

In traditional distributional similarity, the rows  $M_{t:s,*}$  serve as argument vectors of template slots. However, to learn expansion sets we take a “dual” view and consider each matrix column  $M_{*:s,w}$  (denoted  $v_w$ ) as a feature vector for the argument word  $w$ . Under this view, templates (or more specifically, template slots) are the features. For instance, for the word *dollars* the respective feature vector may include entries such as ‘*X can be exchanged for*’, ‘*can be exchanged for Y*’, ‘*purchase Y*’ and ‘*sell Y*’.

We next learn an expansion set per each word  $w$  by computing the distributional similarity between the vectors of  $w$  and any other argument word  $w'$ ,  $\text{sim}(v_w, v_{w'})$ . Then we take the  $N$  most similar words as  $w$ 's expansion set with degree  $N$ , denoted by  $L_w^N = \{w'_1, \dots, w'_N\}$ . Any similarity measure could be used, but as our experiments show, different measures generate sets with different properties, and some may be fitter for argument vector expansion than others.

### 3.2 Expanding Argument Vectors

Given a row count vector  $M_{t:s,*}$  for slot  $s$  of template  $t$ , we enrich it with expansion sets as follows. For each  $w$  in  $M_{t:s,*}$ , the original count in  $v_{t:s}(w)$  is redistributed equally between itself and all words in  $w$ 's expansion set, i.e. all  $w' \in L_w^N$ , (possibly yielding fractional counts) where  $N$  is a global parameter of the model. Specifically, the new count that is assigned to each word  $w$  is its remaining original count after it has been redistributed (or zero if no original count), plus all the counts that were distributed to it from other words.

Next, PMI weights are recomputed according to the new counts, and the resulting expanded vector is denoted by  $v_{t:s}^+$ . Similarity between template slots is now computed over the expanded vectors instead of the original ones, e.g.  $\text{Lin}(v_{t:x}^+, v_{r:x}^+)$ .

## 4 Experimental Settings

We constructed a relatively small learning corpus for investigating the sparseness issues of such corpora. To this end, we used a random sample from the large scale web-based ReVerb corpus<sup>1</sup> (Fader et al., 2011), comprising tuple extractions of predicate templates with their argument instantiations. We applied some clean-up preprocessing to these extractions, discarding stop words, rare words and non-alphabetical words that instantiated either the X or the Y argument slots. In addition, we discarded templates that co-occur with less than 5 unique argument words in either of their slots, assuming that such few arguments cannot convey reliable semantic information, even with expansion. Our final corpus consists of around 350,000 extractions and 14,000 unique templates. In this corpus around one third of the extractions refer to templates that co-occur with at most 35 unique arguments in both their slots.

We evaluated the quality of inference rules using the dataset constructed by Zeichner et al. (2012)<sup>2</sup>, which contains about 6,500 manually annotated template rule applications, each labeled as correct or not. For example, ‘*The game develop eye-hand coordination*  $\rightarrow$  *The game launch eye-hand coordination*’ is a rule application in this dataset of the rule ‘*X develop Y*  $\rightarrow$  *X launch Y*’, labeled as incorrect, and ‘*Captain Cook sail to Australia*  $\rightarrow$  *Captain Cook depart for Australia*’ is a rule application of the rule ‘*X sail to Y*  $\rightarrow$  *X depart for Y*’, labeled as correct. Specifically, we induced two datasets from Zeichner et al.’s dataset, denoted *DS-5-35* and *DS-5-50*, which consist of all rule applications whose templates are present in our learning corpus and co-occurred with at least 5 and at most 35 and 50 unique argument words in both their slots, respectively. *DS-5-35* includes 311 rule applications (104 correct and 207 incorrect) and *DS-5-50* includes 502 rule applications (190 correct and 312 incorrect).

Our evaluation task is to rank all rule applications in each test set based on the similarity scores of the applied rules. Optimal performance would rank all correct rule applications above the incorrect ones. As a baseline for rule scoring we

<sup>1</sup><http://reverb.cs.washington.edu/>

<sup>2</sup><http://www.cs.biu.ac.il/nlp/downloads/annotation-rule-application.htm>

used the DIRT algorithm scheme, denoted *DIRT-LE-None*. We then compared between the performance of this baseline and its expanded versions, testing two similarity measures for generating the expansion sets of arguments: *Lin* and *Cover*. We denote these expanded methods *DIRT-LE-SIM-N*, where *SIM* is the similarity measure used to generate the expansion sets and *N* is the lexical expansion degree, e.g. *DIRT-LE-Lin-2*.

We remind the reader that our scheme utilizes two similarity measures. The first measure assesses the similarity between the argument vectors of the two templates in the rule. This measure is kept constant in our experiments and is identical to DIRT’s similarity measure (*Lin*).<sup>3</sup> The second measure assesses the similarity between words and is used for the lexical expansion of argument vectors. Since this is the research goal of this paper, we experimented with two different measures for lexical expansion: a symmetric measure (*Lin*) and an asymmetric measure (*Cover*). To this end we evaluated their effect on DIRT’s rule ranking performance and compared them to a vanilla version of DIRT without lexical expansion.

As another baseline, we follow Dinu and Lapata (2010) inducing LDA topic vectors for template slots and computing predicate template inference rule scores based on similarity between these vectors. We use standard hyperparameters for learning the LDA model (Griffiths and Steyvers, 2004). This method is denoted *LDA-K*, where *K* is the number of topics in the model.

## 5 Results

We evaluated the performance of each tested method by measuring Mean Average Precision (MAP) (Manning et al., 2008) of the rule application ranking computed by this method. In order to compute MAP values and corresponding statistical significance, we randomly split each test set into 30 subsets. For each method we computed Average Precision on every subset and then took the average as the MAP value. We varied the degree of the lexical expansion in our model and the number of topics in the topic model baseline to analyze their effect on the performance of these methods on our datasets. We note that in our model a greater degree of lexical expansion cor-

<sup>3</sup>Experiments with *Cosine* as the template similarity measure instead of *Lin* for both DIRT and its expanded versions yielded similar results. We omit those for brevity.

responds to more aggressive smoothing (or generalization) of the explicitly observed data, while the same goes for a lower number of topics in the topic model. The results on *DS-5-35* and *DS-5-50* are illustrated in Figure 1.

The most dramatic improvement over the baselines is evident in *DS-5-35*, where *DIRT-LE-Cover-2* achieves a MAP score of 0.577 in comparison to 0.459 achieved by its *DIRT-LE-None* baseline. This is indeed the dataset where we expected expansion to affect most due the extreme sparseness of argument vectors. Both *DIRT-LE-Cover-N* and *DIRT-LE-Lin-N* outperform *DIRT-LE-None* for all tested values of *N*, with statistical significance via a paired t-test at  $p < 0.05$  for *DIRT-LE-Cover-N* where  $1 \leq N \leq 5$ , and  $p < 0.01$  for *DIRT-LE-Cover-2*. On *DS-5-50*, improvement over the *DIRT-LE-None* baseline is still significant with both *DIRT-LE-Cover-N* and *DIRT-LE-Lin-N* outperforming *DIRT-LE-None*. *DIRT-LE-Cover-N* again performs best and achieves a relative improvement of over 10% with statistical significance at  $p < 0.05$  for  $2 \leq N \leq 3$ .

The above shows that expansion is effective for improving rule learning between infrequent templates. Furthermore, the fact that *DIRT-LE-Cover-N* outperforms *DIRT-LE-Lin-N* suggests that using directional expansions, which are biased to *generalizations* of the observed argument words, e.g. *vehicle* as an expansion for *car*, is more effective than using symmetrically related words, such as *bicycle* or *automobile*. This conclusion appears also to be valid from a semantic reasoning perspective, as given an observed predicate-argument occurrence, such as ‘*drive car*’ we can more likely infer that a presumed occurrence of the same predicate with a *generalization* of the argument, such as ‘*drive vehicle*’, is valid, i.e. ‘*drive car*  $\rightarrow$  *drive vehicle*’. On the other hand while ‘*drive car*  $\rightarrow$  *drive automobile*’ is likely to be valid, ‘*drive car*  $\rightarrow$  *drive bicycle*’ and ‘*drive vehicle*  $\rightarrow$  *drive bicycle*’ are not.

Figure 1 also depicts the performance of LDA as a vector smoothing approach. *LDA-K* outperforms the *DIRT-LE-None* baseline under *DS-5-35* but with no statistical significance. Under *DS-5-50* *LDA-K* performs worst, slightly outperforming *DIRT-LE-None* only for  $K=450$ . Furthermore, under both datasets, *LDA-K* is outperformed by *DIRT-LE-Cover-N*. These results indicate that LDA is less effective than our expansion approach.

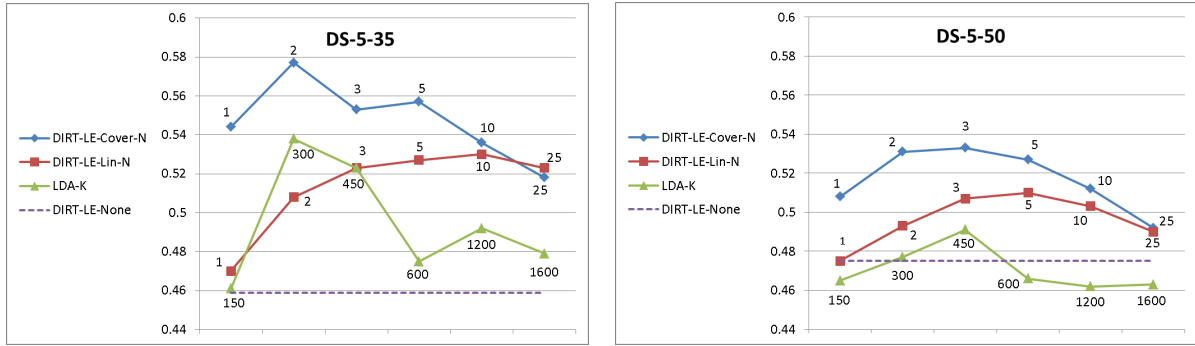


Figure 1: MAP scores on *DS-5-35* and *DS-5-50* for the original DIRT scheme, denoted *DIRT-LE-None*, and for the compared smoothing methods as follows. DIRT with varied degrees of lexical expansion is denoted as *DIRT-LE-Lin-N* and *DIRT-LE-Cover-N*. The topic model with varied number of topics is denoted as *LDA-K*. Data labels indicate the expansion degree ( $N$ ) or the number of LDA topics ( $K$ ), depending on the tested method.

One reason may be that in our model, every expansion set may be viewed as a cluster around a specific word, an outstanding difference in comparison to topics, which provide a global partition over all words. We note that performance improvement of singleton document clusters over global partitions was also shown in IR (Kurland and Lee, 2009).

In order to further illustrate our lexical expansion scheme we focus on the rule application ‘*Captain Cook sail to Australia*  $\rightarrow$  *Captain Cook depart for Australia*’, which is labeled as correct in our test set and corresponds to the rule ‘ $X$  sail to  $Y \rightarrow X$  depart for  $Y$ ’. There are 30 words instantiating the  $X$  slot of the predicate ‘*sail to*’ in our learning corpus including {*Columbus, emperor, James, John, trader*}. On the other hand, there are 18 words instantiating the  $X$  slot of the predicate ‘*depart for*’ including {*Amanda, Jerry, Michael, mother, queen*}. While semantic similarity between these two sets of words is evident, they share no words in common, and therefore the original DIRT algorithm, *DIRT-LE-None*, wrongly assigns a zero score to the rule.

The following are descriptions of some of the argument word expansions performed by *DIRT-LE-Cover-2* (using the notation  $L_w^N$  defined in Section 3.1) for the  $X$  slot of ‘*sail to*’  $L_{John}^2 = \{mr., dr.\}$ ,  $L_{trader}^2 = \{people, man\}$ , and for the  $X$  slot of ‘*depart for*’,  $L_{Michael}^2 = \{John, mr.\}$ ,  $L_{mother}^2 = \{people, woman\}$ . Given these expansions the two slots now share the following words {*mr., people, John*} and the rule score becomes positive.

It is also interesting to compare the expansions

performed by *DIRT-LE-Lin-2* to the above. For instance in this case  $L_{mother}^2 = \{father, sarah\}$ , which does not identify *people* as a shared argument for the rule.

## 6 Conclusions

We propose to improve the learning of inference rules between infrequent predicate templates with sparse argument vectors by utilizing a novel scheme that lexically expands argument vectors with semantically similar words. Similarities between argument words are discovered using a dual distributional representation, in which templates are the features.

We tested the performance of our expansion approach on rule application datasets that were biased towards rare templates. Our evaluation showed that rule learning with expanded vectors outperformed the baseline learning with original vectors. It also outperformed an LDA-based similarity model that overcomes sparseness via dimensionality reduction.

In future work we plan to investigate how our scheme performs when integrated with manually constructed resources for lexical expansion, such as WordNet (Fellbaum, 1998).

## Acknowledgments

This work was partially supported by the Israeli Ministry of Science and Technology grant 3-8705, the Israel Science Foundation grant 880/12, and the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

## References

- Chris Biemann and Martin Riedl. 2013. Text: Now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modeling*, 1(1).
- Georgiana Dinu and Mirella Lapata. 2010. Topic models for meaning similarity in context. In *Proceedings of COLING: Posters*.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.
- Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.
- Oren Kurland and Lillian Lee. 2009. Clusters, language models, and ad hoc information retrieval. *ACM Transactions on Information Systems (TOIS)*, 27(3):13.
- Dekang Lin and Patrick Pantel. 2001. DIRT – discovery of inference rules from text. In *Proceedings of KDD*.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*.
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.
- Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. *Proceedings of COLING, Mumbai, India*.
- Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*.
- Alan Ritter, Oren Etzioni, et al. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of ACL*.
- Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of NAACL*.
- Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of COLING*.
- Ellen M Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of SIGIR*.
- Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of EMNLP*.
- Naomi Zeichner, Jonathan Berant, and Ido Dagan. 2012. Crowdsourcing inference-rule evaluation. In *Proceedings of ACL (short papers)*.



# Mining Equivalent Relations from Linked Data

Ziqi Zhang<sup>1</sup>

Eva Blomqvist<sup>2</sup>

Anna Lisa Gentile<sup>1</sup>

Isabelle Augenstein<sup>1</sup>

Fabio Ciravegna<sup>1</sup>

<sup>1</sup> Department of Computer Science,  
University of Sheffield, UK

<sup>2</sup> Department of Computer and Information  
Science, Linköping University, Sweden

{z.zhang, a.l.gentile, i.augenstein,

f.ciravegna}@dcs.shef.ac.uk, eva.blomqvist@liu.se

## Abstract

Linking heterogeneous resources is a major research challenge in the Semantic Web. This paper studies the task of mining equivalent relations from Linked Data, which was insufficiently addressed before. We introduce an unsupervised method to measure equivalency of relation pairs and cluster equivalent relations. Early experiments have shown encouraging results with an average of 0.75~0.87 precision in predicting relation pair equivalency and 0.78~0.98 precision in relation clustering.

## 1 Introduction

*Linked Data* defines best practices for exposing, sharing, and connecting data on the Semantic Web using uniform means such as URIs and RDF. It constitutes the conjunction between the Web and the Semantic Web, balancing the richness of semantics offered by Semantic Web with the easiness of data publishing. For the last few years Linked Open Data has grown to a gigantic knowledge base, which, as of 2013, comprised 31 billion triples in 295 datasets<sup>1</sup>.

A major research question concerning Linked Data is linking heterogeneous resources, the fact that publishers may describe analogous information using different vocabulary, or may assign different identifiers to the same referents. Among such work, many study mappings between ontology concepts and data instances (e.g., Isaac et al, 2007; Mi et al., 2009; Le et al., 2010; Duan et al., 2012). An insufficiently addressed problem is linking heterogeneous relations, which is also widely found in data and can cause problems in information retrieval (Fu et al., 2012). Existing work in linking relations typically employ string similarity metrics or semantic similarity mea-

asures that require a-priori domain knowledge and are limited in different ways (Zhong et al., 2002; Volz et al., 2009; Han et al., 2011; Zhao and Ichise, 2011; Zhao and Ichise, 2012).

This paper introduces a novel method to discover equivalent groups of relations for Linked Data concepts. It consists of two components: 1) a measure of equivalency between pairs of relations of a concept and 2) a clustering process to group equivalent relations. The method is unsupervised; completely data-driven requiring no a-priori domain knowledge; and also language independent. Two types of experiments have been carried out using two major Linked Data sets: 1) evaluating the precision of predicting equivalency of relation pairs and 2) evaluating the precision of clustering equivalent relations. Preliminary results have shown encouraging results as the method achieves between 0.75~0.85 precision in the first set of experiments while 0.78~0.98 in the latter.

## 2 Related Work

Research on linking heterogeneous ontological resources mostly addresses mapping classes (or concepts) and instances (Isaac et al, 2007; Mi et al., 2009; Le et al., 2010; Duan et al., 2012; Schopman et al., 2012), typically based on the notions of similarity. This is often evaluated by string similarity (e.g. string edit distance), semantic similarity (Budanitsky and Hirst, 2006), and distributional similarity based on the overlap in data usage (Duan et al., 2012; Schopman et al., 2012). There have been insufficient studies on mapping relations (or properties) across ontologies. Typical methods make use of a combination of string similarity and semantic similarity metrics (Zhong et al., 2002; Volz et al., 2009; Han et al., 2011; Zhao and Ichise, 2012). While string similarity fails to identify equivalent relations if their lexicalizations are distinct, semantic similarity often depends on taxonomic structures

---

<sup>1</sup> <http://lod-cloud.net/state/>

in existing ontologies (Budanitsky and Hirst, 2006). Unfortunately many Linked Data instances use relations that are invented arbitrarily or originate in rudimentary ontologies (Parundekar et al., 2012). Distributional similarity has also been used to discover equivalent or similar relations. Mauge et al. (2012) extract product properties from an e-commerce website and align equivalent properties using a supervised maximum entropy classification method. We study linking relations on Linked Data and propose an unsupervised method. Fu et al. (2012) identify similar relations using the overlap of the subjects of two relations and the overlap of their objects. On the contrary, we aim at identifying strictly equivalent relations rather than similarity in general. Additionally, the techniques introduced our work is also related to work on aligning multilingual Wikipedia resources (Adar et al., 2009; Bouma et al., 2009) and semantic relatedness (Budanitsky and Hirst, 2006).

### 3 Method

Let  $t$  denote a 3-tuple (triple) consisting of a subject ( $t_s$ ), predicate ( $t_p$ ) and object ( $t_o$ ). Linked Data resources are *typed* and its type is called *class*. We write  $type(t_s) = c$  meaning that  $t_s$  is of class  $c$ .  $p$  denotes a relation and  $r_p$  is a set of triples whose  $t_p = p$ , i.e.,  $r_p = \{t \mid t_p = p\}$ .

Given a specific class  $c$ , and its pairs of relations ( $p, p'$ ) such that  $r_p = \{t \mid t_p = p, type(t_s) = c\}$  and  $r_{p'} = \{t \mid t_p = p', type(t_s) = c\}$ , we measure the equivalency of  $p$  and  $p'$  and then cluster equivalent relations. The equivalency is calculated locally (within same class  $c$ ) rather than globally (across all classes) because two relations can have identical meaning in specific class context but not necessarily so in general. For example, for the class *Book*, the relations *dbpp:title* and *foaf:name* are used with the same meaning, however for *Actor*, *dbpp:title* is used interchangeably with awards *dbpp:awards* (e.g., Oscar best actor).

In practice, given a class  $c$ , our method starts with retrieving all  $t$  from a Linked Data set where  $type(t_s) = c$ , using the universal query language SPARQL with any SPARQL data endpoint. This data is then used to measure equivalency for each pair of relations (Section 3.1). The equivalence scores are then used to group relations in equivalent clusters (Section 3.2).

#### 3.1 Measure of equivalence

The equivalence for each distinct pair of relations depends on three components.

**Triple overlap** evaluates the degree of overlap<sup>2</sup> in terms of the usage of relations in triples. Let  $SO(p)$  be the collection of subject-object pairs from  $r_p$  and  $SO_{int}$  the intersection

$$SO_{int}(p, p') = SO(r_p) \cap SO(r_{p'}) \quad [1]$$

then the triple overlap  $TO(p, p')$  is calculated as

$$MAX\left\{\frac{|SO_{int}(r_p, r_{p'})|}{|r_p|}, \frac{|SO_{int}(r_p, r_{p'})|}{|r_{p'}|}\right\} \quad [2]$$

Intuitively, if two relations  $p$  and  $p'$  have a large overlap of subject-object pairs in their data instances, they are likely to have identical meaning. The *MAX* function allows addressing infrequently used, but still equivalent relations (i.e., where the overlap covers most triples of an infrequently used relation but only a very small proportion of a much more frequently used).

**Subject agreement** While triple overlap looks at the data in general, subject agreement looks at the overlap of subjects of two relations, and the degree to which these subjects have overlapping objects. Let  $S(p)$  return the set of subjects of relation  $p$ , and  $O(p|s)$  returns the set of objects of relation  $p$  whose subjects are  $s$ , i.e.:

$$O(p|s) = O(r_p|s) = \{t_o \mid t_p = p, t_s = s\} \quad [3]$$

we define:

$$S_{int}(p, p') = S(r_p) \cap S(r_{p'}) \quad [4]$$

$$\alpha = \frac{\sum_{s \in S_{int}(p, p')} 1, \text{if } |O(p|s) \cap O(p'|s)| > 0}{|S_{int}(p, p')|} \quad [5]$$

$$\beta = \sqrt{|S_{int}(p, p')| / |S(p) \cup S(p')|} \quad [6]$$

then the agreement  $AG(p, p')$  is

$$AG(p, p') = \alpha \cdot \beta \quad [7]$$

Equation [5] counts the number of overlapping subjects whose objects have at least one overlap. The higher the value of  $\alpha$ , the more the two relations “agree” in terms of their shared subjects. For each shared subject of  $p$  and  $p'$  we count 1 if they have at least 1 overlapping object and 0 otherwise. This is because both  $p$  and  $p'$  can be *1:many* relations and a low overlap value could mean that one is densely populated while the other is not, which does not necessarily mean they do not “agree”. Equation [6] evaluates the degree to which two relations share the same set of subjects. The agreement  $AG(p, p')$  balances the two factors by taking the product. As a result,

<sup>2</sup> In this paper overlap is based on “exact” match.

relations that have high level of agreement will have more subjects in common, and higher proportion of shared subjects with shared objects.

**Cardinality ratio** is a ratio between cardinality of the two relations. Cardinality of a relation  $CD(p)$  is calculated based on data:

$$CD(p) = \frac{|r_p|}{|S(r_p)|} \quad [8]$$

and the cardinality ratio is calculated as

$$CDR(p, p') = \frac{MIN\{CD(p), CD(p')\}}{MAX\{CD(p), CD(p')\}} \quad [9]$$

The final **equivalency measure** integrates all the three components to return a value in  $[0, 2]$ :

$$E(p, p') = \frac{TO(p, p') + AG(p, p')}{CDR(p, p')} \quad [10]$$

The measure will favor two relations that have similar cardinality.

### 3.2 Clustering

We apply the measure to every pair of relations of a concept, and keep those with a non-zero equivalence score. The goal of clustering is to create groups of equivalent relations based on the pair-wise equivalence scores. We use a simple rule-based agglomerative clustering algorithm for this purpose. First, we rank all relation pairs by their equivalence score, then we keep a pair if (i) its score and (ii) the number of triples covered by each relation are above a certain threshold,  $T_{minEqvl}$  and  $T_{minTP}$  respectively. Each pair forms an initial cluster. To merge clusters, given an existing cluster  $c$  and a new pair  $(p, p')$  where either  $p \in c$  or  $p' \in c$ , the pair is added to  $c$  if  $E(p, p')$  is close (as a fractional number above the threshold  $T_{minEqvlRel}$ ) to the average scores of all connected pairs in  $c$ . This preserves the strong connectivity in a cluster. This is repeated until no merge action is taken. Adjusting these thresholds allows balancing between precision and recall.

## 4 Experiment Design

To our knowledge, there is no publically available gold standard for relation equivalency using Linked Data. We randomly selected 21 concepts (Figure 1) from the DBpedia ontology (v3.8):

Actor, Aircraft, Airline, Airport, Automobile, Band, BasketballPlayer, Book, Bridge, Comedian, Film, Hospital, Magazine, Museum, Restaurant, Scientist, TelevisionShow, TennisPlayer, Theatre, University, Writer

Figure 1. Concepts selected for evaluation.

We apply our method to each concept to discover clusters of equivalent relations, using as *SPARQL endpoint* both DBpedia<sup>3</sup> and Sindice<sup>4</sup> and report results separately. This is to study how the method performs in different conditions: on one hand on a smaller and cleaner dataset (DBpedia); on the other hand on a larger and multi-lingual dataset (Sindice) to also test cross-lingual capability of our method. We chose relatively low *thresholds*, i.e.  $T_{minEqvl}=0.1$ ,  $T_{minTP}=0.01\%$  and  $T_{minEqvlRel}=0.6$ , in order to ensure high recall without sacrificing much precision.

Four human annotators manually annotated the output for each concept. For this preliminary evaluation, we have limited the amount of annotations to a maximum of 100 top scoring pairs of relations per concept, resulting in 16~100 pairs per concept (avg. 40) for DBpedia experiment and 29~100 pairs for Sindice (avg. 91). The annotators were asked to rate each edge in each cluster with -1 (wrong), 1 (correct) or 0 (cannot decide). Pairs with 0 are ignored in the evaluation (about 12% for DBpedia; and 17% for Sindice mainly due to unreadable encoded URLs for certain languages). To evaluate cross-lingual pairs, we asked annotators to use translation tools. Inter-Annotator-Agreement (observed IAA) is shown in Table 1. Also using this data, we derived a gold standard for clustering based on edge connectivity and we evaluate (i) the *precision* of top  $n\%$  ( $p@n\%$ ) ranked equivalent relation pairs and (ii) the *precision* of clustering for each concept.

|         | Mean | High | Low  |
|---------|------|------|------|
| DBpedia | 0.79 | 0.89 | 0.72 |
| Sindice | 0.75 | 0.82 | 0.63 |

Table 1. IAA on annotating pair equivalency

So far the output of 13 concepts has been annotated. This dataset<sup>5</sup> contains  $\approx 1800$  relation pairs and is larger than the one by Fu et al. (2012). Annotation process shows that over 75% of relation pairs in the Sindice experiment contain non-English relations and mostly are cross-lingual. We used this data to report performance, although the method has been applied to all the 21 concepts, and the complete results can be visualized at our demo website link. Some examples are shown in Figure 2.

<sup>3</sup> <http://dbpedia.org/sparql>

<sup>4</sup> <http://sparql.sindice.com/>

<sup>5</sup> <http://staffwww.dcs.shef.ac.uk/people/Z.Zhang/resources/paper/acl2013short/web/>

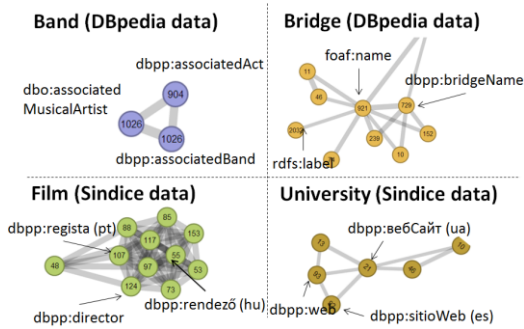


Figure 2. Examples of visualized clusters

## 5 Result and Discussion

Figure 3 shows  $p@n\%$  for pair equivalency<sup>6</sup> and Figure 4 shows clustering precision.

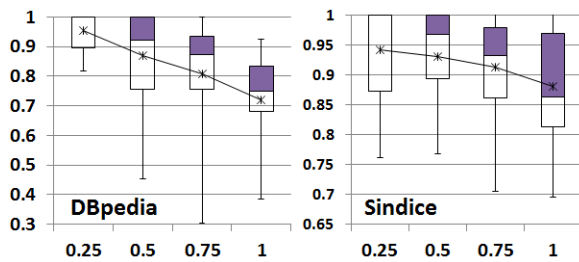


Figure 3.  $p@n\%$ . The box plots show the ranges of precision at each  $n\%$ ; the lines show the average.

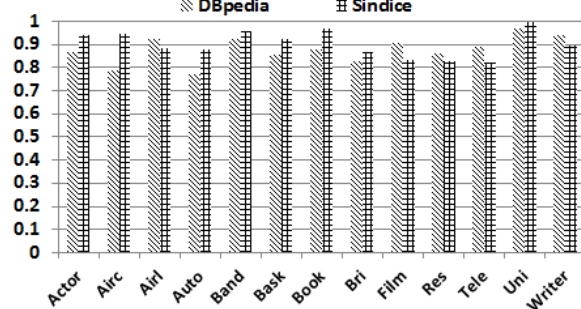


Figure 4. Clustering precision

As it is shown in Figure 2, Linked Data relations are often heterogeneous. Therefore, finding equivalent relations to improve coverage is important. Results in Figure 3 show that in most cases the method identifies equivalent relations with high precision. It is effective for both single- and cross-language relation pairs. The worst performing case for DBpedia is *Aircraft* (for all  $n\%$ ), mostly due to duplicating numeric valued objects of different relations (e.g., weight, length, capacity). The decreasing precision with respect to  $n\%$  suggests the measure effectively ranks correct pairs to the top. This is a useful feature from IR point of view. Figure 4 shows that the method effectively clusters equivalent relations with very high precision: 0.8–0.98 in most cases.

Overall we believe the results of this early proof-of-concept are encouraging. As a concrete example to compare against Fu et al. (2012), for *BasketballPlayer*, our method creates separate clusters for relations meaning “draft team” and “former team” because although they are “similar” they are not “equivalent”.

We noticed that annotating equivalent relations is a non-trivial task. Sometimes relations and their corresponding schemata (if any) are poorly documented and it is impossible to understand the meaning of relations (e.g., due to acronyms) and even very difficult to reason based on data. Analyses of the evaluation output show that errors are typically found between highly similar relations, or whose object values are numeric types. In both cases, there is a very high probability of having a high overlap of subject-object pairs between relations. For example, for *Aircraft*, the relations *dbpp:heightIn* and *dbpp:weight* are predicted to be equivalent because many instances have the same numeric value for the properties. Another example are the *Airport* properties *dbpp:runwaySurface*, *dbpp:r1Surface*, *dbpp:r2Surface* etc., which according to the data seem to describe the construction material (e.g., concrete, asphalt) of airport runways. The relations are semantically highly similar and the object values have a high overlap. A potential solution to such issues is incorporating ontological knowledge if available. For example, if an ontology defines the two distinct properties of *Airport* without explicitly defining an “equivalence” relation between them, they are unlikely to be equivalent even if the data suggests the opposite.

## 6 Conclusion

This paper introduced a data-driven, unsupervised and domain and language independent method to learn equivalent relations for Linked Data concepts. Preliminary experiments show encouraging results as it effectively discovers equivalent relations in both single- and multi-lingual settings. In future, we will revise the equivalence measure and also experiment with clustering algorithms such as (Beeferman et al., 2000). We will also study the contribution of individual components of the measure in such task. Large scale comparative evaluations (incl. recall) are planned and this work will be extended to address other tasks such as ontology mapping and ontology pattern mining (Nuzzolese et al., 2011).

<sup>6</sup> Per-concept results are available on our website.

## Acknowledgement

Part of this research has been sponsored by the EPSRC funded project LODIE: Linked Open Data for Information Extraction, EP/J019488/1. Additionally, we also thank the reviewers for their valuable comments given for this work.

## References

- Eytan Adar, Michael Skinner, Daniel Weld. 2009. *Information Arbitrage across Multilingual Wikipedia*. Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 94 – 103.
- Gosse Bouma, Sergio Duarte, Zahurul Islam. 2009. Cross-lingual Alignment and Completion of Wikipedia Templates. Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies, pp. 61 – 69
- Doug Beeferman, Adam Berger. 2000. Agglomerative clustering of a search engine query log. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 407-416.
- Alexander Budanitsky and Graeme Hirst. 2006. *Evaluating WordNet-based Measures of Semantic Distance*. Computational Linguistics, 32(1), pp.13-47.
- Songyun Duan, Achille Fokoue, Oktie Hasanzadeh, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J. Ward. 2012. *Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing*. ISWC 2012, pp. 46 – 64
- Linyun Fu, Haofen Wang, Wei Jin, Yong Yu. 2012. *Towards better understanding and utilizing relations in DBpedia*. Web Intelligence and Agent Systems , Volume 10 (3)
- Andrea Nuzzolese, Aldo Gangemi, Valentina Presutti, Paolo Ciancarini. 2011. *Encyclopedic Knowledge Patterns from Wikipedia Links*. Proceedings of the 10th International Semantic Web Conference, pp. 520-536
- Lushan Han, Tim Finin and Anupam Joshi. 2011. *GoRelations: An Intuitive Query System for DBpedia*. Proceedings of the Joint International Semantic Technology Conference
- Antoine Isaac, Lourens van der Meij, Stefan Schlobach, Shenghui Wang. 2007. *An empirical study of instance-based ontology matching*. Proceedings of the 6th International Semantic Web Conference and the 2nd Asian conference on Asian Semantic Web Conference, pp. 253-266
- Ngoc-Thanh Le, Ryutaro Ichise, Hoai-Bac Le. 2010. *Detecting hidden relations in geographic data*. Proceedings of the 4th International Conference on Advances in Semantic Processing, pp. 61 – 68
- Karin Mauge, Khash Rohanimanesh, Jean-David Ruvini. 2012. *Structuring E-Commerce Inventory*. Proceedings of ACL2012, pp. 805-814
- Jinhua Mi, Huajun Chen, Bin Lu, Tong Yu, Gang Pan. 2009. *Deriving similarity graphs from open linked data on semantic web*. Proceedings of the 10th IEEE International Conference on Information Reuse and Integration, pp. 157–162.
- Rahul Parundekar, Craig Knoblock, José Luis Ambite. 2012. *Discovering Concept Coverings in Ontologies of Linked Data Sources*. Proceedings of ISWC2012, pp. 427–443.
- Balthasar Schopman, Shenghui Wang, Antoine Isaac, Stefan Schlobach. 2012. *Instance-Based Ontology Matching by Instance Enrichment*. Journal on Data Semantics, 1(4), pp 219-236
- Julius Volz, Christian Bizer, Martin Gaedke, Georgi Kobilarov. 2009. *Silk – A Link Discovery Framework for the Web of Data*. Proceedings of the 2nd Workshop on Linked Data on the Web
- Lihua Zhao, Ryutaro Ichise. 2011. *Mid-ontology learning from linked data*. Proceedings of the Joint International Semantic Technology Conference, pp. 112 – 127.
- Lihua Zhao, Ryutaro Ichise. 2012. *Graph-based ontology analysis in the linked open data*. Proceedings of the 8th International Conference on Semantic Systems, pp. 56 – 63
- Jiwei Zhong, Haiping Zhu, Jianming Li and Yong Yu. 2002. *Conceptual Graph Matching for Semantic Search*. The 2002 International Conference on Computational Science.

# Context-Dependent Multilingual Lexical Lookup for Under-Resourced Languages

Lian Tze Lim<sup>\*†</sup>

<sup>\*</sup>SEST, KDU College Penang  
Georgetown, Penang, Malaysia  
liantze@gmail.com

Enya Kong Tang

Linton University College  
Seremban, Negeri Sembilan, Malaysia  
enyakong1@gmail.com

Lay-Ki Soon and Tek Yong Lim

<sup>†</sup>FCI, Multimedia University  
Cyberjaya, Selangor, Malaysia  
{lksoon, tylim}@mmu.edu.my

Bali Ranaivo-Malançon

FCSIT, Universiti Malaysia Sarawak,  
Kota Samarahan, Sarawak, Malaysia  
mbranaivo@fit.unimas.my

## Abstract

Current approaches for word sense disambiguation and translation selection typically require lexical resources or large bilingual corpora with rich information fields and annotations, which are often infeasible for under-resourced languages. We extract translation context knowledge from a bilingual comparable corpora of a richer-resourced language pair, and inject it into a multilingual lexicon. The multilingual lexicon can then be used to perform context-dependent lexical lookup on texts of any language, including under-resourced ones. Evaluations on a prototype lookup tool, trained on a English–Malay bilingual Wikipedia corpus, show a precision score of 0.65 (baseline 0.55) and mean reciprocal rank score of 0.81 (baseline 0.771). Based on the early encouraging results, the context-dependent lexical lookup tool may be developed further into an intelligent reading aid, to help users grasp the gist of a second or foreign language text.

## 1 Introduction

Word sense disambiguation (WSD) is the task of assigning sense tags to ambiguous lexical items (LIs) in a text. Translation selection chooses target language items for translating ambiguous LIs in a text, and can therefore be viewed as a kind of WSD task, with translations as the sense tags. The translation selection task may also be modified slightly to output a ranked list of translations. This then resembles a dictionary lookup process as performed by a human reader when reading or browsing a text written in a second or foreign language. For convenience's sake, we will call this task (as performed

via computational means) *context-dependent lexical lookup*. It can also be viewed as a simplified version of the Cross-Lingual Lexical Substitution (Mihalcea et al., 2010) and Cross-Lingual Word Sense Disambiguation (Lefever and Hoste, 2010) tasks, as defined in SemEval-2010.

There is a large body of work around WSD and translation selection. However, many of these approaches require lexical resources or large bilingual corpora with rich information fields and annotations, as reviewed in section 2. Unfortunately, not all languages have equal amounts of digital resources for developing language technologies, and such requirements are often infeasible for under-resourced languages.

We are interested in leveraging richer-resourced language pairs to enable context-dependent lexical lookup for under-resourced languages. For this purpose, we model translation context knowledge as a second-order co-occurrence bag-of-words model. We propose a rapid approach for acquiring them from an untagged, comparable bilingual corpus of a (richer-resourced) language pair in section 3. This information is then transferred into a multilingual lexicon to perform context-dependent lexical lookup on input texts, including those in an under-resourced language (section 4). Section 5 describes a prototype implementation, where translation context knowledge is extracted from a English–Malay bilingual corpus to enrich a multilingual lexicon with six languages. Results from a small experiment are presented in 6 and discussed in section 7. The approach is briefly compared with some related work in section 8, before concluding in section 9.

## 2 Typical Resource Requirements for Translation Selection

WSD and translation selection approaches may be broadly classified into two categories depending

on the type of learning resources used: knowledge- and corpus-based. Knowledge-based approaches make use of various types of information from existing dictionaries, thesauri, or other lexical resources. Possible knowledge sources include definition or gloss text (Banerjee and Pedersen, 2003), subject codes (Magnini et al., 2001), semantic networks (Shirai and Yagi, 2004; Mahapatra et al., 2010) and others.

Nevertheless, lexical resources of such rich content types are usually available for medium- to rich-resourced languages only, and are costly to build and verify by hand. Some approaches therefore turn to corpus-based approaches, use bilingual corpora as learning resources for translation selection. (Ide et al., 2002; Ng et al., 2003) used aligned corpora in their work. As it is not always possible to acquire parallel corpora, comparable corpora, or even independent second-language corpora have also been shown to be suitable for training purposes, either by purely numerical means (Li and Li, 2004) or with the aid of syntactic relations (Zhou et al., 2001). Vector-based models, which capture the context of a translation or meaning, have also been used (Schütze, 1998; Papp, 2009). For under-resourced languages, however, bilingual corpora of sufficient size may still be unavailable.

### 3 Enriching Multilingual Lexicon with Translation Context Knowledge

Corpus-driven translation selection approaches typically derive supporting semantic information from an aligned corpus, where a text and its translation are aligned at the sentence, phrase and word level. However, aligned corpora can be difficult to obtain for under-resourced language pairs, and are expensive to construct.

On the other hand, documents in a comparable corpus comprise bilingual or multilingual text of a similar nature, and need not even be exact translations of each other. The texts are therefore unaligned except at the document level. Comparable corpora are relatively easier to obtain, especially for richer-resourced languages.

#### 3.1 Overview of Multilingual Lexicon

Entries in our multilingual lexicon are organised as multilingual translation sets, each corresponding to a coarse-grained concept, and whose members are LIs from different languages  $\{L_1, \dots, L_N\}$  conveying the same concept. We denote an LI as

«item», sometimes with the 3-letter ISO language code in underscript when necessary: «item»<sub>eng</sub>. A list of 3-letter ISO language codes used in this paper is given in Appendix A.

For example, following are two translation sets containing different senses of English «bank» (‘financial institution’ and ‘riverside land’):

$$TS_1 = \{\text{«bank»}_{\text{eng}}, \text{«bank»}_{\text{msa}}, \text{«銀行»}_{\text{zho}}, \dots\}$$

$$TS_2 = \{\text{«bank»}_{\text{eng}}, \text{«tebing»}_{\text{msa}}, \text{«岸»}_{\text{zho}}, \dots\}.$$

Multilingual lexicons with under-resourced languages can be rapidly bootstrapped from simple bilingual translation lists (Lim et al., 2011). Our multilingual lexicon currently contains 24371 English, 13226 Chinese, 35640 Malay, 17063 French, 14687 Thai and 5629 Iban LIs.

#### 3.2 Extracting Translation Context Knowledge from Comparable Corpus

We model translation knowledge as a bag-of-words consisting of the context of a translation equivalence in the corpus. We then run latent semantic indexing (LSI) (Deerwester et al., 1990) on a comparable bilingual corpora. A vector is then obtained for each LI in both languages, which may be regarded as encoding some translation context knowledge.

While LSI is more frequently used in information retrieval, the translation knowledge acquisition task can be recast as a cross-lingual indexing task, following (Dumais et al., 1997). The underlying intuition is that in a comparable bilingual corpus, a document pair about finance would be more likely to contain English «bank»<sub>eng</sub> and Malay «bank»<sub>msa</sub> (‘financial institution’), as opposed to Malay «tebing»<sub>msa</sub> (‘riverside’). The words appearing in this document pair would then be an indicative context for the translation equivalence between «bank»<sub>eng</sub> and «bank»<sub>msa</sub>. In other words, the translation equivalents present serve as a kind of implicit sense tag.

Briefly, a translation knowledge vector is obtained for each multilingual translation set from a bilingual comparable corpus as follows:

1. Each bilingual pair of documents is merged as one single document, with each LI tagged with its respective language code.
2. Pre-process the corpus, e.g. remove closed-class words, perform stemming or lemmatisation, and word segmentation for languages without word boundaries (Chinese, Thai).

3. Construct a term-document matrix (TDM), using the frequency of terms (each made up by a LI and its language tag) in each document. Apply further weighting, e.g. TF-IDF, if necessary.
4. Perform LSI on the TDM. A vector is then obtained for every LI in both languages.
5. Set the vector associated with each translation set to be the sum of all available vectors of its member LIs.

#### 4 Context-Dependent Lexical Lookup

Given an input text in language  $L_i$  ( $1 \leq i \leq N$ ), the lookup module should return a list of multilingual translation set entries, which would contain  $L_1, L_2, \dots, L_N$  translation equivalents of LIs in the input text, wherever available. For polysemous LIs, the lookup module should return translation sets that convey the appropriate meaning in context.

For each input text segment  $Q$  (typically a sentence), a ‘query vector’,  $V_Q$  is computed by taking the vectorial sum of all open class LIs in the input  $Q$ . For each LI  $l$  in the input, the list of all translation sets containing  $l$ , is retrieved into  $TS_l$ .

$TS_l$  is then sorted in descending order of

$$\text{CSim}(V_t, V_Q) = \frac{V_t \cdot V_Q}{|V_t| \times |V_Q|}$$

(i.e. the cosine similarity between the query vector  $V_Q$  and the translation set candidate  $t$ ’s vector) for all  $t \in TS_l$ .

If the language of input  $Q$  is not present in the bilingual training corpus (e.g. Iban, an under-resourced language spoken in Borneo),  $V_Q$  is then computed as the sum of all vectors associated with all translation sets in  $TS_l$ . For example, given the Iban sentence ‘*Lelaki nya tikah enggau emperaja iya, siko dayang ke ligung*’ (‘he married his sweetheart, a pretty girl’),  $V_Q$  would be computed as

$$\begin{aligned} V_Q = & \sum V(\text{lookup}(\langle\langle\text{lelaki}\rangle\rangle_{\text{iba}})) \\ & + \sum V(\text{lookup}(\langle\langle\text{tikah}\rangle\rangle_{\text{iba}})) \\ & + \sum V(\text{lookup}(\langle\langle\text{emperaja}\rangle\rangle_{\text{iba}})) \\ & + \sum V(\text{lookup}(\langle\langle\text{dayang}\rangle\rangle_{\text{iba}})) \\ & + \sum V(\text{lookup}(\langle\langle\text{ligung}\rangle\rangle_{\text{iba}})) \end{aligned}$$

where the function  $\text{lookup}(w)$  returns the translation sets containing LI  $w$ .

#### 5 Prototype Implementation

We have implemented LEXICALSELECTOR, a prototype context-dependent lexical lookup tool in Java, trained on a English–Malay bilingual corpus built from Wikipedia articles. Wikipedia articles are freely available under a Creative Commons license, thus providing a convenient source of bilingual comparable corpus. Note that while the training corpus is English–Malay, the trained lookup tool can be applied to texts of any language included in the multilingual dictionary.

Malay Wikipedia articles<sup>1</sup> and their corresponding English articles of the same topics<sup>2</sup> were first downloaded. To form the bilingual corpus, each Malay article is concatenated with its corresponding English article as one document.

The TDM constructed from this corpus contains 62 993 documents and 67 499 terms, including both English and Malay items. The TDM is weighted by TF-IDF, then processed by LSI using the Gensim Python library<sup>3</sup>. The indexing process, using 1000 factors, took about 45 minutes on a MacBook Pro with a 2.3 GHz processor and 4 GB RAM. The vectors obtained for each English and Malay LIs were then used to populate the translation context knowledge vectors of translation set in a multilingual lexicon, which comprise six languages: English, Malay, Chinese, French, Thai and Iban.

As mentioned earlier, LEXICALSELECTOR can process texts in any member languages of the multilingual lexicon, instead of only the languages of the training corpus (English and Malay). Figure 1 shows the context-dependent lexical lookup outputs for the Iban input ‘*Lelaki nya tikah enggau emperaja iya, siko dayang ke ligung*’. Note that «emperaja» is polysemous (‘rainbow’ or ‘lover’), but is successfully identified as meaning ‘lover’ in this sentence.

#### 6 Early Experimental Results

80 input sentences containing LIs with translation ambiguities were randomly selected from the Internet (English, Malay and Chinese) and contributed by a native speaker (Iban). The test words are:

- English «plant» (vegetation or factory),

<sup>1</sup><http://dumps.wikimedia.org/mswiki/>

<sup>2</sup><http://en.wikipedia.org/wiki/Special:Export>

<sup>3</sup><http://radimrehurek.com/gensim/>



|                                       |  |                                     |
|---------------------------------------|--|-------------------------------------|
| = lelaki =                            | = emperaja =                           | = ligung =                          |
| zho: 男性,                              | zho: 情人,                               | zho: 可爱,                            |
| tha: ตัวผู้,                          | tha: คู่ควง, คู่รัก, ดวงสมร, ยอดรัก,   | tha: น่าเกลียดน่าชัง, น่ารักน่าชัง, |
| fra: mâle, masculin,                  | สุดที่รัก, หวานใจ, แฟน,                | fra: joli, mignon,                  |
| msa: lelaki, jantan,                  | msa: kekasih,                          | msa: comel,                         |
| eng: male,                            | eng: sweetheart,                       | eng: cute, pretty,                  |
| <br>                                  |  |                                     |
| = tikah =                             | = dayang =                             |                                     |
| zho: 结婚,                              | zho: 女孩子, 姑娘,                          |                                     |
| tha: สมรส, ออกเรือน, แต่งงาน, วิวาห์, | tha: กัญญา, ด.ญ., สาวน้อย, สาวรุ่น,    |                                     |
| fra: épouser, se marier,              | เด็กผู้หญิง, เด็กสาว, เด็กหญิง, ดรุณี, |                                     |
| msa: bernikah, menikahi, mengahwini,  | สาว,                                   |                                     |
| berkahwin,                            | msa: pemudi, puteri, perawan, dara,    |                                     |
| eng: marry, wed,                      | eng: girl,                             |                                     |

Figure 1: LEXICALSELECTOR output for Iban input ‘Lelaki nya tikah enggau emperaja iya, siko dayang ke ligung’. Only top ranked translation sets are shown.

- English «bank» (financial institution or river-side land),
- Malay «kabinet» (governmental Cabinet or household furniture),
- Malay «mangga» (mango or padlock),
- Chinese «谷» (*gù*, valley or grain) and
- Iban «emperaja» (rainbow or lover).

Each test sentence was first POS-tagged automatically based on the Penn Treebank tagset. The English test sentences were lemmatised and POS-tagged with the Stanford Parser.<sup>4</sup> The Chinese test sentences segmented with the Stanford Chinese Word Segmenter tool.<sup>5</sup> For Malay POS-tagging, we trained the QTag tagger<sup>6</sup> on a hand-tagged Malay corpus, and applied the trained tagger on our test sentences. As we lacked a Iban POS-tagger, the Iban test sentences were tagged by hand. LIs of each language and their associated vectors can then be retrieved from the multilingual lexicon.

The prototype tool LEXICALSELECTOR then computes the CSim score and ranks potential translation sets for each LI in the input sentences (ranking strategy *wiki-lsi*). The baseline strategy (*base-freq*) selects the translation set whose members occur most frequently in the bilingual Wikipedia corpus.

As a comparison, the English, Chinese and Malay test sentences were fed to Google Translate<sup>7</sup> and translated into Chinese, Malay and English. (Google Translate does not support Iban currently.) The Google Translate interface makes available the ranked list of translation candidates for each word in an input sentence, one language

at a time. The translated word for each of the input test word can therefore be noted. The highest rank of the correct translation for the test words in English/Chinese/Malay are used to evaluate *goog-tr*.

Two metrics were used in this quick evaluation. The first metric is by taking the precision of the first translation set returned by each ranking strategy, i.e. whether the top ranked translation set contains the correct translation of the ambiguous item. The precision metric is important for applications like machine translation, where only the top-ranked meaning or translation is considered.

The results may also be evaluated similar to a document retrieval task, i.e. as a ranked lexical lookup for human consumption. This is measured by the mean reciprocal rank (MRR), the average of the reciprocal ranks of the correct translation set for each input sentence in the test set  $T$ :

$$\text{MRR} = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{\text{rank}_i}$$

The results for the three ranking strategies are summarised in Table 1. For the precision metric, *wiki-lsi* scored 0.650 when all 80 input sentences are tested, while the *base-freq* baseline scored 0.550. *goog-tr* has the highest precision at 0.797. However, if only the Chinese and Malay inputs — which has less presence on the Internet and ‘less resource-rich’ than English — were tested (since *goog-tr* cannot accept Iban inputs), *wiki-lsi* and *goog-tr* actually performs equally well at 0.690 precision.

In our evaluation, the MRR score of *wiki-lsi* is 0.810, while *base-freq* scored 0.771. *wiki-lsi* even outperforms *goog-tr* when only the Chinese and Malay test sentences are considered for the MRR metric, as *goog-tr*

<sup>4</sup><http://www-nlp.stanford.edu/software/lex-parser.shtml>

<sup>5</sup><http://nlp.stanford.edu/software/segmenter.shtml>

<sup>6</sup><http://phrasys.net/uob/om/software>

<sup>7</sup><http://translate.google.com> on 3 October 2012

Table 1: Precision and MRR scores of context-dependent lexical lookup

| Strategy  | Incl. Eng. & Iban |       | W/o Eng. & Iban |       |
|-----------|-------------------|-------|-----------------|-------|
|           | Precision         | MRR   | Precision       | MRR   |
| wiki-lsi  | 0.650             | 0.810 | 0.690           | 0.845 |
| base-freq | 0.550             | 0.771 | 0.524           | 0.762 |
| goog-tr   | 0.797             | 0.812 | 0.690           | 0.708 |

did not present the correct translation in its list of alternative translation candidates for some test sentences. This suggests that the LSI-based translation context knowledge vectors would be helpful in building an intelligent reading aid.

## 7 Discussion

wiki-lsi performed better than base-freq for both the precision and the MRR metrics, although further tests is warranted, given the small size of the current test set. While wiki-lsi is not yet sufficiently accurate to be used directly in an MT system, it is helpful in producing a list of ranked multilingual translation sets depending on the input context, as part of an intelligent reading aid. Specifically, the lookup module would have benefited if syntactic information (e.g. syntactic relations and parse trees) was incorporated during the training and testing phase. This would require more time in parsing the training corpus, as well as assuming that syntactic analysis tools are available to process test sentences of all languages, including the under-resourced ones.

Note that even though the translation context knowledge vectors were extracted from an English–Malay corpus, the same vectors can be applied on Chinese and Iban input sentences as well. This is especially significant for Iban, which otherwise lacks resources from which a lookup or disambiguation tool can be trained. Translation context knowledge vectors mined via LSI from a bilingual comparable corpus, therefore offers a fast, low cost and efficient fallback strategy for acquiring multilingual translation equivalence context information.

## 8 Related Work

Basile and Semeraro (2010) also used Wikipedia articles as a parallel corpus for their participation in the SemEval 2010 Cross-Lingual Lexical Substitution task. Both training and test data were for English–Spanish. The idea behind their system

is to count, for each potential Spanish candidate, the number of documents in which the target English word and the Spanish candidate occurs in an English–Spanish document pair. In the task’s ‘best’ evaluation (which is comparable to our ‘Precision’ metric), Basile and Semeraro’s system scored 26.39 precision on the trial data and 19.68 precision on the SemEval test data. This strategy of selecting the most frequent translation is similar to our base-freq baseline strategy.

Sarrafzadeh et al. (2011) also tackled the problem of cross-lingual disambiguation for under-resourced language pairs (English–Persian) using Wikipedia articles, by applying the *one sense per collocation* and *one sense per discourse* heuristics on a comparable corpus. The authors incorporated English and Persian wordnets in their system, thus achieving 0.68 for the ‘best sense’ (‘Precision’) evaluation. However, developing wordnets for new languages is no trivial effort, as acknowledged by the authors.

## 9 Conclusion

We extracted translation context knowledge from a bilingual comparable corpus by running LSI on the corpus. A context-dependent multilingual lexical lookup module was implemented, using the cosine similarity score between the vector of the input sentence and those of candidate translation sets to rank the latter in order of relevance. The precision and MRR scores outperformed Google Translate’s lexical selection for medium- and under-resourced language test inputs. The LSI-backed translation context knowledge vectors, mined from bilingual comparable corpora, thus provide an fast and affordable data source for building intelligent reading aids, especially for under-resourced languages.

## Acknowledgments

The authors thank Multimedia University and Universiti Malaysia Sarawak for providing support and resources during the conduct of this study. We also thank Panceras Talita for helping to prepare the Iban test sentences for context-dependent lookup.

## A 3-Letter ISO Language Codes

| Code | Language | Code | Language |
|------|----------|------|----------|
| eng  | English  | msa  | Malay    |
| zho  | Chinese  | fra  | French   |
| tha  | Thai     | iba  | Iban     |

## References

- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810.
- Pierpaolo Basile and Giovanni Semeraro. 2010. UBA: Using automatic translation and Wikipedia for cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 242–247, Uppsala, Sweden.
- Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Susan T. Dumais, Michael L. Littman, and Thomas K. Landauer. 1997. Automatic cross-language retrieval using latent semantic indexing. In *AAAI97 Spring Symposium Series: Cross Language Text and Speech Retrieval*, pages 18–24, Stanford University.
- Nancy Ide, Tomaz Erjavec, and Dan Tufiş. 2002. Sense discrimination with parallel corpora. In *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 54–60, Philadelphia, USA.
- 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 2010)*, Uppsala, Sweden.
- Hang Li and Cong Li. 2004. Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, 30(1):1–22.
- Lian Tze Lim, Bali Ranaivo-Malançon, and Enya Kong Tang. 2011. Low cost construction of a multilingual lexicon from bilingual lists. *Polibits*, 43:45–51.
- Bernardo Magnini, Carlo Strapparava, Giovanni Pezulo, and Alfio Gliozzo. 2001. Using domain information for word sense disambiguation. In *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, pages 111–114, Toulouse, France.
- Lipta Mahapatra, Meera Mohan, Mitesh M. Khapra, and Pushpak Bhattacharyya. 2010. OWNS: Cross-lingual word sense disambiguation using weighted overlap counts and Wordnet based similarity measures. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, Uppsala, Sweden.
- Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, Uppsala, Sweden.
- Hwee Tou Ng, Bin Wang, and Yee Seng 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*, pages 455–462, Sapporo, Japan.
- Gyula Papp. 2009. Vector-based unsupervised word sense disambiguation for large number of contexts. In Václav Matoušek and Pavel Mautner, editors, *Text, Speech and Dialogue*, volume 5729 of *Lecture Notes in Computer Science*, pages 109–115. Springer Berlin Heidelberg.
- Bahareh Sarrafzadeh, Nikolay Yakovets, Nick Cercone, and Aijun An. 2011. Cross-lingual word sense disambiguation for languages with scarce resources. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, pages 347–358, St. John’s, Canada.
- Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.
- Kiyoaki Shirai and Tsunekazu Yagi. 2004. Learning a robust word sense disambiguation model using hypernyms in definition sentences. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 917–923, Geneva, Switzerland. Association for Computational Linguistics.
- Ming Zhou, Yuan Ding, and Changning Huang. 2001. Improving translation selection with a new translation model trained by independent monolingual corpora. *Computational Linguistics and Chinese language Processing*, 6(1):1–26.

# Sorani Kurdish versus Kurmanji Kurdish: An Empirical Comparison

**Kyumars Sheykh Esmaili**

Nanyang Technological University  
N4-B2a-02  
Singapore  
kyumarss@ntu.edu.sg

**Shahin Salavati**

University of Kurdistan  
Sanandaj  
Iran

shahin.salavati@ieee.org

## Abstract

Resource scarcity along with diversity—both in dialect and script—are the two primary challenges in Kurdish language processing. In this paper we aim at addressing these two problems by (i) building a text corpus for Sorani and Kurmanji, the two main dialects of Kurdish, and (ii) highlighting some of the orthographic, phonological, and morphological differences between these two dialects from statistical and rule-based perspectives.

## 1 Introduction

Despite having 20 to 30 millions of native speakers (Haig and Matras, 2002; Hassanpour et al., 2012; Thackston, 2006b; Thackston, 2006a), Kurdish is among the less-resourced languages for which the only linguistic resource available on the Web is raw text (Walther and Sagot, 2010).

Apart from the resource-scarcity problem, its diversity—in both dialect and writing systems—is another primary challenge in Kurdish language processing (Gautier, 1998; Gautier, 1996; Esmaili, 2012). In fact, Kurdish is considered a *bi-standard* language (Gautier, 1998; Hassanpour et al., 2012): the Sorani dialect written in an Arabic-based alphabet and the Kurmanji dialect written in a Latin-based alphabet. The features distinguishing these two dialects are phonological, lexical, and morphological.

In this paper we report on the first outcomes of a project<sup>1</sup> at *University of Kurdistan (UoK)* that aims at addressing these two challenges of the Kurdish language processing. More specifically, in this paper:

1. we report on the construction of the first relatively-large and publicly-available text corpus for the Kurdish language,

2. we present some insights into the orthographic, phonological, and morphological differences between Sorani Kurdish and Kurmanji Kurdish.

The rest of this paper is organized as follows. In Section 2, we first briefly introduce the Kurdish language and its two main dialects then underline their differences from a rule-based (a.k.a. corpus-independent) perspective. Next, after presenting the Pewan text corpus in Section 3, we use it to conduct a statistical comparison of the two dialects in Section 4. The paper is concluded in Section 5.

## 2 The Kurdish Language and Dialects

Kurdish belongs to the Indo-Iranian family of Indo-European languages. Its closest better-known relative is Persian. Kurdish is spoken in Kurdistan, a large geographical area spanning the intersections of Turkey, Iran, Iraq, and Syria. It is one of the two official languages of Iraq and has a regional status in Iran.

Kurdish is a dialect-rich language, sometimes referred to as a dialect continuum (Matras and Akin, 2012; Shahsavari, 2010). In this paper, however, we focus on Sorani and Kurmanji which are the two closely-related and widely-spoken dialects of the Kurdish language. Together, they account for more than 75% of native Kurdish speakers (Walther and Sagot, 2010).

As summarized below, these two dialects differ not only in some linguistics aspects, but also in their writing systems.

### 2.1 Morphological Differences

The important morphological differences are (MacKenzie, 1961; Haig and Matras, 2002; Samvelian, 2007):

1. Kurmanji is more conservative in retaining both gender (feminine:male) and case opposition (absolute:oblique) for nouns and

<sup>1</sup><http://eng.uok.ac.ir/esmaili/research/klpp/en/main.htm>

|              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| Arabic-based | ا | ب | ج | چ | د | ئ | ف | گ | ژ | ک  | ل  | م  | ن  | ۆ  | پ  | ق  | ر  | س  | ش  | ت  | وو | ف  | خ  | ز  |
| Latin-based  | A | B | C | Ç | D | Ê | F | G | J | K  | L  | M  | N  | O  | P  | Q  | R  | S  | Ş  | T  | Û  | V  | X  | Z  |

(a) One-to-One Mappings

|              |     |       |       |       |
|--------------|-----|-------|-------|-------|
|              | 25  | 26    | 27    | 28    |
| Arabic-based | / ئ | و     | ى     | ه     |
| Latin-based  | I   | U / W | Y / Î | E / H |

(b) One-to-Two Mappings

|              |      |    |     |     |     |
|--------------|------|----|-----|-----|-----|
|              | 29   | 30 | 31  | 32  | 33  |
| Arabic-based | ړ    | آ  | ع   | غ   | ح   |
| Latin-based  | (RR) | -  | (E) | (X) | (H) |

(c) One-to-Zero Mappings

Figure 1: The Two Standard Kurdish Alphabets

pronouns<sup>2</sup>. Sorani has largely abandoned this system and uses the pronominal suffixes to take over the functions of the cases,

2. in the past-tense transitive verbs, Kurmanji has the full ergative alignment<sup>3</sup> but Sorani, having lost the oblique pronouns, resorts to pronominal enclitics,
3. in Sorani, passive and causative are created via verb morphology, in Kurmanji they can also be formed with the helper verbs *hatin* (“to come”) and *dan* (“to give”) respectively, and
4. the definite marker *-aka* appears only in Sorani.

## 2.2 Scriptural Differences

Due to geopolitical reasons (Matras and Reershemius, 1991), each of the two dialects has been using its own writing system: while Sorani uses an Arabic-based alphabet, Kurmanji is written in a Latin-based one.

Figure 1 shows the two standard alphabets and the mappings between them which we have categorized into three classes:

- one-to-one mappings (Figure 1a), which cover a large subset of the characters,
- one-to-two mappings (Figure 1b); they reflect the inherent ambiguities between the two writing systems (Barkhoda et al., 2009). While transliterating between these two alphabets, the contextual information can provide hints in choosing the right counterpart.

<sup>2</sup>Although there is evidence of gender distinctions weakening in some varieties of Kurmanji (Haig and Matras, 2002).

<sup>3</sup>Recent research suggests that ergativity in Kurmanji is weakening due to either internally-induced change or contact with Turkish (Dixon, 1994; Dorleijn, 1996; Mahalingappa, 2010), perhaps moving towards a full nominative-accusative system.

- one-to-zero mappings (Figure 1c); they can be further split into two distinct subcategories: (i) the strong L and strong R characters ( $\{ل\}$  and  $\{ړ\}$ ) are used only in Sorani Kurdish<sup>4</sup> and demonstrate some of the inherent phonological differences between Sorani and Kurmanji, and (ii) the remaining three characters are primarily used in the Arabic loanwords in Sorani (in Kurmanji they are approximated with other characters).

It should be noted that both of these writing systems are phonetic (Gautier, 1998); that is, vowels are explicitly represented and their use is mandatory.

## 3 The Pewan Corpus

Text corpora are essential to Computational Linguistics and Natural Language Processing. In spite of the few attempts to build corpus (Gautier, 1998) and lexicon (Walther and Sagot, 2010), Kurdish still does not have any large-scale and reliable general or domain-specific corpus.

At *UoK*, we followed TREC (TREC, 2013)’s common practice and used news articles to build a text corpus for the Kurdish language. After surveying a range of options we chose two online news agencies: (i) *Peyamner* (Peyamner, 2013), a popular multi-lingual news agency based in Iraqi Kurdistan, and (ii) the Sorani (VOA, 2013b) and the Kurmanji (VOA, 2013a) websites of *Voice Of America*. Our main selection criteria were: (i) number of articles, (ii) subject diversity, and (iii) crawl-friendliness.

For each agency, we developed a crawler to fetch the articles and extract their textual content. In case of *Peyamner*, since articles have no language label, we additionally implemented a simple classifier that decides each page’s language

<sup>4</sup>Although there are a handful of words with the latter in Kurmanji too.

| Property                |               | Sorani Corpus | Kurmanji Corpus |
|-------------------------|---------------|---------------|-----------------|
| No. of Articles         | from VOA      | 18,420        | 5,699           |
|                         | from Peyamner | 96,920        | 19,873          |
|                         | total         | 115,340       | 25,572          |
| No. of distinct words   |               | 501,054       | 127,272         |
| Total no. of words      |               | 18,110,723    | 4,120,027       |
| Total no. of characters |               | 101,564,650   | 20,138,939      |
| Average word length     |               | 5.6           | 4.8             |

Table 1: The Pewan Corpus’s Basic Statistics

based on the occurrence of language-specific characters.

Overall, 115,340 Sorani articles and 25,572 Kurmanji articles were collected<sup>5</sup>. The articles are dated between 2003 and 2012 and their sizes range from 1KB to 154KB (on average 2.6KB). Table 1 summarizes the important properties of our corpus which we named *Pewan* –a Kurdish word meaning “measurement.”

Using *Pewan* and similar to the approach employed in (Savoy, 1999), we also built a list of Kurdish stopwords. To this end, we manually examined the top 300 frequent words of each dialect and removed the corpus-specific biases (e.g., “Iraq”, “Kurdistan”, “Regional”, “Government”, “Reported” and etc). The final Sorani and Kurmanji lists contain 157 and 152 words respectively, and as in other languages, they mainly consist of prepositions.

*Pewan*, as well as the stopword lists can be obtained from (Pewan, 2013). We hope that making these resources publicly available, will bolster further research on Kurdish language.

## 4 Empirical Study

In the first part of this section, we first look at the character and word frequencies and try to obtain some insights about the phonological and lexical correlations and discrepancies between Sorani and Kurmanji.

In the second part, we investigate two well-known linguistic laws –Heaps’ and Zipf’s. Although these laws have been observed in many of the Indo-European languages (Lü et al., 2013), their coefficients depend on language (Gelbukh and Sidorov, 2001) and therefore they can be

<sup>5</sup>The relatively small size of the Kurmanji collection is part of a more general trend. In fact, despite having a larger number of speakers, Kurmanji has far fewer online sources with raw text readily available and even those sources do not strictly follow its writing standards. This is partly a result of decades of severe restrictions on use of Kurdish language in Turkey, where the majority of Kurmanji speakers live (Hasanpour et al., 2012).

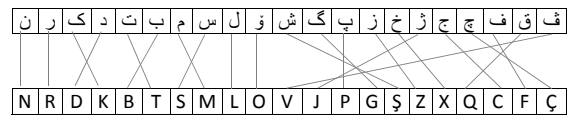


Figure 2: Relative Frequencies of Sorani and Kurmanji Characters in the Pewan Corpus

| #  | English Trans. | Freq.  | Sorani Word | Kurmanji Word | Freq.  | English Trans. | #  |
|----|----------------|--------|-------------|---------------|--------|----------------|----|
| 1  | from           | 859694 | له          | û             | 166401 | and            | 1  |
| 2  | and            | 653876 | و           | ku            | 112453 | which          | 2  |
| 3  | with           | 358609 | یه          | li            | 107259 | from           | 3  |
| 4  | for            | 270053 | بو          | de            | 82727  | -              | 4  |
| 5  | which          | 241046 | که          | bi            | 79422  | with           | 5  |
| 6  | that           | 170096 | ئو          | di            | 77690  | at             | 6  |
| 7  | this           | 83445  | ئهم         | ji            | 75064  | from           | 7  |
| 8  | of             | 74917  | ی           | ji            | 57655  | too            | 8  |
| 9  | together       | 58963  | لگه‌ل       | xwe           | 35579  | oneself        | 9  |
| 10 | made/did       | 55138  | کرد         | ya            | 31972  | of             | 10 |

Figure 3: The Top 10 Most-Frequent Sorani and Kurmanji Words in *Pewan*

used a tool to measure similarity/dissimilarity of languages. It should also be noted that in practice, knowing the coefficients of these laws is important in, for example, full-text database design, since it allows predicting some properties of the index as a function of the size of the database.

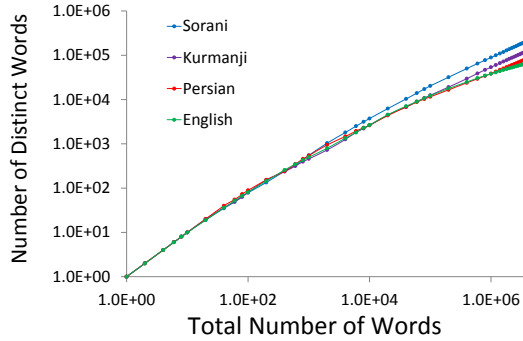
### 4.1 Character Frequencies

In this experiment we measure the character frequencies, as a phonological property of the language. Figure 2 shows the frequency-ranked lists (from left to right, in decreasing order) of characters of both dialects in the *Pewan* corpus. Note that for a fairer comparison, we have excluded characters with 1-to-0 and 1-to-2 mappings as well as three characters from the list of 1-to-1 mappings:  $\hat{A}$ ,  $\hat{E}$ , and  $\hat{U}$ . The first two have a skewed frequency due to their role as *Izafe* construction<sup>6</sup> marker. The third one is mapped to a double-character (وو) in the Sorani alphabet.

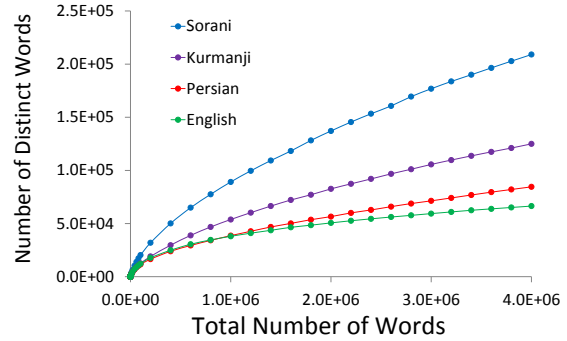
Overall, the relative positions of the equivalent characters in these two lists are comparable (Figure 2). However, there are two notable discrepancies which further exhibit the intrinsic phonological differences between Sorani and Kurmanji:

- use of the character  $\mathcal{J}$  is far more common in Kurmanji (e.g., in prepositions such as *ji* “from” and *ji* “too”),
- same holds for the character  $\mathcal{V}$ ; this is, how-

<sup>6</sup>*Izafe* construction is a shared feature of several Western Iranian languages (Samvelian, 2006). It, approximately, corresponds to the English preposition “of” and is added between prepositions, nouns and adjectives in a phrase (Shamsfard, 2011).



(a) Standard Representation



(b) Non-logarithmic Representation

Figure 4: Heaps' Law for Sorani and Kurmanji Kurdish, Persian, and English.

ever, due to Sorani's phonological tendency to use the phoneme  $\mathbb{W}$  instead of  $\mathbb{V}$ .

## 4.2 Word Frequencies

Figure 3 shows the most frequent Sorani and Kurmanji words in the Pewan corpus. This figure also contains the links between the words that are transliteration-equivalent and again shows a high level of correlation between the two dialects. A thorough examination of the longer version of the frequent terms' lists, not only further confirms this correlation but also reveals some other notable patterns:

- the Sorani generic preposition  $\text{ﻻ}$  (“from”) has a very wide range of use; in fact, as shown in Figure 3, it is the semantic equivalent of three common Kurmanji prepositions ( $\text{ﻻ}$ ,  $\text{ﺟﯩ}$ , and  $\text{ﺪﯨ}$ ),
- in Sorani, a number of the common prepositions (e.g.,  $\text{ﺑﯩﺶ}$  “too”) as well as the verb  $\text{ﺑﻮﻭﻥ}$  “to be” are used as suffix,
- in Kurmanji, some of the most common prepositions are paired with a postposition (mostly  $\text{ﺪﺍ}$ ,  $\text{ﺪﻩ}$ , and  $\text{ﻭﻩ}$ ) and form circumpositions,
- the Kurmanji's passive/accusative helper verbs ( $\text{ﻫﺎﺗﯩﻦ}$  and  $\text{ﺪﺍﻥ}$ ) are among its most frequently used words.

## 4.3 Heaps' Law

Heaps's law (Heaps, 1978) is about the growth of distinct words (a.k.a vocabulary size). More specifically, the number of distinct words in a text is roughly proportional to an exponent of its size:

$$\log n_i \approx D + h \log i \quad (1)$$

| Language        | $\log n_i$           | $h$  |
|-----------------|----------------------|------|
| <b>Sorani</b>   | $1.91 + 0.78 \log i$ | 0.78 |
| <b>Kurmanji</b> | $2.15 + 0.74 \log i$ | 0.74 |
| <b>Persian</b>  | $2.66 + 0.70 \log i$ | 0.70 |
| <b>English</b>  | $2.68 + 0.69 \log i$ | 0.69 |

Table 2: Heaps' Linear Regression

where  $n_i$  is the number of distinct words occurring before the running word number  $i$ ,  $h$  is the exponent coefficient (between 0 and 1), and  $D$  is a constant. In a logarithmic scale, it is a straight line with about  $45^\circ$  angle (Gelbukh and Sidorov, 2001).

We carried out an experiment to measure the growth rate of distinct words for both of the Kurdish dialects as well as the Persian and English languages. In this experiment, the Persian corpus was drawn from the standard Hamshahri Collection (AleAhmad et al., 2009) and The English corpus consisted of the Editorial articles of The Guardian newspaper<sup>7</sup> (Guardian, 2013).

As the curves in Figure 4 and the linear regression coefficients in Table 2 show, the growth rate of distinct words in both Sorani and Kurmanji Kurdish are higher than Persian and English. This result demonstrates the morphological complexity of the Kurdish language (Samvelian, 2007; Walther, 2011). One of the driving factors behind this complexity, is the wide use of suffixes, most notably as: (i) the Izafe construction marker, (ii) the plural noun marker, and (iii) the indefinite marker.

Another important observation from this experiment is that Sorani has a higher growth rate compared to Kurmanji ( $h = 0.78$  vs.  $h = 0.74$ ).

<sup>7</sup>Since they are written by native speakers, cover a wide spectrum of topics between 2006 and 2013, and have clean HTML sources.

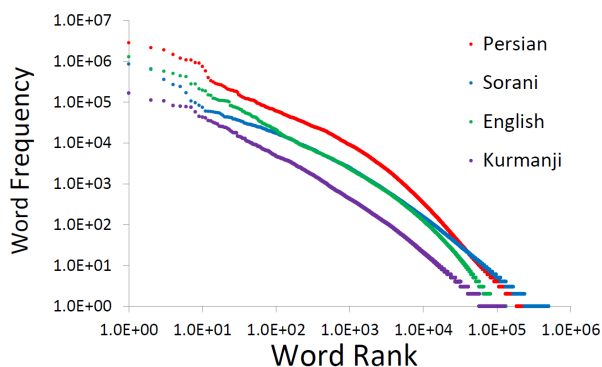


Figure 5: Zipf’s Laws for Sorani and Kurmanji Kurdish, Persian, and English.

| Language        | $\log f_r$           | $z$  |
|-----------------|----------------------|------|
| <b>Sorani</b>   | $7.69 - 1.33 \log r$ | 1.33 |
| <b>Kurmanji</b> | $6.48 - 1.31 \log r$ | 1.31 |
| <b>Persian</b>  | $9.57 - 1.51 \log r$ | 1.51 |
| <b>English</b>  | $9.37 - 1.85 \log r$ | 1.85 |

Table 3: Zipf’s Linear Regression

Two primary sources of these differences are: (i) the inherent linguistic differences between the two dialects as mentioned earlier (especially, Sorani’s exclusive use of definite marker), (ii) the general tendency in Sorani to use prepositions and helper verbs as suffix.

#### 4.4 Zipf’s Law

The Zipf’s law (Zipf, 1949) states that in any large-enough text, the frequency ranks of the words are inversely proportional to the corresponding frequencies:

$$\log f_r \approx C - z \log r \quad (2)$$

where  $f_r$  is the frequency of the word having the rank  $r$ ,  $z$  is the exponent coefficient, and  $C$  is a constant. In a logarithmic scale, it is a straight line with about  $45^\circ$  angle (Gelbukh and Sidorov, 2001).

The results of our experiment–plotted curves in Figure 5 and linear regression coefficients in Table 3– show that: (i) the distribution of the top most frequent words in Sorani is uniquely different; it first shows a sharper drop in the top 10 words and then a slower drop for the words ranked between 10 and 100, and (ii) in the remaining parts of the curves, both Kurmanji and Sorani behave similarly; this is also reflected in their values of coefficient  $z$  (1.33 and 1.31).

## 5 Conclusions and Future Work

In this paper we took the first steps towards addressing the two main challenges in Kurdish language processing, namely, resource scarcity and diversity. We presented Pewan, a text corpus for Sorani and Kurmanji, the two principal dialects of the Kurdish language. We also highlighted a range of differences between these two dialects and their writing systems.

The main findings of our analysis can be summarized as follows: (i) there are phonological differences between Sorani and Kurmanji; while some phonemes are non-existent in Kurmanji, some others are less-common in Sorani, (ii) they differ considerably in their vocabulary growth rates, (iii) Sorani has a peculiar frequency distribution w.r.t. its highly-common words. Some of the discrepancies are due to the existence of a generic preposition (⚡) in Sorani, as well as the general tendency in its writing system and style to use prepositions as suffix.

Our project at *UoK* is a work in progress. Recently, we have used the Pewan corpus to build a test collection to evaluate Kurdish Information Retrieval systems (Esmaili et al., 2013). In future, we plan to first develop stemming algorithms for both Sorani and Kurmanji and then leverage those algorithms to examine the lexical differences between the two dialects. Another avenue for future work is to build a transliteration/translation engine between Sorani and Kurmanji.

## Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments that helped us improve the quality of the paper.

## References

- Abolfazl AleAhmad, Hadi Amiri, Ehsan Darrudi, Masoud Rahgozar, and Farhad Oroumchian. 2009. Hamshahri: A standard Persian Text Collection. *Knowledge-Based Systems*, 22(5):382–387.
- Wafa Barkhoda, Bahram ZahirAzami, Anvar Bahrampour, and Om-Kolsoom Shahryari. 2009. A Comparison between Allophone, Syllable, and Di-phone based TTS Systems for Kurdish Language. In *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, pages 557–562.
- Robert MW Dixon. 1994. *Ergativity*. Cambridge University Press.



- Margreet Dorleijn. 1996. The Decay of Ergativity in Kurdish.
- Kyumars Sheykh Esmaili, Shahin Salavati, Somayeh Yosefi, Donya Eliassi, Purya Aliabadi, Shownm Hakimi, and Asrin Mohammadi. 2013. Building a Test Collection for Sorani Kurdish. In *(to appear) Proceedings of the 10th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '13)*.
- Kyumars Sheykh Esmaili. 2012. Challenges in Kurdish Text Processing. *CoRR*, abs/1212.0074.
- Gérard Gautier. 1996. A Lexicographic Environment for Kurdish Language using 4th Dimension. In *Proceedings of ICEMCO*.
- Gérard Gautier. 1998. Building a Kurdish Language Corpus: An Overview of the Technical Problems. In *Proceedings of ICEMCO*.
- Alexander Gelbukh and Grigori Sidorov. 2001. Zipf and Heaps Laws' Coefficients Depend on Language. In *Computational Linguistics and Intelligent Text Processing*, pages 332–335. Springer.
- Guardian. 2013. The Guardian. [www.guardian.co.uk/](http://www.guardian.co.uk/).
- Goeffrey Haig and Yaron Matras. 2002. Kurdish Linguistics: A Brief Overview. *Sprachtypologie und Universalienforschung / Language Typology and Universals*, 55(1).
- Amir Hassanpour, Jaffer Sheyholislami, and Tove Skutnabb-Kangas. 2012. Introduction. Kurdish: Linguicide, Resistance and Hope. *International Journal of the Sociology of Language*, 2012(217):118.
- Harold Stanley Heaps. 1978. *Information Retrieval: Computational and Theoretical Aspects*. Academic Press, Inc. Orlando, FL, USA.
- Linyuan Lü, Zi-Ke Zhang, and Tao Zhou. 2013. Deviation of Zipf's and Heaps' Laws in Human Languages with Limited Dictionary Sizes. *Scientific reports*, 3.
- David N. MacKenzie. 1961. *Kurdish Dialect Studies*. Oxford University Press.
- Laura Mahalingappa. 2010. The Acquisition of Split-Ergativity in Kurmanji Kurdish. In *The Proceedings of the Workshop on the Acquisition of Ergativity*.
- Yaron Matras and Salih Akin. 2012. A Survey of the Kurdish Dialect Continuum. In *Proceedings of the 2nd International Conference on Kurdish Studies*.
- Yaron Matras and Gertrud Reershemius. 1991. Standardization Beyond the State: the Cases of Yidish, Kurdish and Romani. *Von Gleich and Wolff*, 1991:103–123.
- Pewan. 2013. Pewan's Download Link. <https://dl.dropbox.com/u/10883132/Pewan.zip>.
- Peyamner. 2013. Peyamner News Agency. <http://www.peyamner.com/>.
- Pollet Samvelian. 2006. When Morphology Does Better Than Syntax: The Ezafe Construction in Persian. *Ms., Université de Paris*.
- Pollet Samvelian. 2007. A Lexical Account of Sorani Kurdish Prepositions. In *The Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*, pages 235–249, Stanford. CSLI Publications.
- Jacques Savoy. 1999. A Stemming Procedure and Stopword List for General French Corpora. *JASIS*, 50(10):944–952.
- Faramarz Shahsavari. 2010. Laki and Kurdish. *Iran and the Caucasus*, 14(1):79–82.
- Mehrnoosh Shamsfard. 2011. Challenges and Open Problems in Persian Text Processing. In *Proceedings of LTC'11*.
- Wheeler M. Thackston. 2006a. *Kurmanji Kurdish: A Reference Grammar with Selected Readings*. Harvard University.
- Wheeler M. Thackston. 2006b. *Sorani Kurdish: A Reference Grammar with Selected Readings*. Harvard University.
- TREC. 2013. Text REtrieval Conference. <http://trec.nist.gov/>.
- VOA. 2013a. Voice of America - Kurdish (Kurmanji). <http://www.dengeamerika.com/>.
- VOA. 2013b. Voice of America - Kurdish (Sorani). <http://www.dengiamerika.com/>.
- Géraldine Walther and Benoît Sagot. 2010. Developing a Large-scale Lexicon for a Less-Resourced Language. In *SaLTMiL's Workshop on Less-resourced Languages (LREC)*.
- Géraldine Walther. 2011. Fitting into Morphological Structure: Accounting for Sorani Kurdish Endoclitics. In Stefan Müller, editor, *The Proceedings of the Eighth Mediterranean Morphology Meeting (MMM8)*, pages 299–322, Cagliari, Italy.
- George Kingsley Zipf. 1949. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley.

# Enhanced and Portable Dependency Projection Algorithms Using Interlinear Glossed Text

**Ryan Georgi**

University of Washington  
Seattle, WA 98195, USA  
rgeorgi@uw.edu

**Fei Xia**

University of Washington  
Seattle, WA 98195, USA  
fxia@uw.edu

**William D. Lewis**

Microsoft Research  
Redmond, WA 98052, USA  
wilewis@microsoft.com

## Abstract

As most of the world’s languages are under-resourced, projection algorithms offer an enticing way to bootstrap the resources available for one resource-poor language from a resource-rich language by means of parallel text and word alignment. These algorithms, however, make the strong assumption that the language pairs share common structures and that the parse trees will resemble one another. This assumption is useful but often leads to errors in projection. In this paper, we will address this weakness by using trees created from instances of Interlinear Glossed Text (IGT) to discover patterns of divergence between the languages. We will show that this method improves the performance of projection algorithms significantly in some languages by accounting for divergence between languages using only the partial supervision of a few corrected trees.

## 1 Introduction

While thousands of languages are spoken in the world, most of them are considered *resource-poor* in the sense that they do not have a large number of electronic resources that can be used to build NLP systems. For instance, some languages may lack treebanks, thus making it difficult to build a high-quality statistical parser.

One common approach to address this problem is to take advantage of bitext between a resource-rich language (e.g., English) and a resource-poor language by projecting information from the former to the latter (Yarowsky and Ngai, 2001; Hwa et al., 2004). While pro-

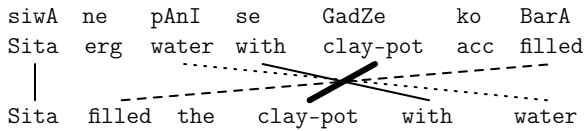
jection methods can provide a great deal of information at minimal cost to the researchers, they do suffer from structural divergence between the language-poor language (aka target language) and the resource-rich language (aka source language).

In this paper, we propose a middle ground between manually creating a large-scale treebank (which is expensive and time-consuming) and relying on the syntactic structures produced by a projection algorithm alone (which are error-prone).

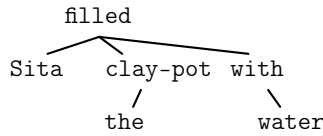
Our approach has several steps. First, we utilize instances of Interlinear Glossed Text (IGT) following Xia and Lewis (2007) as seen in Figure 1(a) to create a small set of parallel dependency trees through projection and then manually correct the dependency trees. Second, we automatically analyze this small set of parallel trees to find patterns where the corrected data differs from the projection. Third, those patterns are incorporated to the projection algorithm to improve the quality of projection. Finally, the features extracted from the projected trees are added to a statistical parser to improve parsing quality. The outcome of this work are both an enhanced projection algorithm and a better parser for resource-poor languages that require a minimal amount of manual effort.

## 2 Previous Work

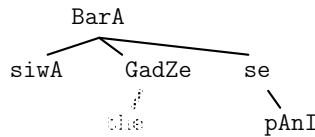
For this paper, we will be building upon the standard projection algorithm for dependency structures as outlined in Quirk et al. (2005) and illustrated in Figure 1. First, a sentence pair between resource-rich (source) and resource-poor (target) languages is word aligned [Fig 1(a)]. Second, the source sentence is parsed by a dependency parser for the source language [Fig 1(b)]. Third, sponta-



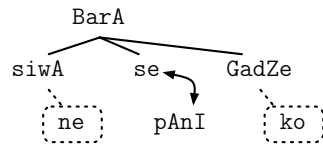
(a) An Interlinear Glossed Text (IGT) instance in Hindi and word alignment between the gloss line and the English translation.



(b) Dependency parse of English translation.



(c) English words are replaced with Hindi words and spontaneous word “the” are removed from the tree.



(d) Siblings in the tree are reordered based on the word order of the Hindi sentence and spontaneous Hindi words are attached as indicated by dotted lines. The words *pAnI* and *se* are incorrectly inverted, as indicated by the curved arrow.

Figure 1: An example of projecting a dependency tree from English to Hindi.

neous (unaligned) source words are removed, and the remaining words are replaced with corresponding words in the target side [Fig 1(c)]. Finally, spontaneous target words are re-attached heuristically and the children of a head are ordered based on the word order in the target sentence [Fig 1(d)]. The resulting tree may have errors (e.g., *pAni* should depend on *se* in Figure 1(d)), and the goal of this study is to reduce common types of projection errors.

In Georgi et al. (2012a), we proposed a method for analyzing parallel dependency corpora in which word alignment between trees was used to determine three types of edge configurations: **merged**, **swapped**, and **spontaneous**. Merged alignments were those in which multiple words in the target tree aligned to a single word in the source tree, as in Figure 2. Swapped alignments were those in which a parent node in the source tree aligned to a

child in the target tree and vice-versa. Finally, spontaneous alignments were those for which a word did not align to any word on the other side. These edge configurations could be detected from simple parent–child edges and the alignment (or lack of) between words in the language pairs. Using these simple, language-agnostic measures allows one to look for divergence types such as those described by Dorr (1994).

Georgi et al. (2012b) described a method in which new features were extracted from the projected trees and added to the feature vectors for a statistical dependency parser. The rationale was that, although the projected trees were error-prone, the parsing model should be able to set appropriate weights of these features based on how reliable these features were in indicating the dependency structure. We started with the MSTParser (McDonald et al., 2005) and modified it so that the edges from the projected trees could be used as features at parse time. Experiments showed that adding new features improved parsing performance.

In this paper, we use the small training corpus built in Georgi et al. (2012b) to improve the projection algorithm itself. The improved projected trees are in turn fed to the statistical parser to further improve parsing results.

### 3 Enhancements to the projection algorithm

We propose to enhance the projection algorithm by addressing the three alignment types discussed earlier:

1. Merge: better informed choice for head for multiply-aligned words.
2. Swap: post-projection correction of frequently swapped word pairs.
3. Spontaneous: better informed attachment of target spontaneous words.

The detail of the enhancements are explained below.

#### 3.1 Merge Correction

“Merged” words, or multiple words on the target side that align to a single source word, are problematic for the projection algorithm because it is not clear which target word should be the head and which word should be the

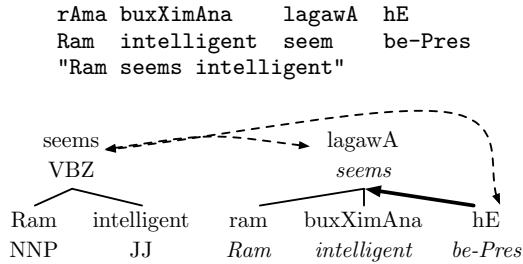


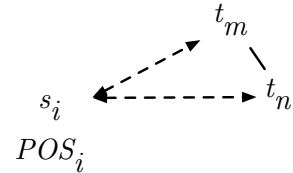
Figure 2: An example of merged alignment, where the English word *seems* align to two Hindi words *hE* and *lagawA*. Below the IGT are the dependency trees for English and Hindi. Dotted arrows indicate word alignment, and the solid arrow indicates that *hE* should depend on *lagawA*.

dependent. An example is given in Figure 2, where the English word *seems* align to two Hindi words *hE* and *lagawA*.

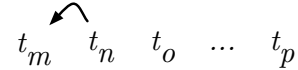
On the other hand, from the small amount of labeled training data (i.e., a set of hand-corrected tree pairs), we can learn what kind of source words are likely to align to multiple target words, and which target word is likely to the head. The process is illustrated in Figure 3. In this example, the target words  $t_m$  and  $t_n$  are both aligned with the source word  $s_i$  whose POS tag is  $POS_i$ , and  $t_m$  appears before  $t_n$  in the target sentence. Going through the examples of merged alignments in the training data, we keep a count for the POS tag of the source word and the position of the head on the target side.<sup>1</sup> Based on these counts, our system will generate rules such as the ones in Figure 3(c) which says if a source word whose POS is  $POS_i$  aligns to two target words, the probability of the right target word depending on the left one is 75%, and the probability of the left target word depending on the right one is 25%. We use maximum likelihood estimate (MLE) to calculate the probability.

The projection algorithm will use those rules to handle merged alignment; that is, when a source word aligns to multiple target words, the algorithm determines the direction of dependency edge based on the direction preference stored in the rules. In addition to rules for

<sup>1</sup>We use the position of the head, not the POS tag of the head, because the POS tags of the target words are not available when running the projection algorithm on the test data.



(a) Alignment between a source word and two target words, and one target word  $t_m$  is the parent of the other word  $t_n$ .



(b) Target sentence showing the “left” dependency between  $t_m$  and  $t_n$ .

$POS_i \rightarrow \text{left} \quad 0.75$   
 $POS_i \rightarrow \text{right} \quad 0.25$

(c) Rules for handling merged alignment

Figure 3: Example of merged alignment and rules derived from such an example

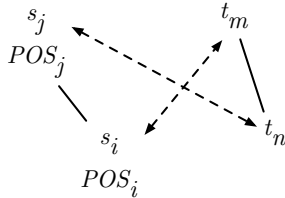
an individual source POS tag, our method also keeps track of the overall direction preference for all the merged examples in that language. For merges in which the source POS tag is unseen or there are no rules for that tag, this language-wide preference is used as a backoff.

### 3.2 Swap Correction

An example of swapped alignment is in Figure 4(a), where  $(s_j, s_i)$  is an edge in the source tree,  $(t_m, t_n)$  is an edge in the target tree, and  $s_j$  aligns to  $t_n$  and  $s_i$  aligns to  $t_m$ . Figure 1(d) shows an error made by the projection algorithm due to swapped alignment. In order to correct such errors, we count the number of  $(POS_{child}, POS_{parent})$  dependency edges in the source trees, and the number of times that the directions of the edges are reversed on the target side. Figure 4(b) shows a possible set of counts resulting from this approach. Based on the counts, we keep only the POS pairs that appear in at least 10% of training sentences and the percentage of swap for the pairs are no less than 70%.<sup>2</sup> We say that those pairs trigger a swap operation.

At the test time, swap rules are applied as a post-processing step to the projected tree. After the projected tree is completed, our swap handling step checks each edge in the source tree. If the POS tag pair for the edge triggers

<sup>2</sup>These thresholds are set empirically.



(a) A swapped alignment between source words  $s_j$  and  $s_i$  and target words  $t_m$  and  $t_n$ .

| POS Pair         |               | Swaps | Total | %   |
|------------------|---------------|-------|-------|-----|
| $(POS_i, POS_j)$ | $\rightarrow$ | 16    | 21    | 76  |
| $(POS_k, POS_l)$ | $\rightarrow$ | 1     | 1     | 100 |
| $(POS_n, POS_o)$ | $\rightarrow$ | 1     | 10    | 10  |

(b) Example set of learned swap rules. **Swaps** counts the number of times the given (child, parent) pair is seen in a swap configuration in the source side, and **total** is the number of times said pair occurs overall.

Figure 4: Example swap configuration and collected statistics.

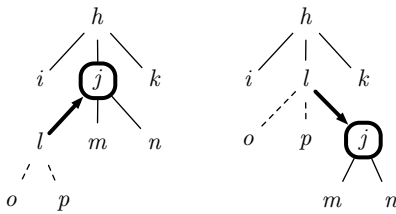


Figure 5: Swap operation: on the left is the original tree; on the right is the tree after swapping node  $l$  with its parent  $j$ .

a swap operation, the corresponding nodes in the projected tree will be swapped, as illustrated in Figure 5.

### 3.3 Spontaneous Reattachment

Target spontaneous words are difficult to handle because they do not align to any source word and thus there is nothing to project to them. To address this problem, we collect two types of information from the training data. First, we keep track of all the lexical items that appear in the training trees, and the relative position of their head. This lexical approach may be useful in handling closed-class words which account for a large percentage of spontaneous words. Second, we use the training trees to determine the favored attachment direction for the language as a whole.

At the test time, for each spontaneous word in the target sentence, if it is one of the words for which we have gathered statistics from the training data, we attach it to the next word in the preferred direction for that word. If the

word is unseen, we attach it using the overall language preference as a backoff.

### 3.4 Parser Enhancements

In addition to above enhancements to the projection algorithm itself, we train a dependency parser on the training data, with new features from the projected trees following Georgi et al. (2012b). Furthermore, we add features that indicate whether the current word appears in a merge or swap configuration. The results of this combination of additional features and improved projection is shown in Table 1(b).

## 4 Results

For evaluation, we use the same data sets as in Georgi et al. (2012b), where there is a small number (ranging from 46 to 147) of tree pairs for each of the eight languages. The IGT instances for those tree pairs come from the Hindi Treebank (Bhatt et al., 2009) and the Online Database of Interlinear Text (ODIN) (Lewis and Xia, 2010).

We ran 10-fold cross validation and reported the average of 10 runs in Table 1. The top table shows the accuracy of the projection algorithm, and the bottom table shows parsing accuracy of MSTParser with or without adding features from the projected trees. In both tables, the *Best* row uses the enhanced projection algorithm. The *Baseline* rows use the original projection algorithm in Quirk et al. (2005) where the word in the parentheses indicates the direction of merge. The *Error Reduction* row shows the error reduction of the *Best* system over the best performing baseline for each language. The *No Projection* row in the second table shows parsing results when no features from the projected trees are added to the parser, and the last row in that table shows the error reduction of the *Best* row over the *No Projection* row.

Table 1 shows that using features from the projected trees provides a big boost to the quality of the statistical parser. Furthermore, the enhancements laid out in Section 3 improve the performance of both the projection algorithm and the parser that uses features from projected trees. The degree of improvement may depend on the properties of a particular language pair and the labeled data we

- (a) The accuracies of the original projection algorithm (the *Baselin* rows) and the enhanced algorithm (the *Best* row) on eight language pairs. For each language, the best performing baseline is in italic. The last row shows the error reduction of the Best row over the best performing baseline, which is calculated by the formula  $ErrorRate = \frac{Best - BestBaseline}{100 - BestBaseline} \times 100$

|                  | YAQ          | WLS          | HIN          | KKN          | GLI          | HUA          | GER          | MEX          |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Best             | 88.03        | 94.90        | 77.44        | 91.75        | 87.70        | 90.11        | 88.71        | 93.05        |
| Baseline (Right) | <i>87.28</i> | <i>89.80</i> | 57.48        | <i>90.34</i> | <i>86.90</i> | 79.31        | <i>88.03</i> | <i>89.57</i> |
| Baseline (Left)  | 84.29        | <i>89.80</i> | <i>68.11</i> | 88.93        | 76.98        | <i>79.54</i> | <i>88.03</i> | <i>89.57</i> |
| Error Reduction  | 5.90         | 50.00        | 29.26        | 14.60        | 6.11         | 51.66        | 5.68         | 33.37        |

- (b) The parsing accuracies of the MSTParser with or without new features extracted from projected trees. There are two error reduction rows: one is with respect to the best performing baseline for each language, the other is with respect to *No Projection* where the parser does not use features from projected trees.

|                                 | YAQ          | WLS          | HIN          | KKN          | GLI          | HUA          | GER          | MEX          |
|---------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Best                            | 89.28        | 94.90        | 81.35        | 92.96        | 81.35        | 88.74        | 92.93        | 93.05        |
| Baseline (Right)                | <i>88.28</i> | <i>94.22</i> | 78.03        | <i>92.35</i> | <i>80.95</i> | 87.59        | 90.48        | <i>92.43</i> |
| Baseline (Left)                 | 87.88        | 94.22        | <i>79.64</i> | 90.95        | <i>80.95</i> | <i>89.20</i> | <i>90.48</i> | <i>92.43</i> |
| No Projection                   | 66.08        | 91.32        | 65.16        | 80.75        | 55.16        | 72.22        | 62.72        | 73.03        |
| Error Reduction (BestBaseline)  | 8.53         | 11.76        | 8.40         | 7.97         | 2.10         | -4.26        | 25.74        | 8.19         |
| Error Reduction (No Projection) | 68.39        | 41.24        | 46.47        | 63.43        | 58.41        | 59.47        | 81.04        | 74.23        |

Table 1: System performance on eight languages: Yaqui (YAQ), Welsh (WLS), Hindi (HIN), Korean (KKN), Gaelic (GLI), Hausa (HUA), German (GER), and Malagasy (MEX).

have for that language pair. For instance, swap is quite common for the Hindi-English pair because postpositions depend on nouns in Hindi whereas nouns depend on prepositions in English. As a result, the enhancement for the swapped alignment alone results in a large error reduction, as in Table 2. This table shows the projection accuracy on the Hindi data when each of the three enhancements is turned on or off. The rows are sorted by descending overall accuracy, and the row that corresponds to the system labeled “Best” in Table 1 is in bold.

## 5 Conclusion

Existing projection algorithms suffer from the effects of structural divergence between language pairs. We propose to learn common divergence types from a small number of tree pairs and use the learned rules to improve projection accuracy. Our experiments show notable gains for both projection and parsing when tested on eight language pairs. As IGT data is available for hundreds of languages through the ODIN database and other sources, one could produce a small parallel treebank for a language pair after spending a few hours manually correcting the output of a projection algorithm. From the treebank, a better projection algorithm and a better parser can be built automatically using our approach.

| Spont | Swap | Merge Direction | Accuracy     |
|-------|------|-----------------|--------------|
| ✓     | ✓    | Left            | 78.07        |
| ✓     | ✓    | <b>Informed</b> | <b>77.44</b> |
|       | ✓    | Left            | 76.69        |
|       | ✓    | Informed        | 76.06        |
| ✓     |      | Left            | 69.49        |
| ✓     |      | Informed        | 68.96        |
|       |      | Left            | 68.11        |
|       |      | Informed        | 67.58        |
| ✓     | ✓    | Right           | 66.32        |
|       | ✓    | Right           | 64.97        |
| ✓     |      | Right           | 58.84        |
|       |      | Right           | 57.48        |

Table 2: Projection accuracy on the Hindi data, with the three enhancements turning on or off. The “spont” and “swap” columns show a checkmark when the enhancements are turned on. The merge direction indicates whether a left or right choice was made as a baseline, or whether the choice was *informed* by the rules learned from the training data.

While the improvements for some languages are incremental, the scope of coverage for this method is potentially enormous, enabling the rapid creation of tools for under-resourced languages of all kinds at a minimal cost.

## Acknowledgment

This work is supported by the National Science Foundation Grant BCS-0748919. We would also like to thank the reviewers for helpful comments.

## References

- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. A multi-representational and multi-layered treebank for Hindi/Urdu. In *ACL-IJCNLP '09: Proceedings of the Third Linguistic Annotation Workshop*. Association for Computational Linguistics, August 2009.
- Bonnie Jean Dorr. Machine translation divergences: a formal description and proposed solution. *Computational Linguistics*, 20:597–633, December 1994.
- R. Georgi, F Xia, and W D Lewis. Measuring the Divergence of Dependency Structures Cross-Linguistically to Improve Syntactic Projection Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, May 2012a.
- Ryan Georgi, Fei Xia, and William D Lewis. Improving Dependency Parsing with Interlinear Glossed Text and Syntactic Projection. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, Mumbai, India, December 2012b.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 1(1):1–15, 2004.
- William D Lewis and Fei Xia. Developing ODIN: A Multilingual Repository of Annotated Language Data for Hundreds of the World’s Languages. 2010.
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. Non-projective dependency parsing using spanning tree algorithms. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530, 2005.
- Chris Quirk, Arul Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Microsoft Research, 2005.
- Fei Xia and William D Lewis. Multilingual Structural Projection across Interlinear Text. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2007.
- David Yarowsky and Grace Ngai. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second meeting of the North American Association for Computational Linguistics (NAACL)*, Stroudsburg, PA, 2001. Johns Hopkins University.

# Cross-lingual Projections between Languages from Different Families

Mo Yu<sup>1</sup> Tiejun Zhao<sup>1</sup> Yalong Bai<sup>1</sup> Hao Tian<sup>2</sup> Dianhai Yu<sup>2</sup>

<sup>1</sup>School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China  
{yumo,tjzhao,ylbai}@mtlab.hit.edu.cn

<sup>2</sup>Baidu Inc., Beijing, China  
{tianhao,yudianhai}@baidu.com

## Abstract

Cross-lingual projection methods can benefit from resource-rich languages to improve performances of NLP tasks in resources-scarce languages. However, these methods confronted the difficulty of syntactic differences between languages especially when the pair of languages varies greatly. To make the projection method well-generalize to diverse languages pairs, we enhance the projection method based on word alignments by introducing target-language word representations as features and proposing a novel noise removing method based on these word representations. Experiments showed that our methods improve the performances greatly on projections between English and Chinese.

## 1 Introduction

Most NLP studies focused on limited languages with large sets of annotated data. English and Chinese are examples of these resource-rich languages. Unfortunately, it is impossible to build sufficient labeled data for all tasks in all languages. To address NLP tasks in resource-scarce languages, cross-lingual projection methods were proposed, which make use of existing resources in resource-rich language (also called *source language*) to help NLP tasks in resource-scarce language (also named as *target language*).

There are several types of projection methods. One intuitive and effective method is to build a common feature space for all languages, so that the model trained on one language could be directly used on other languages (McDonald et al., 2011; Täckström et al., 2012). We call it *direct projection*, which becomes very popular recently. The main limitation of these methods is

that target language has to be similar to source language. Otherwise the performance will degrade especially when the orders of phrases between source and target languages differ a lot.

Another common type of projection methods map labels from resource-rich language sentences to resource-scarce ones in a parallel corpus using word alignment information (Yarowsky et al., 2001; Hwa et al., 2005; Das and Petrov, 2011). We refer them as *projection based on word alignments* in this paper. Compared to other types of projection methods, this type of methods is more robust to syntactic differences between languages since it trained models on the target side thus following the topology of the target language.

This paper aims to build an accurate projection method with strong generality to various pairs of languages, even when the languages are from different families and are typologically divergent. As far as we know, only a few works focused on this topic (Xia and Lewis 2007; Täckström et al., 2013). We adopted the projection method based on word alignments since it is less affected by language differences. However, such methods also have some disadvantages. Firstly, the models trained on projected data could only cover words and cases appeared in the target side of parallel corpus, making it difficult to generalize to test data in broader domains. Secondly, the performances of these methods are limited by the accuracy of word alignments, especially when words between two languages are not one-one aligned. So the obtained labeled data contains a lot of noises, making the models built on them less accurate.

This paper aims to build an accurate projection method with strong generality to various pairs of languages. We built the method on top of projection method based on word alignments because of its advantage of being less affected by syntactic differences, and proposed two solutions to solve the above two difficulties of this type of methods.



Firstly, we introduce Brown clusters of target language to make the projection models cover broader cases. Brown clustering is a kind of word representations, which assigns word with similar functions to the same cluster. They can be efficiently learned on large-scale unlabeled data in target language, which is much easier to acquire even when the scales of parallel corpora of minor languages are limited. Brown clusters have been first introduced to the field of cross-lingual projections in (Täckström et al., 2012) and have achieved great improvements on projection between European languages. However, their work was based on the direct projection methods so that it do not work very well between languages from different families as will be shown in Section 3.

Secondly, to reduce the noises in projection, we propose a noise removing method to detect and correct noisy projected labels. The method was also built on Brown clusters, based on the assumption that instances with similar representations of Brown clusters tend to have similar labels. As far as we know, no one has done any research on removing noises based on the space of word representations in the field of NLP.

Using above techniques, we achieved a projection method that adapts well on different language pairs even when the two languages differ enormously. Experiments of NER and POS tagging projection from English to Chinese proved the effectiveness of our methods.

In the rest of our paper, Section 2 describes the proposed cross-lingual projection method. Evaluations are in Section 3. Section 4 gives concluding remarks.

## 2 Proposed Cross-lingual Projection Methods

In this section, we first briefly introduce the cross-lingual projection method based on word alignments. Then we describe how the word representations (Brown clusters) were used in the projection method. Section 2.3 describes the noise removing methods.

### 2.1 Projection based on word alignments

In this paper we consider cross-lingual projection based on word alignment, because we want to build projection methods that can be used between language pairs with large differences. Figure 1 shows the procedure of cross-lingual projec-

tion methods, taking projection of NER from English to Chinese as an example. Here English is the resource-rich language and Chinese is the target language. First, sentences from the source side of the parallel corpus are labeled by an accurate model in English (e.g., "Rongji Zhu" and "Gan Luo" were labeled as "PER"), since the source language has rich resources to build accurate NER models. Then word alignments are generated from the parallel corpus and serve as a bridge, so that unlabeled words in the target language will get the same labels with words aligning to them in the source language, e.g. the first word '朱(金容)基' in Chinese gets the projected label 'PER', since it is aligned to "Rongji" and "Zhu". In this way, labels in source language sentences are projected to the target sentences.



Figure 1: An example of projection of NER. Labels of Chinese sentence (right) in brackets are projected from the source sentence.

From the projection procedure we can see that a labeled dataset of target language is built based on the projected labels from source sentences. The projected dataset has a large size, but with a lot of noises. With this labeled dataset, models of the target language can be trained in a supervised way. Then these models can be used to label sentences in target language. Since the models are trained on the target language, this projection approach is less affected by language differences, comparing with direct projection methods.

### 2.2 Word Representation features for Cross-lingual Projection

One disadvantage of above method is that the coverage of projected labeled data used for training

|            |  |
|------------|--|
| Words      | $w_{i,i \in \{-2:2\}}, w_{i-1}/w_{i,i \in \{0,1\}}$              |
| Cluster    | $c_{i,i \in \{-2:2\}}, c_{i-1}/c_{i,i \in \{-1,2\}}, c_{-1}/c_1$ |
| Transition | $y_{-1}/y_0/\{w_0, c_0, c_{-1}/c_1\}$                            |

Table 1: NER features.  $c_i$  is the cluster id of  $w_i$ .

target language models are limited by the coverage of parallel corpora. For example in Figure 1, some Chinese politicians in 1990’s will be learned as person names, but some names of recent politicians such as “Obama”, which did not appeared in the parallel corpus, would not be recognized.

To broader the coverage of the projected data, we introduced word representations as features. Same or similar word representations will be assigned to words appearing in similar contexts, such as person names. Since word representations are trained on large-scale unlabeled sentences in target language, they cover much more words than the parallel corpus does. So the information of a word in projected labeled data will apply to other words with the same or similar representations, even if they did not appear in the parallel data.

In this work we use Brown clusters as word representations on target languages. Brown clustering assigns words to hierarchical clusters according to the distributions of words before and after them. Taking NER as an example, the feature template may contain features shown in Table 1. The cluster id of the word to predict ( $c_0$ ) and those of context words ( $c_i, i \in \{-2, -1, 1, 2\}$ ), as well as the conjunctions of these clusters were used as features in CRF models in the same way the traditional word features were used. Since Brown clusters are hierarchical, the cluster for each word can be represented as a binary string. So we also use prefix of cluster IDs as features, in order to compensate for clusters containing small number of words. For languages lacking of morphological changes, such as Chinese, there are no pre/suffix or orthography features. However the cluster features are always available for any languages.

### 2.3 Noise Removing in Word Representation Space

Another disadvantage of the projection method is that the accuracy of projected labels is badly affected by non-literate translation and word alignment errors, making the data contain many noises. For example in Figure 1, the word “吴仪(Wu Yi)” was not labeled as a named entity since it was

not aligned to any words in English due to the alignment errors. A more accurate model will be trained if such noises can be reduced.

A direct way to remove the noises is to modify the label of a word to make it consistent with the majority of labels assigned to the same word in the parallel corpus. The method is limited when a word with low frequency has many of its appearances incorrectly labeled because of alignment errors. In this situation the noises are impossible to remove according to the word itself. The error in Figure 1 is an example of this case since the other few occurrences of the word “吴仪(Wu Yi)” also happened to fail to get the correct label.

Such difficulties can be easily solved when we turned to the space of Brown clusters, based on the observation that words in a same cluster tend to have same labels. For example in Figure 1, the word “吴仪(Wu Yi)”, “朱(金容)基(Zhu Rongji)” and “罗干(Luo Gan)” are in the same cluster, because they are all names of Chinese politicians and usually appear in similar contexts. Having observed that a large portion of words in this cluster are person names, it is reasonable to modified the label of “吴仪(Wu Yi)” to “PER”.

The space of clusters is also less sparse so it is also possible to use combination of the clusters to help noise removing, in order to utilize the context information of data instances. For example, we could represent a instance as bigram of the cluster of target word and that of the previous word. And it is reasonable that its label should be same with other instances with the same cluster bigrams.

The whole noise removing method can be represented as following: Suppose a target word  $w_i$  was assigned label  $y_i$  during projection with probability of alignment  $p_i$ . From the whole projected labeled data, we can get the distribution  $p_w(y)$  for the word  $w_i$ , the distribution  $p_c(y)$  for its cluster  $c_i$  and the distribution  $p_b(y)$  for the bigram  $c_{i-1}c_i$ . We choose  $y'_i = y'$ , which satisfies

$$y' = \operatorname{argmax}_y (\delta_{y,y_i} p_i + \sum_{x \in \{w,c,b\}} p_x(y)) \quad (1)$$

$\delta_{y,y_i}$  is an indicator function, which is 1 when  $y$  equals to  $y_i$ . In practices, we set  $p_{w/c/b}(y)$  to 0 for the  $y$ s that make the probability less than 0.5. With the noise removing method, we can build a more accurate labeled dataset based on the projected data and then use it for training models.

### 3 Experimental Results

#### 3.1 Data Preparation

We took English as resource-rich language and used Chinese to imitate resource-scarce languages, since the two languages differ a lot. We conducted experiments on projections of NER and POS tagging. The resource-scarce languages were assumed to have no training data. For the NER experiments, we used data from People’s Daily (April, 1998) as test data (55,177 sentences). The data was converted following the style of Penn Chinese Treebank (CTB) (Xue et al., 2005). For evaluation of projection of POS tagging, we used the test set of CTB. Since English and Chinese have different annotation standards, labels in the two languages were converted to the universal POS tag set (Petrov et al., 2011; Das and Petrov, 2011) so that the labels between the source and target languages were consistent. The universal tag set made the task of POS tagging easier since the fine-grained types are no more cared.

The Brown clusters were trained on Chinese Wikipedia. The bodies of all articles are retained to induce 1000 clusters using the algorithm in (Liang, 2005). Stanford word segmentor (Tseng et al., 2005) was used for Chinese word segmentation. When English Brown clusters were in need, we trained the word clusters on the tokenized English Wikipedia.

We chose LDC2003E14 as the parallel corpus, which contains about 200,000 sentences. GIZA++ (Och and Ney, 2000) was used to generate word alignments. It is easier to obtain similar amount of parallel sentences between English and minor languages, making the conclusions more general for problems of projection in real applications.

#### 3.2 Performances of NER Projection

Table 2 shows the performances of NER projection. We re-implemented the direct projection method with projected clusters in (Täckström et al., 2012). Although their method was proven to work well on European language pairs, the results showed that projection based on word alignments (WA) worked much better since the source and target languages are from different families.

After we add the clusters trained on Chinese Wikipedia as features as in Section 2.2, a great improvement of about 9 points on the average F1-score of the three entity types was achieved, showing that the word representation features help to

| System             | avg<br>Prec | avg<br>Rec | avg<br>F1    |
|--------------------|-------------|------------|--------------|
| Direct projection  | 47.48       | 28.12      | 33.91        |
| Proj based on WA   | 71.6        | 37.84      | 47.66        |
| +clusters(from en) | 63.96       | 46.59      | 53.75        |
| +clusters(ch wiki) | 73.44       | 47.63      | <b>56.60</b> |

Table 2: Performances of NER projection.

recall more named entities in the test set. The performances of all three categories of named entities were improved greatly after adding word representation features. Larger improvements were observed on person names (14.4%). One of the reasons for the improvements is that in Chinese, person names are usually single words. Thus Brown-clustering method can learn good word representations for those entities. Since in test set, most entities that are not covered are person names, Brown clusters helped to increase the recall greatly.

In (Täckström et al., 2012), Brown clusters trained on the source side were projected to the target side based on word alignments. Rather than building a same feature space for both the source language and the target language as in (Täckström et al., 2012), we tried to use the projected clusters as features in projection based on word alignments. In this way the two methods used exactly the same resources. In the experiments, we tried to project clusters trained on English Wikipedia to Chinese words. They improved the performance by about 6.1% and the result was about 20% higher than that achieved by the direct projection method, showing that even using exactly the same resources, the proposed method outperformed that in (Täckström et al., 2012) much on diverse language pairs.

Next we studied the effects of noise removing methods. Firstly, we removed noises according to Eq(1), which yielded another huge improvement of about 6% against the best results based on cluster features. Moreover, we conducted experiments to see the effects of each of the three factors. The results show that both the noise removing methods based on words and on clusters achieved improvements between 1.5-2 points. The method based on bigram features got the largest improvement of 3.5 points. It achieved great improvement on person names. This is because a great proportion of the vocabulary was made up of person names, some of which are mixed in clusters with common nouns.

While noise removing method based on clusters failed to recognize them as name entities, cluster bigrams will make use of context information to help the discrimination of these mixed clusters.

| System      | PER   | LOC   | ORG   | AVG          |
|-------------|-------|-------|-------|--------------|
| By Eq(1)    | 59.77 | 55.56 | 72.26 | <b>62.53</b> |
| By clusters | 49.75 | 53.10 | 72.46 | 58.44        |
| By words    | 49.00 | 54.69 | 70.59 | 58.09        |
| By bigrams  | 58.39 | 55.01 | 66.88 | 60.09        |

Table 3: Performances of noise removing methods

### 3.3 Performances of POS Projection

In this section we test our method on projection of POS tagging from English to Chinese, to show that our methods can well extend to other NLP tasks. Unlike named entities, POS tags are associated with single words. When one target word is aligned to more than one words with different POS tags on the source side, it is hard to decide which POS tag to choose. So we only retained the data labeled by 1-to-1 alignments, which also contain less noises as pointed out by (Hu et al., 2011). The same feature template as in the experiments of NER was used for training POS taggers.

The results are listed in Table 4. Because of the great differences between English and Chinese, projection based on word alignments worked better than direct projection did. After adding word cluster features and removing noises, an error reduction of 12.7% was achieved.

POS tagging projection can benefit more from our noise removing methods than NER projection could, i.e. noise removing gave rise to a higher improvement (2.7%) than that achieved by adding cluster features on baseline system (1.5%). One possible reason is that our noise removing methods assume that labels are associated with single words, which is more suitable for POS tagging.

| Methods                       | Accuracy     |
|-------------------------------|--------------|
| Direct projection (Täckström) | 62.71        |
| Projection based on WA        | 66.68        |
| +clusters (ch wiki)           | 68.23        |
| +cluster(ch)&noise removing   | <b>70.92</b> |

Table 4: Performances of POS tagging projection.

## 4 Conclusion and perspectives

In this paper we introduced Brown clusters of target languages to cross-lingual projection and proposed methods for removing noises on projected labels. Experiments showed that both the two techniques could greatly improve the performances and could help the projection method well generalize to languages differ a lot.

Note that although projection methods based on word alignments are less affected by syntactic differences, the topological differences between languages still remain an importance reason for the limitation of performances of cross-lingual projection. In the future we will try to make use of representations of sub-structures to deal with syntactic differences in more complex tasks such as projection of dependency parsing. Future improvements also include combining the direct projection methods based on joint feature representations with the proposed method as well as making use of projected data from multiple languages.

### Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work was supported by National Natural Science Foundation of China (61173073), and the Key Project of the National High Technology Research and Development Program of China (2011AA01A207).

### References

- P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra, and J.C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.
- D. Das and S. 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.
- P.L. Hu, M. Yu, J. Li, C.H. Zhu, and T.J. Zhao. 2011. Semi-supervised learning framework for cross-lingual projection. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 3, pages 213–216. IEEE.
- R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–326.

- W. Jiang and Q. Liu. 2010. Dependency parsing and projection based on word-pair classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL*, volume 10, pages 12–20.
- P. Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.
- R. McDonald, S. Petrov, and K. Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics.
- F.J. Och and H. Ney. 2000. Giza++: Training of statistical translation models.
- S. Petrov, D. Das, and R. McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.
- O. Täckström, R. McDonald, and J. Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure.
- O. Täckström, R. McDonald, and J. Nivre. 2013. Target language adaptation of discriminative transfer parsers. *Proceedings of NAACL-HLT*.
- H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, volume 171. Jeju Island, Korea.
- F. Xia and W. Lewis. 2007. Multilingual structural projection across interlinear text. In *Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007)*, pages 452–459.
- N. Xue, F. Xia, F.D. Chiou, and M. Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207.
- D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

# Using Context Vectors in Improving a Machine Translation System with Bridge Language

Samira Tofighi Zahabi    Somayeh Bakhshaei    Shahram Khadivi

Human Language Technology Lab  
Amirkabir University of Technology  
Tehran, Iran

{Samiratofighi,bakhshaei,khadivi}@aut.ac.ir

## Abstract

Mapping phrases between languages as translation of each other by using an intermediate language (pivot language) may generate translation pairs that are wrong. Since a word or a phrase has different meanings in different contexts, we should map source and target phrases in an intelligent way. We propose a pruning method based on the context vectors to remove those phrase pairs that connect to each other by a polysemous pivot phrase or by weak translations. We use context vectors to implicitly disambiguate the phrase senses and to recognize irrelevant phrase translation pairs.

Using the proposed method a relative improvement of 2.8 percent in terms of BLEU score is achieved.

## 1 Introduction

Parallel corpora as an important component of a statistical machine translation system are unfortunately unavailable for all pairs of languages, particularly in low resource languages and also producing it consumes time and cost. So, new ideas have been developed about how to make a MT system which has lower dependency on parallel data like using comparable corpora for improving performance of a MT system with small parallel corpora or making a MT system without parallel corpora. Comparable corpora have segments with the same translations. These segments might be in the form of words, phrases or sentences. So, this extracted information can be added to the parallel corpus or might be used for adaptation of the language model or translation model.

Comparable corpora are easily available resources. All texts that are about the same topic can be considered as comparable corpora. Another idea for solving the scarce resource

problem is to use a high resource language as a pivot to bridge between source and target languages. In this paper we use the bridge technique to make a source-target system and we will prune the phrase table of this system. In Section 2, the related works of the bridge approach are considered, in Section 3 the proposed approach will be explained and it will be shown how to prune the phrase table using context vectors, and experiments on German-English-Farsi systems will be presented in Section 4.

## 2 Related Works

There are different strategies of bridge techniques to make a MT system. The simplest way is to build two MT systems in two sides: one system is source-pivot and the other is pivot-target system, then in the translation stage the output of the first system is given to the second system as an input and the output of the second system is the final result. The disadvantage of this method is its time consuming translation process, since until the first system's output is not ready; the second system cannot start the translation process. This method is called *cascading of two translation systems*.

In the other approach the target side of the training corpus of the source-pivot system is given to the pivot-target system as its input. The output of the pivot-target system is parallel with the source side of the training corpus of the source-pivot system. A source-to-target system can be built by using this noisy parallel corpus which in it each source sentence is directly translated to a target sentence. This method is called the *pseudo corpus approach*.

Another way is combining the phrase tables of the source-pivot and pivot-target systems to directly make a source-target phrase table. This combination is done if the pivot phrase is

identical in both phrase tables. Since one phrase has many translations in the other language, a large phrase table will be produced. This method is called *combination of phrase tables approach*.

Since in the bridge language approach two translation systems are used to make a final translation system, the errors of these two translation systems will affect the final output. Therefore in order to decrease the propagation of these errors, a language should be chosen as pivot which its structure is similar to the source and target languages. But even by choosing a good language as pivot there are some other errors that should be handled or decreased such as the errors of ploysemous words and etc.

For making a MT system using pivot language several ideas have been proposed. Wu and Wang (2009) suggested a cascading method which is explained in Section 1.

Bertoldi (2008) proposed his method in bridging at translation time and bridging at training time by using the cascading method and the combination of phrase tables.

Bakhshaei (2010) used the combination of phrase tables of source-pivot and pivot-target systems and produced a phrase table for the source-target system.

Paul (2009) did several experiments to show the effect of pivot language in the final translation system. He showed that in some cases if training data is small the pivot should be more similar to the source language, and if training data is large the pivot should be more similar to the target language. In Addition, it is more suitable to use a pivot language that its structure is similar to both of source and target languages.

Saralegi (2011) showed that there is not transitive property between three languages. So many of the translations produced in the final phrase table might be wrong. Therefore for pruning wrong and weak phrases in the phrase table two methods have been used. One method is based on the structure of source dictionaries and the other is based on distributional similarity.

Rapp (1995) suggested his idea about the usage of context vectors in order to find the words that are the translation of each other in comparable corpora.

In this paper the combination of phrase tables approach is used to make a source-target system. We have created a base source-target system just similar to previous works. But the contribution of our work compared to other works is that here we decrease the size of the produced phrase table and improve the performance of the system. Our

pruning method is different from the method that Saralegi (2011) has used. He has pruned the phrase table by computing distributional similarity from comparable corpora or by the structure of source dictionaries. Here we use context vectors to determine the concept of phrases and we use the pivot language to compare source and target vectors.

### 3 Approach

For the purpose of showing how to create a pruned phrase table, in Section 3.1 we will explain how to create a simple source-to-target system. In Section 3.2 we will explain how to remove wrong and weak translations in the pruning step. Figure 1 shows the pseudo code of the proposed algorithm.

In the following we have used these abbreviations: **f**, **e** stands for source and target phrases. **pl**, **src-pl**, **pl-trg**, **src-trg** respectively stand for pivot phrase, source-pivot phrase table, pivot-target phrase table and source-target phrase table.

#### 3.1 Creating source-to-target system

At first, we assume that there is transitive property between three languages in order to make a base system, and then we will show in different ways that there is not transitive property between three languages.

```

for each source phrase f
  pls = { translations of f in src-pl }
  for each pl in pls
    Es = { translations of pl in pl-trg }
    for each e in Es
       $p(\mathbf{e}|\mathbf{f}) = p(\mathbf{pl}|\mathbf{f}) * p(\mathbf{e}|\mathbf{pl})$  and add (e,f) to src-trg
create source-to-destination system with src-trg
create context vector V for each source phrase f
using source corpora
create context vector V' for each target phrase e
using target corpora
convert Vs to pivot language vectors using src-pl
system
convert V' s to pivot language vectors using pl-trg
system
for each f in src-trg
  Es = { translations of f in src-trg }
  For each e in Es calculate similarity of its context
  vector with f context vector
  Select k top similar as translations of f
  delete other translations of f in src-trg

```

Figure 1. Pseudo code for proposed method

For each phrase **f** in **src-pl** phrase table, all the phrases **pl** which are translations of **f**, are considered. Then for each of these **pls** every phrase **e** from the **pl-trg** phrase table that are translations of **pl**, are found. Finally **f** is mapped to all of these **es** in the new **src-trg** phrase table.

The probability of these new phrases is calculated using equation (1) through the algorithm that is shown in figure 1.

$$p(e|f) = p(pl|f) \times p(e|pl) \quad (1)$$

A simple **src-trg** phrase table is created by this approach. **Pl** phrases might be ploysemous and produce target phrases that have different meaning in comparison to each other. The concept of some of these target phrases are similar to the corresponding source phrase and the concept of others are irrelevant to the source phrase.

The language model can ignore some of these wrong translations. But it cannot ignore these translations if they have high probability.

Since the probability of translations is calculated using equation (1), therefore wrong translations have high probability in three cases: first when  $p(\mathbf{pl}|\mathbf{f})$  is high, second when  $p(\mathbf{e}|\mathbf{pl})$  is high and third when  $p(\mathbf{pl}|\mathbf{f})$  and  $p(\mathbf{e}|\mathbf{pl})$  are high.

In the first case **pl** might be a good translation for **f** and refers to concept *c*, but **pl** and **e** refer to concept *c'* so mapping **f** to **e** as a translation of each other is wrong. The second case is similar to the first case but **e** might be a good translation for **pl**. The third case is also similar to the first case, but **pl** is a good translation for both **f** and **e**.

The pruning method that is explained in Section 3.2, tries to find these translations and delete them from the **src-trg phrase table**.

### 3.2 Pruning method

To determine the concept of each phrase (*p*) in language L at first a vector (*V*) with length N is created. Each element of *V* is set to zero and N is the number of unique phrases in language L.

In the next step all sentences of the corpus in language L are analyzed. For each phrase *p* if *p* occurs with *p'* in the same sentence the element of context vector *V* that corresponds to *p'* is pulsed by 1. This way of calculating context vectors is similar to Rapp (1999), but here the window length of phrase co-occurrence is considered a sentence. Two phrases are considered as co-occurrence if they occur in the same sentence. The distance between them does not matter. In other words phrase *p* might be at the beginning of the sentence while *p'* being at

the end of the sentence, but they are considered as co-occurrence phrases.

For each source (target) phrase its context vector should be calculated within the source (target) corpus as shown in figure 1.

The number of unique phrases in the source (target) language is equal to the number of unique source (target) phrases in the **src-trg** phrase table that are created in the last Section.

So, the length of source context vectors is **m** and the length of target context vectors is **n**. These variables (**m** and **n**) might not be equal. In addition to this, source vectors and target vectors are in two different languages, so they are not comparable.

One method to translate source context vectors to target context vectors is using an additional source-target dictionary. But instead here, source and target context vectors are translated to pivot context vectors. In other words if source context vectors have length **m** and target context vectors have length **n**, they are converted to pivot context vectors with length **z**. The variable **z** is the number of unique pivot phrases in **src-pl** or **pl-trg** phrase tables.

To map the source context vector  $S(s_1, s_2, \dots, s_m)$  to the pivot context vector, we use a fixed size vector  $V_1^z$ . Elements of vector  $V_1^z = (v_1, v_2, \dots, v_z)$  are the unique phrases extracted from **src-pl** or **pl-trg** phrase tables.

$$V_1^z = (v_1, v_2, \dots, v_z) = (0, 0, \dots, 0)$$

In the first step  $v_i$ s are set to 0. For each element,  $s_i$ , of vector *S* if  $s_i > 0$  it will be translated to *k* pivot phrases. These phrases are the output of *k*-best translations of  $s_i$  by using the **src-pl** phrase table.

$$s_i \xrightarrow{\text{src-pl phrase table}} \{V_1^{k'} = (v'_1, v'_2, \dots, v'_k)\}$$

For each element  $v'$  of  $V_1^{k'}$  its corresponding element  $v$  of  $V_1^z$  which are equal, will be found, then the amount of  $v$  will be increased by  $s_i$ .

$$\forall v' \in V_1^{k'} \text{ find } (v \in V_1^z) \ni v = v' \\ val(v) \leftarrow val(v) + s_i$$

Using *K*-best translations as middle phrases is for reducing the effect of translation errors that cause wrong concepts. This work is done for each target context vector. Source and target context vectors will be mapped to identical length vectors and are also in the same language (pivot language). Now source and target context vectors are comparable, so with a simple similarity metric their similarity can be calculated.

Here we use cosine similarity. The similarity between each source context vector and each



target context vector that are translations of the source phrase in **src-trg, are calculated**. For each source phrase, the **N**-most similar target phrases are kept as translations of the source phrase. These translations are also similar in context. Therefore this pruning method deletes irrelevant translations from the **src-trg** phrase table. The size of the phrase table is decreased very much and the system performance is increased. Reduction of the phrase table size is considerable while its performance is increased.

## 4 Experiments

In this work, we try to make a German-Farsi system without using parallel corpora. We use English language as a bridge between German and Farsi languages because English language is a high resource language and parallel corpora of German-English and English-Farsi are available.

We use Moses<sup>1</sup> (Koehn et al., 2007) as the MT decoder and IRSTLM<sup>2</sup> tools for making the language model. Table 1 shows the statistics of the corpora that we have used in our experiments. The German-English corpus is from Verbmobil project (Ney et al., 2000). We manually translate 22K English sentences to Farsi to build a small Farsi-English-German corpus. Therefore, we have a small English-German corpus as well.

With the German-English parallel corpus and an additional German-English dictionary with 118480 entries we have made a German-English (**De-En**) system and with English-Farsi parallel corpus we have made a German-Farsi (**En-Fa**) system. The BLEU score of these systems are shown in Table 1.

Now, we create a translation system by combining phrase tables of De-En and En-Fa systems. Details of creating the source-target system are explained in Section 3.1. The size of this phrase table is very large because of polysemous and some weak translations.

|                | Sentences | BLEU |
|----------------|-----------|------|
| German-English | 58,073    | 40.1 |
| English-Farsi  | 22,000    | 31.6 |

Table 1. Information of two parallel systems that are used in our experiments.

The size of the phrase table is about 55.7 MB. Then, we apply the pruning method that we

<sup>1</sup>Available under the LGPL from

<http://sourceforge.net/projects/mosesdecoder/>

<sup>2</sup>Available under the LGPL from

<http://hlt.fbk.eu/en/irstlm>

explained in Section 3.2. With this method only the phrases are kept that their context vectors are similar to each other. For each source phrase the 35-most similar target translations are kept. The number of phrases in the phrase table is decreased dramatically while the performance of the system is increased by 2.8 percent BLEU. The results of these experiments are shown in Table 2. The last row in this table is the result of using small parallel corpus to build German-Farsi system. We observe that the pruning method has gain better results compared to the system trained on the parallel corpus. This is maybe because of some translations that are made in the parallel system and do not have enough training data and their probabilities are not precise. But when we use context vectors to measure the contextual similarity of phrases and their translations, the impact of these training samples are decreased. In Table 3, two wrong phrase pairs that pruning method has removed them are shown.

|                    | BLEU | # of phrases |
|--------------------|------|--------------|
| Base bridge system | 25.1 | 500,534      |
| Pruned system      | 27.9 | 26,911       |
| Parallel system    | 27.6 | 348,662      |

Table 2. The MT results of the base system, the pruned system and the parallel system.

| German phrase          | Wrong translation | Correct translation |
|------------------------|-------------------|---------------------|
| vorschlagen , wir      | به تاثر           | پیشنهاد میکنیم      |
| um neun Uhr<br>morgens | ده                | ساعت نه صبح         |

Table 3. Sample wrong translations that the pruning method removed them.

In Table 4, we extend the experiments with two other methods to build German-Farsi system using English as bridging language. We see that the proposed method obtains competitive result with the pseudo parallel method.

| System                    | BLEU | size (MB) |
|---------------------------|------|-----------|
| Phrase tables combination | 25.1 | 55.7      |
| Cascade method            | 25.2 | NA        |
| Pseudo parallel corpus    | 28.2 | 73.2      |
| Phrase tables comb.+prune | 27.9 | 3.0       |

Table 4. Performance results of different ways of bridging

Now, we run a series of significance tests to measure the superiority of each method. In the first significance test, we set the pruned system as our base system and we compare the result of the pseudo parallel corpus system with it, the significance level is 72%. For another significance test we set the combined phrase table system without pruning as our base system and we compare the result of the pruned system with it, the significance level is 100%. In the last significance test we set the combined phrase table system without pruning as our base system and we compare the result of the pseudo system with it, the significance level is 99%. Therefore, we can conclude the proposed method obtains the best results and its difference with pseudo parallel corpus method is not significant.

## 5 Conclusion and future work

With increasing the size of the phrase table, the MT system performance will not necessarily increase. Maybe there are wrong translations with high probability which the language model cannot remove them from the best translations.

By removing these translation pairs, the produced phrase table will be more consistent, and irrelevant words or phrases are much less. In addition, the performance of the system will be increased by about 2.8% BLEU.

In the future work, we investigate how to use the word alignments of the source-to-pivot and pivot-to-target systems to better recognize good translation pairs.

## References

- 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.
- Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-Based Statistical Machine Translation with Pivot Language. In *Proc. Of IWSLT*, pages 143-149, Hawaii, USA.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *proc. of EMNLP*, pages 388-395, Barcelona, Spain.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris C. 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 *Proc. of ACL Demonstration Session*, pages 177-180, Prague.
- Hermann Ney, Franz J. Och, Stephan Vogel. 2000. Statistical Translation of Spoken Dialogues in the VerbMobil System. In *Workshop on Multi Lingual Speech Communication*, pages 69-74.
- Michael Paul, Hirofumi Yamamoto, Eiichiro Sumita, and Satoshi Nakamura. 2009. On the Importance of Pivot Language Selection for Statistical Machine Translation. In *proc. Of NAACL HLT*, pages 221-224, Boulder, Colorado.
- Reinhard Rapp. 1995. Identifying Word Translations in Non-Parallel Texts. In *proc. Of ACL*, pages 320-322, Stroudsburg, PA, USA.
- Reinhard Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proc. Of ACL*, pages 519-525, Stroudsburg, PA, USA.
- Xabeir Saralegi, Iker Manterola, and Inaki S. Vicente. 2011. Analyzing Methods for Improving Precision of Pivot Based Bilingual Dictionaries. In *proc. of the EMNLP*, pages 846-856, Edinburgh, Scotland.
- Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based SMT. In *Proc. of HLT*, pages 484-491, New York, US.
- Hua Wu and Haifeng Wang. 2007. Pivot Language Approach for Phrase-Based SMT. In *Proc. of ACL*, pages 856-863, Prague, Czech Republic.

# Task Alternation in Parallel Sentence Retrieval for Twitter Translation

Felix Hieber and Laura Jehl and Stefan Riezler

Department of Computational Linguistics

Heidelberg University

69120 Heidelberg, Germany

{jehl,hieber,riezler}@cl.uni-heidelberg.de

## Abstract

We present an approach to mine comparable data for parallel sentences using translation-based cross-lingual information retrieval (CLIR). By iteratively alternating between the tasks of retrieval and translation, an initial general-domain model is allowed to adapt to in-domain data. Adaptation is done by training the translation system on a few thousand sentences retrieved in the step before. Our setup is time- and memory-efficient and of similar quality as CLIR-based adaptation on millions of parallel sentences.

## 1 Introduction

Statistical Machine Translation (SMT) crucially relies on large amounts of bilingual data (Brown et al., 1993). Unfortunately sentence-parallel bilingual data are not always available. Various approaches have been presented to remedy this problem by mining parallel sentences from comparable data, for example by using cross-lingual information retrieval (CLIR) techniques to retrieve a target language sentence for a source language sentence treated as a query. Most such approaches try to overcome the noise inherent in automatically extracted parallel data by sheer size. However, finding good quality parallel data from noisy resources like Twitter requires sophisticated retrieval methods. Running these methods on millions of queries and documents can take weeks.

Our method aims to achieve improvements similar to large-scale parallel sentence extraction approaches, while requiring only a fraction of the extracted data and considerably less computing resources. Our key idea is to extend a straightforward application of translation-based CLIR to an iterative method: Instead of attempting to retrieve in one step as many parallel sentences as possible,

we allow the retrieval model to gradually adapt to new data by using an SMT model trained on the freshly retrieved sentence pairs in the translation-based retrieval step. We alternate between the tasks of translation-based retrieval of target sentences, and the task of SMT, by re-training the SMT model on the data that were retrieved in the previous step. This task alternation is done iteratively until the number of newly added pairs stabilizes at a relatively small value.

In our experiments on Arabic-English Twitter translation, we achieved improvements of over 1 BLEU point over a strong baseline that uses in-domain data for language modeling and parameter tuning. Compared to a CLIR-approach which extracts more than 3 million parallel sentences from a noisy comparable corpus, our system produces similar results in terms of BLEU using only about 40 thousand sentences for training in each of a few iterations, thus being much more time- and resource-efficient.

## 2 Related Work

In the terminology of semi-supervised learning (Abney, 2008), our method resembles self-training and co-training by training a learning method on its own predictions. It is different in the aspect of task alternation: The SMT model trained on retrieved sentence pairs is not used for generating training data, but for scoring noisy parallel data in a translation-based retrieval setup. Our method also incorporates aspects of transductive learning in that candidate sentences used as queries are filtered for out-of-vocabulary (OOV) words and similarity to sentences in the development set in order to maximize the impact of translation-based retrieval.

Our work most closely resembles approaches that make use of variants of SMT to mine comparable corpora for parallel sentences. Recent work uses word-based translation (Munteanu and

Marcu, 2005; Munteanu and Marcu, 2006), full-sentence translation (Abdul-Rauf and Schwenk, 2009; Uszkoreit et al., 2010), or a sophisticated interpolation of word-based and contextual translation of full sentences (Snover et al., 2008; Jehl et al., 2012; Ture and Lin, 2012) to project source language sentences into the target language for retrieval. The novel aspect of task alternation introduced in this paper can be applied to all approaches incorporating SMT for sentence retrieval from comparable data.

For our baseline system we use in-domain language models (Bertoldi and Federico, 2009) and meta-parameter tuning on in-domain development sets (Koehn and Schroeder, 2007).

### 3 CLIR for Parallel Sentence Retrieval

#### 3.1 Context-Sensitive Translation for CLIR

Our CLIR model extends the translation-based retrieval model of Xu et al. (2001). While translation options in this approach are given by a lexical translation table, we also select translation options estimated from the decoder’s  $n$ -best list for translating a particular query. The central idea is to let the language model choose fluent, context-aware translations for each query term during decoding.

For mapping source language query terms to target language query terms, we follow Ture et al. (2012a; 2012). Given a source language query  $Q$  with query terms  $q_j$ , we project it into the target language by representing each source token  $q_j$  by its probabilistically weighted translations. The score of target document  $D$ , given source language query  $Q$ , is computed by calculating the Okapi BM25 rank (Robertson et al., 1998) over projected term frequency and document frequency weights as follows:

$$\begin{aligned} score(D|Q) &= \sum_{j=1}^{|Q|} bm25(tf(q_j, D), df(q_j)) \\ tf(q, D) &= \sum_{i=1}^{|T_q|} tf(t_i, D)P(t_i|q) \\ df(q) &= \sum_{i=1}^{|T_q|} df(t_i)P(t_i|q) \end{aligned}$$

where  $T_q = \{t|P(t|q) > L\}$  is the set of translation options for query term  $q$  with probability greater than  $L$ . Following Ture et al. (2012a; 2012) we impose a cumulative threshold  $C$ , so that only the most probable options are added until  $C$  is reached.

Like Ture et al. (2012a; 2012) we achieved best retrieval performance when translation probabilities are calculated as an interpolation between (context-free) lexical translation probabilities  $P_{lex}$  estimated on symmetrized word alignments, and (context-aware) translation probabilities  $P_{nbest}$  estimated on the  $n$ -best list of an SMT decoder:

$$P(t|q) = \lambda P_{nbest}(t|q) + (1 - \lambda)P_{lex}(t|q) \quad (1)$$

$P_{nbest}(t|q)$  is the decoder’s confidence to translate  $q$  into  $t$  within the context of query  $Q$ . Let  $a_k(t, q)$  be a function indicating an alignment of target term  $t$  to source term  $q$  in the  $k$ -th derivation of query  $Q$ . Then we can estimate  $P_{nbest}(t|q)$  as follows:

$$P_{nbest}(t|q) = \frac{\sum_{k=1}^n a_k(t, q)\mathcal{D}(k, Q)}{\sum_{k=1}^n a_k(\cdot, q)\mathcal{D}(k, Q)} \quad (2)$$

$\mathcal{D}(k, Q)$  is the model score of the  $k$ -th derivation in the  $n$ -best list for query  $Q$ .

In our work, we use hierarchical phrase-based translation (Chiang, 2007), as implemented in the `cdec` framework (Dyer et al., 2010). This allows us to extract word alignments between source and target text for  $Q$  from the SCFG rules used in the derivation. The concept of self-translation is covered by the decoder’s ability to use pass-through rules if words or phrases cannot be translated.

#### 3.2 Task Alternation in CLIR

The key idea of our approach is to iteratively alternate between the tasks of retrieval and translation for efficient mining of parallel sentences. We allow the initial general-domain CLIR model to adapt to in-domain data over multiple iterations. Since our set of in-domain queries was small (see 4.2), we trained an adapted SMT model on the concatenation of general-domain sentences and in-domain sentences retrieved in the step before, rather than working with separate models.

Algorithm 1 shows the iterative task alternation procedure. In terms of semi-supervised learning, we can view algorithm 1 as non-persistent as we do not keep labels/pairs from previous iterations. We have tried different variations of label persistence but did not find any improvements. A similar effect of preventing the SMT model to “forget” general-domain knowledge across iterations is achieved by mixing models from current and previous iterations. This is accomplished in two ways: First, by linearly interpolating the translation option weights  $P(t|q)$  from the current and

---

**Algorithm 1** Task Alternation

---

**Require:** source language Tweets  $Q_{src}$ , target language Tweets  $D_{trg}$ , general-domain parallel sentences  $S_{gen}$ , general-domain SMT model  $M_{gen}$ , interpolation parameter  $\theta$

```
procedure TASK-ALTERNATION( $Q_{src}, D_{trg}, S_{gen}, M_{gen}, \theta$ )  
   $t \leftarrow 1$   
  while true do  
     $S_{in} \leftarrow \emptyset$  ▷ Start with empty parallel in-domain sentences  
    if  $t == 1$  then  
       $M_{clir}^{(t)} \leftarrow M_{gen}$  ▷ Start with general-domain SMT model for CLIR  
    else  
       $M_{clir}^{(t)} \leftarrow \theta M_{smt}^{(t-1)} + (1 - \theta) M_{smt}^{(t)}$  ▷ Use mixture of previous and current SMT model for CLIR  
    end if  
     $S_{in} \leftarrow \text{CLIR}(Q_{src}, D_{trg}, M_{clir}^{(t)})$  ▷ Retrieve top 1 target language Tweets for each source language query  
     $M_{smt}^{(t+1)} \leftarrow \text{TRAIN}(S_{gen} + S_{in})$  ▷ Train SMT model on general-domain and retrieved in-domain data  
     $t \leftarrow t + 1$   
  end while  
end procedure
```

---

|                  | BLEU (test) | # of in-domain sents |
|------------------|-------------|----------------------|
| Standard DA      | 14.05       | -                    |
| Full-scale CLIR  | 14.97       | 3,198,913            |
| Task alternation | 15.31       | ~40k                 |

Table 1: Standard Domain Adaptation with in-domain LM and tuning; Full-scale CLIR yielding over 3M in-domain parallel sentences; Task alternation ( $\theta = 0.1$ , iteration 7) using ~40k parallel sentences per iteration.

previous model with interpolation parameter  $\theta$ . Second, by always using  $P_{lex}(t|q)$  weights estimated from word alignments on  $S_{gen}$ .

We experimented with different ways of using the ranked retrieval results for each query and found that taking just the highest ranked document yielded the best results. This returns one pair of parallel Twitter messages per query, which are then used as additional training data for the SMT model in each iteration.

## 4 Experiments

### 4.1 Data

We trained the general domain model  $M_{gen}$  on data from the NIST evaluation campaign, including UN reports, newswire, broadcast news and blogs. Since we were interested in relative improvements rather than absolute performance, we sampled 1 million parallel sentences  $S_{gen}$  from the originally over 5.8 million parallel sentences.

We used a large corpus of Twitter messages, originally created by Jehl et al. (2012), as comparable in-domain data. Language identification was carried out with an off-the-shelf tool (Lui and Baldwin, 2012). We kept only Tweets classified as Arabic or English with over 95% confidence. After removing duplicates, we obtained 5.5 mil-

lion Arabic Tweets and 3.7 million English Tweets ( $D_{trg}$ ). Jehl et al. (2012) also supply a set of 1,022 Arabic Tweets with 3 English translations each for evaluation purposes, which was created by crowdsourcing translation on Amazon Mechanical Turk. We randomly split the parallel sentences into 511 sentences for development and 511 sentences for testing. All URLs and user names in Tweets were replaced by common placeholders. Hashtags were kept, since they might be helpful in the retrieval step. Since the evaluation data do not contain any hashtags, URLs or user names, we apply a post-processing step after decoding in which we remove those tokens.

### 4.2 Transductive Setup

Our method can be considered transductive in two ways. First, all Twitter data were collected by keyword-based crawling. Therefore, we can expect a topical similarity between development, test and training data. Second, since our setup aims for speed, we created a small set of queries  $Q_{src}$ , consisting of the source side of the evaluation data and similar Tweets. Similarity was defined by two criteria: First, we ranked all Arabic Tweets with respect to their term overlap with the development and test Tweets. Smoothed per-sentence BLEU (Lin and Och, 2004) was used as a similarity metric. OOV-coverage served as a second criterion to remedy the problem of unknown words in Twitter translation. We first created a general list of all OOVs in the evaluation data under  $M_{gen}$  (3,069 out of 7,641 types). For each of the top 100 BLEU-ranked Tweets, we counted OOV-coverage with respect to the corresponding source Tweet and the general OOV list. We only kept Tweets

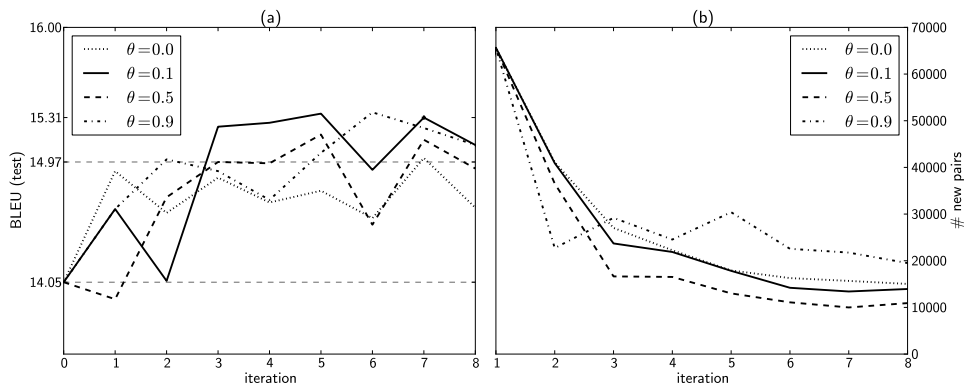


Figure 1: Learning curves for varying  $\theta$  parameters. (a) BLEU scores and (b) number of new pairs added per iteration.

containing at least one OOV term from the corresponding source Tweet and two OOV terms from the general list, resulting in 65,643 Arabic queries covering 86% of all OOVs. Our query set  $Q_{src}$  performed better (14.76 BLEU) after one iteration than a similar-sized set of random queries (13.39).

### 4.3 Experimental Results

We simulated the full-scale retrieval approach by Jehl et al. (2012) with the CLIR model described in section 3. It took 14 days to run 5.5M Arabic queries on 3.7M English documents. In contrast, our iterative approach completed a single iteration in less than 24 hours.<sup>1</sup>

In the absence of a Twitter data set for retrieval, we selected the parameters  $\lambda = 0.6$  (eq.1),  $L = 0.005$  and  $C = 0.95$  in a mate-finding task on Wikipedia data. The  $n$ -best list size for  $P_{nbest}(t|q)$  was 1000. All SMT models included a 5-gram language model built from the English side of the NIST data plus the English side of the Twitter corpus  $D_{trg}$ . Word alignments were created using GIZA++ (Och and Ney, 2003). Rule extraction and parameter tuning (MERT) was carried out with `cdéc`, using standard features. We ran MERT 5 times per iteration, carrying over the weights which achieved median performance on the development set to the next iteration.

Table 1 reports median BLEU scores on test of our standard adaptation baseline, the full-scale retrieval approach and the best result from our task alternation systems. Approximate randomization tests (Noreen, 1989; Riezler and Maxwell, 2005) showed that improvements of full-scale retrieval and task alternation over the baseline were statis-

<sup>1</sup>Retrieval was done in 4 batches on a Hadoop cluster using 190 mappers at once.

tically significant. Differences between full-scale retrieval and task alternation were not significant.<sup>2</sup>

Figure 1 illustrates the impact of  $\theta$ , which controls the importance of the previous model compared to the current one, on median BLEU (a) and change of  $S_{in}$  (b) over iterations. For all  $\theta$ , few iterations suffice to reach or surpass full-scale retrieval performance. Yet, no run achieved good performance after one iteration, showing that the transductive setup must be combined with task alternation to be effective. While we see fluctuations in BLEU for all  $\theta$ -values,  $\theta = 0.1$  achieves high scores faster and more consistently, pointing towards selecting a bolder updating strategy. This is also supported by plot (b), which indicates that choosing  $\theta = 0.1$  leads to faster stabilization in the pairs added per iteration ( $S_{in}$ ). We used this stabilization as a stopping criterion.

## 5 Conclusion

We presented a method that makes translation-based CLIR feasible for mining parallel sentences from large amounts of comparable data. The key of our approach is a translation-based high-quality retrieval model which gradually adapts to the target domain by iteratively re-training the underlying SMT model on a few thousand parallel sentences retrieved in the step before. The number of new pairs added per iteration stabilizes to a few thousand after 7 iterations, yielding an SMT model that improves 0.35 BLEU points over a model trained on millions of retrieved pairs.

<sup>2</sup>Note that our full-scale results are not directly comparable to those of Jehl et al. (2012) since our setup uses less than one fifth of the NIST data, a different decoder, a new CLIR approach, and a different development and test split.

## 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 Association for Computational Linguistics (EACL'09)*, Athens, Greece.
- Steven Abney. 2008. *Semisupervised Learning for Computational Linguistics*. Chapman and Hall.
- Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the 4th EACL Workshop on Statistical Machine Translation (WMT'09)*, Athens, Greece.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).
- Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations (ACL'10)*, Uppsala, Sweden.
- Laura Jehl, Felix Hieber, and Stefan Riezler. 2012. Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, Montreal, Quebec, Canada.
- Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.
- Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings the 20th International Conference on Computational Linguistics (COLING'04)*.
- 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, Demo Session (ACL'12)*, Jeju, Republic of Korea.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4).
- 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 (COLING-ACL'06)*, Sydney, Australia.
- Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley, New York.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1).
- Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI.
- Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1998. Okapi at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD.
- Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, Honolulu, Hawaii.
- Ferhan Ture and Jimmy Lin. 2012. Why not grab a free lunch? mining large corpora for parallel sentences to improve translation modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'12)*, Montreal, Canada.
- Ferhan Ture, Jimmy Lin, and Douglas W. Oard. 2012. Combining statistical translation techniques for cross-language information retrieval. In *Proceedings of the International Conference on Computational Linguistics (COLING'12)*, Mumbai, India.
- Ferhan Ture, Jimmy Lin, and Douglas W. Oard. 2012a. Looking inside the box: Context-sensitive translation for cross-language information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*, Portland, OR.
- Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, Beijing, China.
- Jinxi Xu, Ralph Weischedel, and Chanh Nguyen. 2001. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New York, NY.

# Sign Language Lexical Recognition With Propositional Dynamic Logic

**Arturo Curiel**

Université Paul Sabatier  
118 route de Narbonne, IRIT,  
31062, Toulouse, France  
curiel@irit.fr

**Christophe Collet**

Université Paul Sabatier  
118 route de Narbonne, IRIT,  
31062, Toulouse, France  
collet@irit.fr

## Abstract

This paper explores the use of Propositional Dynamic Logic (PDL) as a suitable formal framework for describing Sign Language (SL), the language of deaf people, in the context of natural language processing. SLs are visual, complete, standalone languages which are just as expressive as oral languages. Signs in SL usually correspond to sequences of highly specific body postures interleaved with movements, which make reference to real world objects, characters or situations. Here we propose a formal representation of SL signs, that will help us with the analysis of automatically-collected hand tracking data from French Sign Language (FSL) video corpora. We further show how such a representation could help us with the design of computer aided SL verification tools, which in turn would bring us closer to the development of an automatic recognition system for these languages.

## 1 Introduction

Sign languages (SL), the vernaculars of deaf people, are complete, rich, standalone communication systems which have evolved in parallel with oral languages (Valli and Lucas, 2000). However, in contrast to the last ones, research in automatic SL processing has not yet managed to build a complete, formal definition oriented to their automatic recognition (Cuxac and Dalle, 2007). In SL, both hands and non-manual features (NMF), *e.g.* facial muscles, can convey information with their placements, configurations and movements. These particular conditions can difficult the construction of

a formal description with common natural language processing (NLP) methods, since the existing modeling techniques are mostly designed to work with one-channel sound productions inherent to oral languages, rather than with the multi-channel partially-synchronized information induced by SLs.

Our research strives to address the formalization problem by introducing a logical language that lets us represent SL from the lowest level, so as to render the recognition task more approachable. For this, we use an instance of a formal logic, specifically Propositional Dynamic Logic (PDL), as a possible description language for SL signs.

For the rest of this section, we will present a brief introduction to current research efforts in the area. Section 2 presents a general description of our formalism, while section 3 shows how our work can be used when confronted with real world data. Finally, section 4 present our final observations and future work.

Images for the examples where taken from (DictaSign, 2012) corpus.

### 1.1 Current Sign Language Research

Extensive efforts have been made to achieve efficient automatic capture and representation of the subtle nuances commonly present in sign language discourse (Ong and Ranganath, 2005). Research ranges from the development of hand and body trackers (Dreuw et al., 2009; Gianni and Dalle, 2009), to the design of high level SL representation models (Lejeune, 2004; Lenseigne and Dalle, 2006). Linguistic research in the area has focused on the characterization of corporal expressions into meaningful transcriptions (Dreuw et al., 2010; Stokoe, 2005) or common patterns across SL (Aronoff et al., 2005; Meir et al., 2006; Wittmann, 1991), so as to gain understanding of the un-



derlying mechanisms of SL communication.

Works like (Losson and Vannobel, 1998) deal with the creation of a lexical description oriented to computer-based sign animation. Report (Filhol, 2009) describes a lexical specification to address the same problem. Both propose a thoroughly geometrical parametric encoding of signs, thus leaving behind meaningful information necessary for recognition and introducing data beyond the scope of recognition. This complicates the reutilization of their formal descriptions. Besides, they don't take in account the presence of partial information. Treating partiality is important for us, since it is often the case with automatic tools that incomplete or unrecognizable information arises. Finally, little to no work has been directed towards the unification of raw collected data from SL corpora with higher level descriptions (Dalle, 2006).

## 2 Propositional Dynamic Logic for SL

*Propositional Dynamic Logic* (PDL) is a multimodal logic, first defined by (Fischer and Ladner, 1979). It provides a language for describing programs, their correctness and termination, by allowing them to be modal operators. We work with our own variant of this logic, the *Propositional Dynamic Logic for Sign Language* (PDL<sub>SL</sub>), which is just an instantiation of PDL where we take signers' movements as programs.

Our sign formalization is based on the approach of (Liddell and Johnson, 1989) and (Filhol, 2008). They describe signs as sequences of immutable *key postures* and movement *transitions*.

In general, each key posture will be characterized by the concurrent parametric state of each *body articulator* over a time-interval. For us, a body articulator is any relevant body part involved in signing. The parameters taken in account can vary from articulator to articulator, but most of the time they comprise their configurations, orientations and their placement within one or more *places of articulation*. Transitions will correspond to the movements executed between fixed postures.

## 2.1 Syntax

We need to define some primitive sets that will limit the domain of our logical language.

**Definition 2.1 (Sign Language primitives).** *Let  $\mathcal{B}_{SL} = \{\mathbb{D}, \mathbb{W}, \mathbb{R}, \mathbb{L}\}$  be the set of relevant body articulators for SL, where  $\mathbb{D}$ ,  $\mathbb{W}$ ,  $\mathbb{R}$  and  $\mathbb{L}$  represent the dominant, weak, right and left hands, respectively. Both  $\mathbb{D}$  and  $\mathbb{W}$  can be aliases for the right or left hands, but they change depending on whether the signer is right-handed or left-handed, or even depending on the context.*

*Let  $\Psi$  be the two-dimensional projection of a human body skeleton, seen by the front. We define the set of places of articulation for SL as  $\Lambda_{SL} = \{\text{HEAD}, \text{CHEST}, \text{NEUTRAL}, \dots\}$ , such that for each  $\lambda \in \Lambda_{SL}$ ,  $\lambda$  is a sub-plane of  $\Psi$ , as shown graphically in figure 1.*

*Let  $\mathcal{C}_{SL}$  be the set of possible morphological configurations for a hand.*

*Let  $\Delta = \{\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow\}$  be the set of relative directions from the signer's point of view, where each arrow represents one of eight possible two-dimensional direction vectors that share the same origin. For vector  $\delta \in \Delta$ , we define vector  $\overleftarrow{\delta}$  as the same as  $\delta$  but with the inverted abscissa axis, such that  $\overleftarrow{\delta} \in \Delta$ . Let vector  $\widehat{\delta}$  indicate movement with respect to the dominant or weak hand in the following manner:*

$$\widehat{\delta} = \begin{cases} \delta & \text{if } \mathbb{D} \equiv \mathbb{R} \text{ or } \mathbb{W} \equiv \mathbb{L} \\ \overleftarrow{\delta} & \text{if } \mathbb{D} \equiv \mathbb{L} \text{ or } \mathbb{W} \equiv \mathbb{R} \end{cases}$$

*Finally, let  $\vec{v}_1$  and  $\vec{v}_2$  be any two vectors with the same origin. We denote the rotation angle between the two as  $\theta(\vec{v}_1, \vec{v}_2)$ .*

Now we define the set of atomic propositions that we will use to characterize fixed states, and a set of atomic actions to describe movements.

**Definition 2.2 (Atomic Propositions for SL Body Articulators  $\Phi_{SL}$ ).** *The set of atomic propositions for SL articulators ( $\Phi_{SL}$ ) is defined as:*

$$\Phi_{SL} = \{\beta_{1\beta_2}^\delta, \Xi_\lambda^{\beta_1}, \mathcal{T}_{\beta_2}^{\beta_1}, \mathcal{F}_c^{\beta_1}, \angle_{\beta_1}^\delta\}$$

*where  $\beta_1, \beta_2 \in \mathcal{B}_{SL}$ ,  $\delta \in \Delta$ ,  $\lambda \in \Lambda_{SL}$  and  $c \in \mathcal{C}_{SL}$ .*

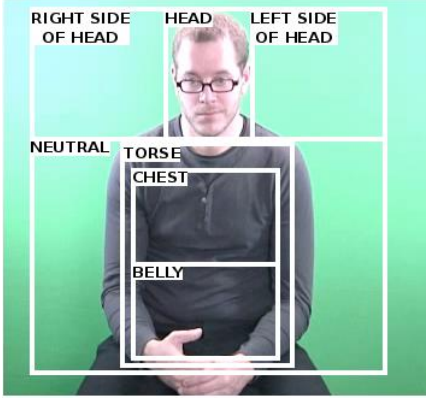


Figure 1: Possible places of articulation in  $\mathcal{B}_{\text{SL}}$ .

Intuitively,  $\beta_1^{\delta}_{\beta_2}$  indicates that articulator  $\beta_1$  is placed in relative direction  $\delta$  with respect to articulator  $\beta_2$ . Let the current place of articulation of  $\beta_2$  be the origin point of  $\beta_2$ 's Cartesian system ( $\mathcal{C}_{\beta_2}$ ). Let vector  $\vec{\beta}_1$  describe the current place of articulation of  $\beta_1$  in  $\mathcal{C}_{\beta_2}$ . Proposition  $\beta_1^{\delta}_{\beta_2}$  holds when  $\forall \vec{v} \in \Delta$ ,  $\theta(\vec{\beta}_1, \delta) \leq \theta(\vec{\beta}_1, \vec{v})$ .

$\Xi_{\lambda}^{\beta_1}$  asserts that articulator  $\beta_1$  is located in  $\lambda$ .

$\mathcal{T}_{\beta_2}^{\beta_1}$  is active whenever articulator  $\beta_1$  physically touches articulator  $\beta_2$ .

$\mathcal{F}_c^{\beta_1}$  indicates that  $c$  is the morphological configuration of articulator  $\beta_1$ .

Finally,  $\angle_{\beta_1}^{\delta}$  means that an articulator  $\beta_1$  is oriented towards direction  $\delta \in \Delta$ . For hands,  $\angle_{\beta_1}^{\delta}$  will hold whenever the vector perpendicular to the plane of the palm has the smallest rotation angle with respect to  $\delta$ .

**Definition 2.3 (Atomic Actions for SL Body Articulators  $\Pi_{\text{SL}}$ ).** The atomic actions for SL articulators ( $\Pi_{\text{SL}}$ ) are given by the following set:

$$\Pi_{\text{SL}} = \{\delta_{\beta_1}, \rightsquigarrow_{\beta_1}\}$$

where  $\delta \in \Delta$  and  $\beta_1 \in \mathcal{B}_{\text{SL}}$ .

Let  $\beta_1$ 's position before movement be the origin of  $\beta_1$ 's Cartesian system ( $\mathcal{C}_{\beta_1}$ ) and  $\vec{\beta}_1$  be the position vector of  $\beta_1$  in  $\mathcal{C}_{\beta_1}$  after moving. Action  $\delta_{\beta_1}$  indicates that  $\beta_1$  moves in relative direction  $\delta$  in  $\mathcal{C}_{\beta_1}$  if  $\forall \vec{v} \in \Delta$ ,  $\theta(\vec{\beta}_1, \delta) \leq \theta(\vec{\beta}_1, \vec{v})$ .

Action  $\rightsquigarrow_{\beta_1}$  occurs when articulator  $\beta_1$  moves rapidly and continuously (thrills) with-

out changing its current place of articulation.

**Definition 2.4 (Action Language for SL Body Articulators  $\mathcal{A}_{\text{SL}}$ ).** The action language for body articulators ( $\mathcal{A}_{\text{SL}}$ ) is given by the following rule:

$$\alpha ::= \pi \mid \alpha \cap \alpha \mid \alpha \cup \alpha \mid \alpha; \alpha \mid \alpha^*$$

where  $\pi \in \Pi_{\text{SL}}$ .

Intuitively,  $\alpha \cap \alpha$  indicates the concurrent execution of two actions, while  $\alpha \cup \alpha$  means that at least one of two actions will be non-deterministically executed. Action  $\alpha; \alpha$  describes the sequential execution of two actions. Finally, action  $\alpha^*$  indicates the reflexive transitive closure of  $\alpha$ .

**Definition 2.5 (Language  $\text{PDL}_{\text{SL}}$ ).** The formulae  $\varphi$  of  $\text{PDL}_{\text{SL}}$  are given by the following rule:

$$\varphi ::= \top \mid p \mid \neg \varphi \mid \varphi \wedge \varphi \mid [\alpha] \varphi$$

where  $p \in \Phi_{\text{SL}}$ ,  $\alpha \in \mathcal{A}_{\text{SL}}$ .

## 2.2 Semantics

$\text{PDL}_{\text{SL}}$  formulas are interpreted over labeled transition systems (LTS), in the spirit of the possible worlds model introduced by (Hintikka, 1962). Models correspond to connected graphs representing key postures and transitions: states are determined by the values of their propositions, while edges represent sets of executed movements. Here we present only a small extract of the logic semantics.

**Definition 2.6 (Sign Language Utterance Model  $\mathcal{U}_{\text{SL}}$ ).** A sign language utterance model ( $\mathcal{U}_{\text{SL}}$ ), is a tuple  $\mathcal{U}_{\text{SL}} = (S, R, \llbracket \cdot \rrbracket_{\Pi_{\text{SL}}}, \llbracket \cdot \rrbracket_{\Phi_{\text{SL}}})$  where:

- $S$  is a non-empty set of states
- $R$  is a transition relation  $R \subseteq S \times S$  where,  $\forall s \in S, \exists s' \in S$  such that  $(s, s') \in R$ .
- $\llbracket \cdot \rrbracket_{\Pi_{\text{SL}}} : \Pi_{\text{SL}} \rightarrow R$ , denotes the function mapping actions to the set of binary relations.
- $\llbracket \cdot \rrbracket_{\Phi_{\text{SL}}} : S \rightarrow 2^{\Phi_{\text{SL}}}$ , maps each state to a set of atomic propositions.

We also need to define a structure over sequences of states to model internal dependencies between them, nevertheless we decided to omit the rest of our semantics, alongside satisfaction conditions, for the sake of readability.

### 3 Use Case: Semi-Automatic Sign Recognition

We now present an example of how we can use our formalism in a semi-automatic sign recognition system. Figure 2 shows a simple module diagram exemplifying information flow in the system’s architecture. We proceed to briefly describe each of our modules and how they work together.

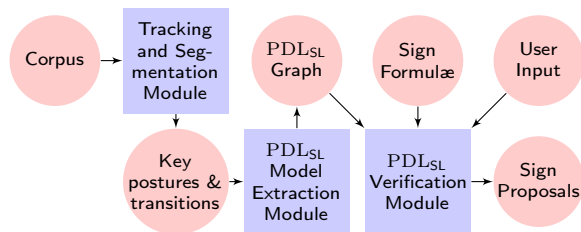


Figure 2: Information flow in a semi-automatic SL lexical recognition system.

#### 3.1 Tracking and Segmentation Module

The process starts by capturing relevant information from video corpora. We use an existing head and hand tracker expressly developed for SL research (Gonzalez and Collet, 2011). This tool analyses individual video instances, and returns the frame-by-frame positions of the tracked articulators. By using this information, the module can immediately calculate speeds and directions on the fly for each hand.

The module further employs the method proposed by the authors in (Gonzalez and Collet, 2012) to achieve sub-lexical segmentation from the previously calculated data. Like them, we use the relative velocity between hands to identify when hands either move at the same time, independently or don’t move at all. With these, we can produce a set of possible key postures and transitions that will serve as input to the modeling module.

#### 3.2 Model Extraction Module

This module calculates a propositional state for each static posture, where atomic PDL<sub>SL</sub>

formulas codify the information tracked in the previous part. Detected movements are interpreted as PDL<sub>SL</sub> actions between states.

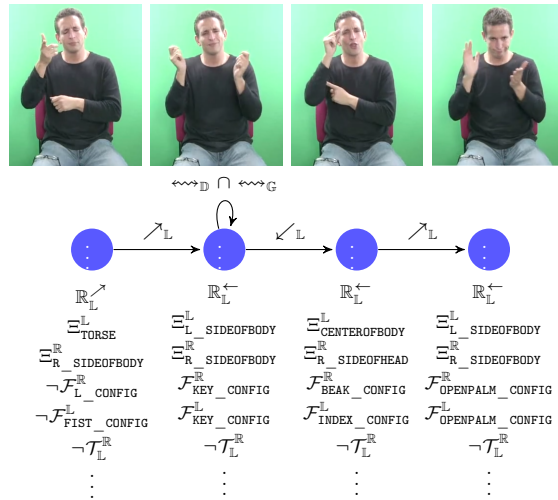


Figure 3: Example of modeling over four automatically identified frames as possible key postures.

Figure 3 shows an example of the process. Here, each key posture is codified into propositions acknowledging the hand positions with respect to each other ( $\mathbb{R}_L^{\leftarrow}$ ), their place of articulation (e.g. “left hand floats over the torse” with  $\Xi_{\text{TORSE}}^L$ ), their configuration (e.g. “right hand is open” with  $\mathcal{F}_{\text{OPENPALM\_CONFIG}}^R$ ) and their movements (e.g. “left hand moves to the up-left direction” with  $\nearrow_L$ ).

This module also checks that the generated graph is correct: it will discard simple tracking errors to ensure that the resulting LTS will remain consistent.

#### 3.3 Verification Module

First of all, the verification module has to be loaded with a database of sign descriptions encoded as PDL<sub>SL</sub> formulas. These will characterize the specific sequence of key postures that morphologically describe a sign. For example, let’s take the case for sign “route” in FSL, shown in figure 4, with the following PDL<sub>SL</sub> formulation,

**Example 3.1 (ROUTE<sub>FSL</sub> formula).**

$$\begin{aligned}
 & (\Xi_{\text{FACE}}^R \wedge \Xi_{\text{FACE}}^L \wedge \mathbb{L}_R^{\rightarrow} \wedge \mathcal{F}_{\text{CLAMP}}^R \wedge \mathcal{F}_{\text{CLAMP}}^L \wedge \mathcal{T}_L^R) \rightarrow \\
 & \quad [\leftarrow_R \cap \rightarrow_L](\mathbb{L}_R^{\rightarrow} \wedge \mathcal{F}_{\text{CLAMP}}^R \wedge \mathcal{F}_{\text{CLAMP}}^L \wedge \neg \mathcal{T}_L^R)
 \end{aligned} \tag{1}$$

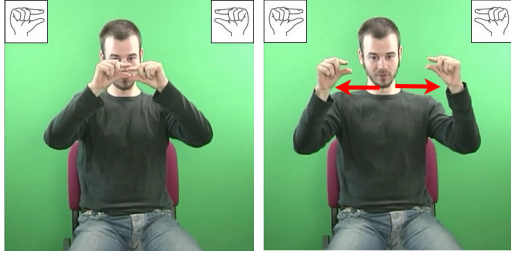


Figure 4:  $ROUTE_{FSL}$  production.

Formula (1) describes  $ROUTE_{FSL}$  as a sign with two key postures, connected by a two-hand simultaneous movement (represented with operator  $\cap$ ). It also indicates the position of each hand, their orientation, whether they touch and their respective configurations (in this example, both hold the same  $CLAMP$  configuration).

The module can then verify whether a sign formula in the lexical database holds in any sub-sequence of states of the graph generated in the previous step. Algorithm 1 sums up the process.

---

**Algorithm 1**  $PDL_{SL}$  Verification Algorithm

---

**Require:** SL model  $\mathcal{M}_{SL}$

**Require:** connected graph  $\mathcal{G}_{SL}$

**Require:** lexical database  $\mathcal{DB}_{SL}$

- 1:  $Proposals\_For[state\_qty]$
  - 2: **for** state  $s \in \mathcal{G}_{SL}$  **do**
  - 3:     **for** sign  $\varphi \in \mathcal{DB}_{SL}$  where  $s \in \varphi$  **do**
  - 4:         **if**  $\mathcal{M}_{SL}, s \models \varphi$  **then**
  - 5:              $Proposals\_For[s].append(\varphi)$
  - 6:         **end if**
  - 7:     **end for**
  - 8: **end for**
  - 9: **return**  $Proposals\_For$
- 

For each state, the algorithm returns a set of possible signs. Expert users (or higher level algorithms) can further refine the process by introducing additional information previously missed by the tracker.

## 4 Conclusions and Future Work

We have shown how a logical language can be used to model SL signs for semi-automatic recognition, albeit with some restrictions. The traits we have chosen to represent were imposed by the limits of the tracking tools we had to our disposition, most notably working

with 2D coordinates. With these in mind, we tried to design something flexible that could be easily adapted by computer scientists and linguists alike. Our primitive sets, were intentionally defined in a very general fashion due to the same reason: all of the perceived directions, articulators and places of articulation can easily change their domains, depending on the SL we are modeling or the technological constraints we have to deal with. Propositions can also be changed, or even induced, by existing written sign representation languages such as Zebedee (Filhol, 2008) or HamNoSys (Hanke, 2004), mainly for the sake of extendability.

From the application side, we still need to create an extensive sign database codified in  $PDL_{SL}$  and try recognition on other corpora, with different tracking information. For verification and model extraction, further optimizations are expected, including the handling of data inconsistencies and repairing broken queries when verifying the graph.

Regarding our theoretical issues, future work will be centered in improving our language to better comply with SL research. This includes adding new features, like incorporating probability representation to improve recognition. We also expect to finish the definition of our formal semantics, as well as proving correction and complexity of our algorithms.

## References

- Mark Aronoff, Irit Meir, and Wendy Sandler. 2005. The paradox of sign language morphology. *Language*, 81(2):301.
- Christian Cuxac and Patrice Dalle. 2007. *Problématique des chercheurs en traitement automatique des langues des signes*, volume 48 of *Traitement Automatique des Langues*. Lavoisier, <http://www.editions-hermes.fr/>, October.
- Patrice Dalle. 2006. High level models for sign language analysis by a vision system. In *Workshop on the Representation and Processing of Sign Language: Lexicographic Matters and Didactic Scenarios (LREC), Italy, ELDA*, page 17–20.
- DictaSign. 2012. <http://www.dictasign.eu>.
- Philippe Dreuw, Daniel Stein, and Hermann Ney. 2009. Enhancing a sign language translation system with vision-based features. In Miguel Sales Dias, Sylvie Gibet, Marcelo M.

- Wanderley, and Rafael Bastos, editors, *Gesture-Based Human-Computer Interaction and Simulation*, number 5085 in Lecture Notes in Computer Science, pages 108–113. Springer Berlin Heidelberg, January.
- Philippe Dreuw, Hermann Ney, Gregorio Martinez, Onno Crasborn, Justus Piater, Jose Miguel Moya, and Mark Wheatley. 2010. The Sign-Speak project - bridging the gap between signers and speakers. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, and *et. al.*, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).
- Michael Filhol. 2008. *Modèle descriptif des signes pour un traitement automatique des langues des signes*. Ph.D. thesis, Université Paris-sud (Paris 11).
- Michael Filhol. 2009. Zebedee: a lexical description model for sign language synthesis. Internal, LIMSI.
- Michael J. Fischer and Richard E. Ladner. 1979. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, April.
- Frédéric Gianni and Patrice Dalle. 2009. Robust tracking for processing of videos of communication's gestures. *Gesture-Based Human-Computer Interaction and Simulation*, page 93–101.
- Matilde Gonzalez and Christophe Collet. 2011. Robust body parts tracking using particle filter and dynamic template. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 529–532, September.
- Matilde Gonzalez and Christophe Collet. 2012. Sign segmentation using dynamics and hand configuration for semi-automatic annotation of sign language corpora. In Eleni Efthimiou, Georgios Kouroupetroglou, and Stavroula-Evita Fotinea, editors, *Gesture and Sign Language in Human-Computer Interaction and Embodied Communication*, number 7206 in Lecture Notes in Computer Science, pages 204–215. Springer Berlin Heidelberg, January.
- Thomas Hanke. 2004. HamNoSys—Representing sign language data in language resources and language processing contexts. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages “From Sign Writing to Image Processing. Information*, Lisbon, Portugal, 30 May.
- Jaakko Hintikka. 1962. *Knowledge and Belief*. Ithaca, N.Y., Cornell University Press.
- Fanch Lejeune. 2004. *Analyse sémantico-cognitive d'énoncés en Langue des Signes Française pour une génération automatique de séquences gestuelles*. Ph.D. thesis, PhD thesis, Orsay University, France.
- Boris Lenseigne and Patrice Dalle. 2006. Using signing space as a representation for sign language processing. In Sylvie Gibet, Nicolas Courty, and Jean-François Kamp, editors, *Gesture in Human-Computer Interaction and Simulation*, number 3881 in Lecture Notes in Computer Science, pages 25–36. Springer Berlin Heidelberg, January.
- S. K. Liddell and R. E. Johnson. 1989. *American sign language: The phonological base*. Gallaudet University Press, Washington. DC.
- Olivier Losson and Jean-Marc Vannobel. 1998. Sign language formal description and synthesis. *INT.JOURNAL OF VIRTUAL REALITY*, 3:27–34.
- Irit Meir, Carol Padden, Mark Aronoff, and Wendy Sandler. 2006. Re-thinking sign language verb classes: the body as subject. In *Sign Languages: Spinning and Unraveling the Past, Present and Future. 9th Theoretical Issues in Sign Language Research Conference, Florianopolis, Brazil*, volume 382.
- Sylvie C. W. Ong and Surendra Ranganath. 2005. Automatic sign language analysis: a survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):873–891, June.
- William C. Stokoe. 2005. Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of Deaf Studies and Deaf Education*, 10(1):3–37, January.
- Clayton Valli and Ceil Lucas. 2000. *Linguistics of American Sign Language Text, 3rd Edition: An Introduction*. Gallaudet University Press.
- Henri Wittmann. 1991. Classification linguistique des langues signées non vocalement. *Revue québécoise de linguistique théorique et appliquée*, 10(1):88.

# Stacking for Statistical Machine Translation\*

Majid Razmara and Anoop Sarkar

School of Computing Science

Simon Fraser University

Burnaby, BC, Canada

{razmara,anoop}@sfu.ca

## Abstract

We propose the use of *stacking*, an ensemble learning technique, to the statistical machine translation (SMT) models. A diverse ensemble of weak learners is created using the same SMT engine (a hierarchical phrase-based system) by manipulating the training data and a strong model is created by combining the weak models on-the-fly. Experimental results on two language pairs and three different sizes of training data show significant improvements of up to 4 BLEU points over a conventionally trained SMT model.

## 1 Introduction

Ensemble-based methods have been widely used in machine learning with the aim of reducing the instability of classifiers and regressors and/or increase their bias. The idea behind ensemble learning is to combine multiple models, *weak learners*, in an attempt to produce a *strong model* with less error. It has also been successfully applied to a wide variety of tasks in NLP (Tomeh et al., 2010; Surdeanu and Manning, 2010; F. T. Martins et al., 2008; Sang, 2002) and recently has attracted attention in the statistical machine translation community in various work (Xiao et al., 2013; Song et al., 2011; Xiao et al., 2010; Lagarda and Casacuberta, 2008).

In this paper, we propose a method to adopt *stacking* (Wolpert, 1992), an ensemble learning technique, to SMT. We manipulate the full set of training data, creating  $k$  disjoint sets of *held-out* and *held-in* data sets as in  $k$ -fold cross-validation and build a model on each partition. This creates a diverse ensemble of statistical machine translation models where each member of the ensemble has different feature function values for the SMT log-linear model (Koehn, 2010). The weights of model are then tuned using minimum error rate training (Och, 2003) on the *held-out* fold to provide  $k$  weak models. We then create a strong

model by stacking another meta-learner on top of weak models to combine them into a single model. The particular second-tier model we use is a model combination approach called *ensemble decoding* which combines hypotheses from the weak models on-the-fly in the decoder.

Using this approach, we take advantage of the diversity created by manipulating the training data and obtain a significant and consistent improvement over a conventionally trained SMT model with a fixed training and tuning set.

## 2 Ensemble Learning Methods

Two well-known instances of general framework of ensemble learning are *bagging* and *boosting*. Bagging (Breiman, 1996a) (bootstrap aggregating) takes a number of samples with replacement from a training set. The generated sample set may have 0, 1 or more instances of each original training instance. This procedure is repeated a number of times and the base learner is applied to each sample to produce a weak learner. These models are aggregated by doing a uniform voting for classification or averaging the predictions for regression. Bagging reduces the variance of the base model while leaving the bias relatively unchanged and is most useful when a small change in the training data affects the prediction of the model (i.e. the model is unstable) (Breiman, 1996a). Bagging has been recently applied to SMT (Xiao et al., 2013; Song et al., 2011)

*Boosting* (Schapire, 1990) constructs a strong learner by repeatedly choosing a weak learner and applying it on a re-weighted training set. In each iteration, a weak model is learned on the training data, whose instance weights are modified from the previous iteration to concentrate on examples on which the model predictions were poor. By putting more weight on the wrongly predicted examples, a diverse ensemble of weak learners is created. Boosting has also been used in SMT (Xiao et al., 2013; Xiao et al., 2010; Lagarda

\*This research was partially supported by an NSERC, Canada (RGPIN: 264905) grant and a Google Faculty Award to the second author.

---

Algorithm 1: Stacking for SMT

---

**Input:**  $\mathcal{D} = \{\langle f_j, e_j \rangle\}_{j=1}^N$   $\triangleright$  A parallel corpus  
**Input:**  $k$   $\triangleright$  # of folds (i.e. weak learners)  
**Output:** STRONGMODEL  $s$   
1:  $\mathcal{D}^1, \dots, \mathcal{D}^k \leftarrow \text{SPLIT}(\mathcal{D}, k)$   
2: **for**  $i = 1 \rightarrow k$  **do**  
3:  $\mathcal{T}^i \leftarrow \mathcal{D} - \mathcal{D}^i$   $\triangleright$  Use all but current partition as training set.  
4:  $\phi_i \leftarrow \text{TRAIN}(\mathcal{T}^i)$   $\triangleright$  Train feature functions.  
5:  $\mathcal{M}_i \leftarrow \text{TUNE}(\phi_i, \mathcal{D}^i)$   $\triangleright$  Tune the model on the current partition.  
6: **end for**  
7:  $s \leftarrow \text{COMBINEMODELS}(\mathcal{M}_1, \dots, \mathcal{M}_k)$   $\triangleright$  Combine all the base models to produce a strong stacked model.

---

and Casacuberta, 2008).

Stacking (or stacked generalization) (Wolpert, 1992) is another ensemble learning algorithm that uses a second-level learning algorithm on top of the base learners to reduce the bias. The first level consists of predictors  $g_1, \dots, g_k$  where  $g_i : \mathbb{R}^d \rightarrow \mathbb{R}$ , receiving input  $x \in \mathbb{R}^d$  and producing a prediction  $g_i(x)$ . The next level consists of a single function  $h : \mathbb{R}^{d+k} \rightarrow \mathbb{R}$  that takes  $\langle x, g_1(x), \dots, g_k(x) \rangle$  as input and produces an ensemble prediction  $\hat{y} = h(x, g_1(x), \dots, g_k(x))$ .

Two categories of ensemble learning are *homogeneous learning* and *heterogeneous learning*. In homogeneous learning, a single base learner is used, and diversity is generated by data sampling, feature sampling, randomization and parameter settings, among other strategies. In heterogeneous learning different learning algorithms are applied to the same training data to create a pool of diverse models. In this paper, we focus on homogeneous ensemble learning by manipulating the training data.

In the primary form of stacking (Wolpert, 1992), the training data is split into multiple disjoint sets of *held-out* and *held-in* data sets using  $k$ -fold cross-validation and  $k$  models are trained on the held-in partitions and run on held-out partitions. Then a meta-learner uses the predictions of all models on their held-out sets and the actual labels to learn a final model. The details of the first-layer and second-layer predictors are considered to be a “black art” (Wolpert, 1992).

Breiman (1996b) linearly combines the weak learners in the stacking framework. The weights of the base learners are learned using ridge regression:  $s(x) = \sum_k \alpha_k m_k(x)$ , where  $m_k$  is a base model trained on the  $k$ -th partition of the data and  $s$  is the resulting strong model created by linearly interpolating the weak learners.

Stacking (aka blending) has been used in the system that won the Netflix Prize<sup>1</sup>, which used a multi-level stacking algorithm.

Stacking has been actively used in statistical parsing: Nivre and McDonald (2008) integrated two models for dependency parsing by letting one model learn from features generated by the other; F. T. Martins et al. (2008) further formalized the stacking algorithm and improved on Nivre and McDonald (2008); Surdeanu and Manning (2010) includes a detailed analysis of ensemble models for statistical parsing: *i*) the diversity of base parsers is more important than the complexity of the models; *ii*) unweighted voting performs as well as weighted voting; and *iii*) ensemble models that combine at decoding time significantly outperform models that combine multiple models at training time.

### 3 Our Approach

In this paper, we propose a method to apply stacking to statistical machine translation (SMT) and our method is the first to successfully exploit stacking for statistical machine translation. We use a standard statistical machine translation engine and produce multiple diverse models by partitioning the training set using the  $k$ -fold cross-validation technique. A diverse ensemble of weak systems is created by learning a model on each  $k - 1$  fold and tuning the statistical machine translation log-linear weights on the remaining fold. However, instead of learning a model on the output of base models as in (Wolpert, 1992), we combine hypotheses from the base models in the decoder with uniform weights. For the base learner, we use Kriya (Sankaran et al., 2012), an in-house hierarchical phrase-based machine translation system, to produce multiple weak models. These models are combined together using *Ensemble Decoding* (Razmara et al., 2012) to produce a strong model in the decoder. This method is briefly explained in next section.

#### 3.1 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  $\phi$ :

$$p(e|f) \propto \exp(\mathbf{w} \cdot \phi)$$

---

<sup>1</sup><http://www.netflixprize.com/>

Ensemble decoding combines several models dynamically at decoding time. The scores are combined for each partial hypothesis using a user-defined mixture operation  $\otimes$  over component models.

$$p(e|f) \propto \exp(\mathbf{w}_1 \cdot \phi_1 \otimes \mathbf{w}_2 \cdot \phi_2 \otimes \dots)$$

We previously 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 (Razmara et al., 2012). 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  $\bar{e}, \bar{f}$  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:

$$p(\bar{e}|\bar{f}) \propto \max_m (\lambda_m \exp(\mathbf{w}_m \cdot \phi_m))$$

- **Prod or log-wsum** is defined as:

$$p(\bar{e}|\bar{f}) \propto \exp\left(\sum_m^M \lambda_m (\mathbf{w}_m \cdot \phi_m)\right)$$

- **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 as 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,  $\psi(\bar{f}, n)$ , could be based on max

|                | Train size | Src tokens | Tgt tokens |
|----------------|------------|------------|------------|
| <b>Fr - En</b> | 0+dev      | 67K        | 58K        |
|                | 10k+dev    | 365K       | 327K       |
|                | 100k+dev   | 3M         | 2.8M       |
| <b>Es - En</b> | 0+dev      | 60K        | 58K        |
|                | 10k+dev    | 341K       | 326K       |
|                | 100k+dev   | 2.9M       | 2.8M       |

Table 1: Statistics of the training set for different systems and different language pairs.

(SW:MAX), i.e. for each cell, the model that has the highest weighted score wins:

$$\psi(\bar{f}, n) = \lambda_n \max_{\bar{e}} (\mathbf{w}_n \cdot \phi_n(\bar{e}, \bar{f}))$$

Alternatively, we can pick the model with highest weighted sum of the probabilities of the rules (SW:SUM). This sum has to take into account the translation table limit (*ttl*), on the number of rules suggested by each model for each cell:

$$\psi(\bar{f}, n) = \lambda_n \sum_{\bar{e}} \exp(\mathbf{w}_n \cdot \phi_n(\bar{e}, \bar{f}))$$

The probability of each phrase-pair  $(\bar{e}, \bar{f})$  is then:

$$p(\bar{e}|\bar{f}) = \sum_m^M \delta(\bar{f}, m) p_m(\bar{e}|\bar{f})$$

## 4 Experiments & Results

We experimented with two language pairs: French to English and Spanish to English on the *Europarl* corpus (v7) (Koehn, 2005) and used ACL/WMT 2005<sup>2</sup> data for dev and test sets.

For the base models, we used an in-house implementation of hierarchical phrase-based systems, Kriya (Sankaran et al., 2012), which uses the same features mentioned in (Chiang, 2005): forward and backward relative-frequency and lexical TM probabilities; LM; word, phrase and glue-rules penalty. GIZA++ (Och and Ney, 2003) has been used for word alignment with phrase length limit of 10. Feature weights were optimized using MERT (Och, 2003). We built a 5-gram language model on the English side of *Europarl* and used the Kneser-Ney smoothing method and SRILM (Stolcke, 2002) as the language model toolkit.

<sup>2</sup><http://www.statmt.org/wpt05/mt-shared-task/>



| Direction | k-fold | Resub | Mean  | WSUM         | WMAX         | PROD         | SW:MAX | SW:SUM       |
|-----------|--------|-------|-------|--------------|--------------|--------------|--------|--------------|
| Fr - En   | 2      | 18.08 | 19.67 | 22.32        | <b>22.48</b> | 22.06        | 21.70  | 21.81        |
|           | 4      | 18.08 | 21.80 | 23.14        | 23.48        | <b>23.55</b> | 22.83  | 22.95        |
|           | 8      | 18.08 | 22.47 | 23.76        | 23.75        | <b>23.78</b> | 23.02  | 23.47        |
| Es - En   | 2      | 18.61 | 19.23 | <b>21.62</b> | 21.33        | 21.49        | 21.48  | 21.51        |
|           | 4      | 18.61 | 21.52 | 23.42        | 22.81        | <b>22.91</b> | 22.81  | <b>22.92</b> |
|           | 8      | 18.61 | 22.20 | 23.69        | <b>23.89</b> | 23.51        | 22.92  | 23.26        |

Table 2: Testset BLEU scores when applying stacking on the devset only (using no specific training set).

| Direction | Corpus   | k-fold  | Baseline | BMA   | WSUM         | WMAX         | PROD  | SW:MAX | SW:SUM       |
|-----------|----------|---------|----------|-------|--------------|--------------|-------|--------|--------------|
| Fr - En   | 10k+dev  | 6       | 28.75    | 29.49 | <b>29.87</b> | 29.78        | 29.21 | 29.69  | 29.59        |
|           | 100k+dev | 11 / 51 | 29.53    | 29.75 | 34.00        | <b>34.07</b> | 33.11 | 34.05  | 33.96        |
| Es - En   | 10k+dev  | 6       | 28.21    | 28.76 | <b>29.59</b> | 29.51        | 29.15 | 29.10  | 29.21        |
|           | 100k+dev | 11 / 51 | 33.25    | 33.44 | <b>34.21</b> | 34.00        | 33.17 | 34.19  | <b>34.22</b> |

Table 3: Testset BLEU scores when using 10k and 100k sentence training sets along with the devset.

#### 4.1 Training on devset

We first consider the scenario in which there is no parallel data between a language pair except a small bi-text used as a devset. We use no specific training data and construct a SMT system completely on the devset by using our approach and compare to two different baselines. A natural baseline when having a limited parallel text is to do re-substitution validation where the model is trained on the whole devset and is tuned on the same set. This validation process suffers seriously from over-fitting. The second baseline is the mean of BLEU scores of all base models.

Table 2 summarizes the BLEU scores on the testset when using stacking only on the devset on two different language pairs. As the table shows, increasing the number of folds results in higher BLEU scores. However, doing such will generally lead to higher variance among base learners.

Figure 1 shows the BLEU score of each of the base models resulted from a 20-fold partitioning of the devset along with the strong models’ BLEU scores. As the figure shows, the strong models are generally superior to the base models whose mean is represented as a horizontal line.

#### 4.2 Training on train+dev

When we have some training data, we can use the cross-validation-style partitioning to create  $k$  splits. We then train a system on  $k - 1$  folds and tune on the devset. However, each system eventually wastes a fold of the training data. In order to take advantage of that remaining fold, we concatenate the devset to the training set and partition the whole union. In this way, we use all data available to us. We experimented with two sizes of train-

ing data: 10k sentence pairs and 100k, that with the addition of the devset, we have 12k and 102k sentence-pair corpora.

Table 1 summarizes statistics of the data sets used in this scenario. Table 3 reports the BLEU scores when using stacking on these two corpus sizes. The baselines are the conventional systems which are built on the training-set only and tuned on the devset as well as *Bayesian Model Averaging* (BMA, see §5). For the 100k+dev corpus, we sampled 11 partitions from all 51 possible partitions by taking every fifth partition as training data. The results in Table 3 show that stacking can improve over the baseline BLEU scores by up to 4 points.

Examining the performance of the different mixture operations, we can see that WSUM and WMAX typically outperform other mixture operations. Different mixture operations can be dominant in different language pairs and different sizes of training sets.

### 5 Related Work

Xiao et al. (2013) have applied both boosting and bagging on three different statistical machine translation engines: phrase-based (Koehn et al., 2003), hierarchical phrase-based (Chiang, 2005) and syntax-based (Galley et al., 2006) and showed SMT can benefit from these methods as well.

Duan et al. (2009) creates an ensemble of models by using feature subspace method in the machine learning literature (Ho, 1998). Each member of the ensemble is built by removing one non-LM feature in the log-linear framework or varying the order of language model. Finally they use a sentence-level system combination on the outputs of the base models to pick the best system for each

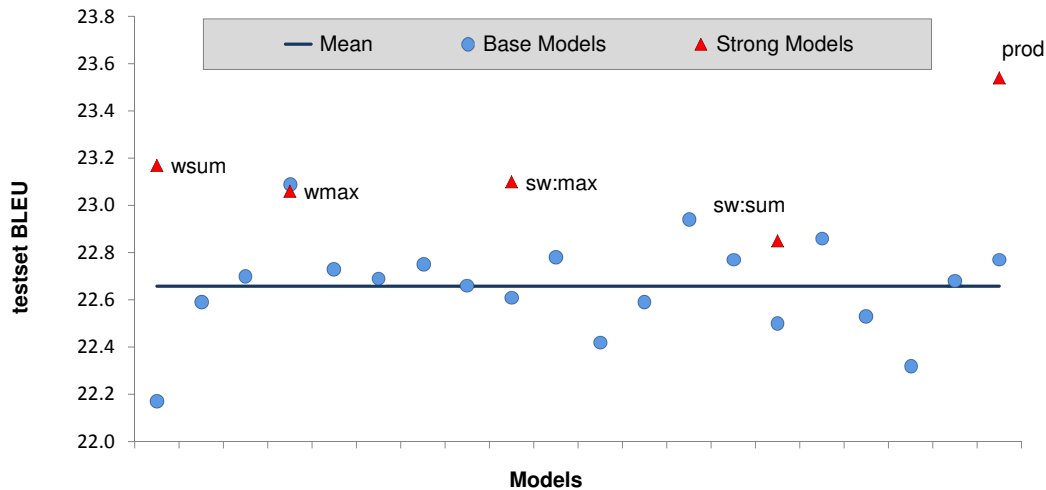


Figure 1: BLEU scores for all the base models and stacked models on the Fr-En devset with 20-fold cross validation. The horizontal line shows the mean of base models’ scores.

sentence. Though, they do not combine the hypotheses search spaces of individual base models.

Our work is most similar to that of Duan et al. (2010) which uses *Bayesian model averaging* (BMA) (Hoeting et al., 1999) for SMT. They used sampling without replacement to create a number of base models whose phrase-tables are combined with that of the baseline (trained on the full training-set) using linear mixture models (Foster and Kuhn, 2007).

Our approach differs from this approach in a number of ways: *i)* we use cross-validation-style partitioning for creating training subsets while they do sampling without replacement (80% of the training set); *ii)* in our approach a number of base models are trained and tuned and they are combined on-the-fly in the decoder using *ensemble decoding* which has been shown to be more effective than offline combination of phrase-table-only features; *iii)* in Duan et al. (2010)’s method, each system gives up 20% of the training data in exchange for more diversity, but in contrast, our method not only uses all available data for training, but promotes diversity through allowing each model to tune on a different data set; *iv)* our approach takes advantage of held out data (the tuning set) in the training of base models which is beneficial especially when little parallel data is available or tuning/test sets and training sets are from different domains.

Empirical results (Table 3) also show that our approach outperforms the Bayesian model averaging approach (BMA).

## 6 Conclusion & Future Work

In this paper, we proposed a novel method on applying stacking to the statistical machine translation task. The results when using no, 10k and 100k sentence-pair training sets (along with a development set for tuning) show that stacking can yield an improvement of up to 4 BLEU points over conventionally trained SMT models which use a fixed training and tuning set.

Future work includes experimenting with larger training sets to investigate how useful this approach can be when having different sizes of training data.

## References

- Leo Breiman. 1996a. Bagging predictors. *Machine Learning*, 24(2):123–140, August.
- Leo Breiman. 1996b. Stacked regressions. *Machine Learning*, 24(1):49–64, July.
- 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.
- Nan Duan, Mu Li, Tong Xiao, and Ming Zhou. 2009. The feature subspace method for smt system combination. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP ’09, pages 1096–1104, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Nan Duan, Hong Sun, and Ming Zhou. 2010. Translation model generalization using probability averaging for machine translation. In *Proceedings of the*

- 23rd International Conference on Computational Linguistics, COLING '10, pages 304–312, Stroudsburg, PA, USA. Association for Computational Linguistics.
- André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Honolulu, Hawaii, October. Association for Computational Linguistics.
- George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation, StatMT '07*, pages 128–135, Stroudsburg, PA, USA. ACL.
- 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, ACL-44*, pages 961–968, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, August.
- Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. 1999. Bayesian Model Averaging: A Tutorial. *Statistical Science*, 14(4):382–401.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 127–133, Edmonton, May. NAACL.
- 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.
- Antonio Lagarda and Francisco Casacuberta. 2008. Applying boosting to statistical machine translation. In *Annual Meeting of European Association for Machine Translation (EAMT)*, pages 88–96.
- 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.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.
- 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.
- Erik F. Tjong Kim Sang. 2002. Memory-based shallow parsing. *J. Mach. Learn. Res.*, 2:559–594, March.
- 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.
- Robert E. Schapire. 1990. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, July.
- Linfeng Song, Haitao Mi, Yajuan Lü, and Qun Liu. 2011. Bagging-based system combination for domain adaptation. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 293–299. International Association for Machine Translation, September.
- Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286.
- Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble models for dependency parsing: cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 649–652, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Nadi Tomeh, Alexandre Allauzen, Guillaume Wisniewski, and François Yvon. 2010. Refining word alignment with discriminative training. In *Proceedings of The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.
- David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.
- 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 '10*, pages 739–748, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tong Xiao, Jingbo Zhu, and Tongran Liu. 2013. Bagging and boosting statistical machine translation systems. *Artificial Intelligence*, 195:496–527, February.

# Bilingual Data Cleaning for SMT using Graph-based Random Walk\*

Lei Cui<sup>†</sup>, Dongdong Zhang<sup>‡</sup>, Shujie Liu<sup>‡</sup>, Mu Li<sup>‡</sup>, and Ming Zhou<sup>‡</sup>

<sup>†</sup>School of Computer Science and Technology  
Harbin Institute of Technology, Harbin, China  
leicui@hit.edu.cn

<sup>‡</sup>Microsoft Research Asia, Beijing, China  
{dozhang, shujliu, muli, mingzhou}@microsoft.com

## Abstract

The quality of bilingual data is a key factor in Statistical Machine Translation (SMT). Low-quality bilingual data tends to produce incorrect translation knowledge and also degrades translation modeling performance. Previous work often used supervised learning methods to filter low-quality data, but a fair amount of human labeled examples are needed which are not easy to obtain. To reduce the reliance on labeled examples, we propose an unsupervised method to clean bilingual data. The method leverages the mutual reinforcement between the sentence pairs and the extracted phrase pairs, based on the observation that better sentence pairs often lead to better phrase extraction and vice versa. End-to-end experiments show that the proposed method substantially improves the performance in large-scale Chinese-to-English translation tasks.

## 1 Introduction

Statistical machine translation (SMT) depends on the amount of bilingual data and its quality. In real-world SMT systems, bilingual data is often mined from the web where low-quality data is inevitable. The low-quality bilingual data degrades the quality of word alignment and leads to the incorrect phrase pairs, which will hurt the translation performance of phrase-based SMT systems (Koehn et al., 2003; Och and Ney, 2004). Therefore, it is very important to exploit data quality information to improve the translation modeling.

Previous work on bilingual data cleaning often involves some supervised learning methods. Several bilingual data mining systems (Resnik and

Smith, 2003; Shi et al., 2006; Munteanu and Marcu, 2005; Jiang et al., 2009) have a post-processing step for data cleaning. Maximum entropy or SVM based classifiers are built to filter some non-parallel data or partial-parallel data. Although these methods can filter some low-quality bilingual data, they need sufficient human labeled training instances to build the model, which may not be easy to acquire.

To this end, we propose an unsupervised approach to clean the bilingual data. It is intuitive that high-quality parallel data tends to produce better phrase pairs than low-quality data. Meanwhile, it is also observed that the phrase pairs that appear frequently in the bilingual corpus are more reliable than less frequent ones because they are more reusable, hence most good sentence pairs are prone to contain more frequent phrase pairs (Foster et al., 2006; Wuebker et al., 2010). This kind of mutual reinforcement fits well into the framework of graph-based random walk. When a phrase pair  $p$  is extracted from a sentence pair  $s$ ,  $s$  is considered casting a vote for  $p$ . The higher the number of votes a phrase pair has, the more reliable of the phrase pair. Similarly, the quality of the sentence pair  $s$  is determined by the number of votes casted by the extracted phrase pairs from  $s$ .

In this paper, a PageRank-style random walk algorithm (Brin and Page, 1998; Mihalcea and Tarau, 2004; Wan et al., 2007) is conducted to iteratively compute the importance score of each sentence pair that indicates its quality: the higher the better. Unlike other data filtering methods, our proposed method utilizes the importance scores of sentence pairs as fractional counts to calculate the phrase translation probabilities based on Maximum Likelihood Estimation (MLE), thereby none of the bilingual data is filtered out. Experimental results show that our proposed approach substantially improves the performance in large-scale Chinese-to-English translation tasks.

This work has been done while the first author was visiting Microsoft Research Asia.

## 2 The Proposed Approach

### 2.1 Graph-based random walk

Graph-based random walk is a general algorithm to approximate the importance of a vertex within the graph in a global view. In our method, the vertices denote the sentence pairs and phrase pairs. The importance of each vertex is propagated to other vertices along the edges. Depending on different scenarios, the graph can take directed or undirected, weighted or un-weighted forms. Starting from the initial scores assigned in the graph, the algorithm is applied to recursively compute the importance scores of vertices until it converges, or the difference between two consecutive iterations falls below a pre-defined threshold.

### 2.2 Graph construction

Given the sentence pairs that are word-aligned automatically, an *undirected, weighted* bipartite graph is constructed which maps the sentence pairs and the extracted phrase pairs to the vertices. An edge between a sentence pair vertex and a phrase pair vertex is added if the phrase pair can be extracted from the sentence pair. Mutual reinforcement scores are defined on edges, through which the importance scores are propagated between vertices. Figure 1 illustrates the graph structure. Formally, the bipartite graph is defined as:

$$G = (V, E)$$

where  $V = S \cup P$  is the vertex set,  $S = \{s_i | 1 \leq i \leq n\}$  is the set of all sentence pairs.  $P = \{p_j | 1 \leq j \leq m\}$  is the set of all phrase pairs which are extracted from  $S$  based on the word alignment.  $E$  is the edge set in which the edges are between  $S$  and  $P$ , thereby  $E = \{\langle s_i, p_j \rangle | s_i \in S, p_j \in P, \phi(s_i, p_j) = 1\}$ .

$$\phi(s_i, p_j) = \begin{cases} 1 & \text{if } p_j \text{ can be extracted from } s_i \\ 0 & \text{otherwise} \end{cases}$$

### 2.3 Graph parameters

For sentence-phrase mutual reinforcement, a non-negative score  $r(s_i, p_j)$  is defined using the standard TF-IDF formula:

$$r(s_i, p_j) = \begin{cases} \frac{PF(s_i, p_j) \times IPF(p_j)}{\sum_{p' \in \{p | \phi(s_i, p) = 1\}} PF(s_i, p') \times IPF(p')} & \text{if } \phi(s_i, p_j) = 1 \\ 0 & \text{otherwise} \end{cases}$$

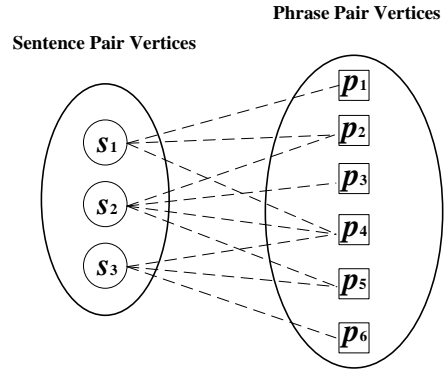


Figure 1: The circular nodes stand for  $S$  and square nodes stand for  $P$ . The lines capture the sentence-phrase mutual reinforcement.

where  $PF(s_i, p_j)$  is the phrase pair frequency in a sentence pair and  $IPF(p_j)$  is the inverse phrase pair frequency of  $p_j$  in the whole bilingual corpus.  $r(s_i, p_j)$  is abbreviated as  $r_{ij}$ .

Inspired by (Brin and Page, 1998; Mihalcea and Tarau, 2004; Wan et al., 2007), we compute the importance scores of sentence pairs and phrase pairs using a PageRank-style algorithm. The weights  $r_{ij}$  are leveraged to reflect the relationships between two types of vertices. Let  $u(s_i)$  and  $v(p_j)$  denote the scores of a sentence pair vertex and a phrase pair vertex. They are computed iteratively by:

$$u(s_i) = (1 - d) + d \times \sum_{j \in N(s_i)} \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} v(p_j)$$

$$v(p_j) = (1 - d) + d \times \sum_{i \in N(s_i)} \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{ik}} u(s_i)$$

where  $d$  is empirically set to the default value 0.85 that is same as the original PageRank,  $N(s_i) = \{j | \langle s_i, p_j \rangle \in E\}$ ,  $M(p_j) = \{i | \langle s_i, p_j \rangle \in E\}$ . The detailed process is illustrated in Algorithm 1. Algorithm 1 iteratively updates the scores of sentence pairs and phrase pairs (lines 10-26). The computation ends when difference between two consecutive iterations is lower than a pre-defined threshold  $\delta$  ( $10^{-12}$  in this study).

### 2.4 Parallelization

When the random walk runs on some large bilingual corpora, even filtering phrase pairs that appear only once would still require several days of CPU time for a number of iterations. To overcome this problem, we use a distributed algorithm

**Algorithm 1** Modified Random Walk

---

```

1: for all  $i \in \{0 \dots |S| - 1\}$  do
2:    $u(s_i)^{(0)} \leftarrow 1$ 
3: end for
4: for all  $j \in \{0 \dots |P| - 1\}$  do
5:    $v(p_j)^{(0)} \leftarrow 1$ 
6: end for
7:  $\delta \leftarrow \text{Infinity}$ 
8:  $\epsilon \leftarrow \text{threshold}$ 
9:  $n \leftarrow 1$ 
10: while  $\delta > \epsilon$  do
11:   for all  $i \in \{0 \dots |S| - 1\}$  do
12:      $F(s_i) \leftarrow 0$ 
13:     for all  $j \in N(s_i)$  do
14:        $F(s_i) \leftarrow F(s_i) + \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} \cdot v(p_j)^{(n-1)}$ 
15:     end for
16:      $u(s_i)^{(n)} \leftarrow (1 - d) + d \cdot F(s_i)$ 
17:   end for
18:   for all  $j \in \{0 \dots |P| - 1\}$  do
19:      $G(p_j) \leftarrow 0$ 
20:     for all  $i \in M(p_j)$  do
21:        $G(p_j) \leftarrow G(p_j) + \frac{r_{ij}}{\sum_{k \in N(s_i)} r_{ik}} \cdot u(s_i)^{(n-1)}$ 
22:     end for
23:      $v(p_j)^{(n)} \leftarrow (1 - d) + d \cdot G(p_j)$ 
24:   end for
25:    $\delta \leftarrow \max(\Delta u(s_i)|_{i=1}^{|S|-1}, \Delta v(p_j)|_{j=1}^{|P|-1})$ 
26:    $n \leftarrow n + 1$ 
27: end while
28: return  $u(s_i)^{(n)}|_{i=0}^{|S|-1}$ 

```

---

based on the iterative computation in the Section 2.3. Before the iterative computation starts, the sum of the outlink weights for each vertex is computed first. The edges are randomly partitioned into sets of roughly equal size. Each edge  $\langle s_i, p_j \rangle$  can generate two key-value pairs in the format  $\langle s_i, r_{ij} \rangle$  and  $\langle p_j, r_{ij} \rangle$ . The pairs with the same key are summed locally and accumulated across different machines. Then, in each iteration, the score of each vertex is updated according to the sum of the normalized inlink weights. The key-value pairs are generated in the format  $\langle s_i, \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} \cdot v(p_j) \rangle$  and  $\langle p_j, \frac{r_{ij}}{\sum_{k \in N(s_i)} r_{ik}} \cdot u(s_i) \rangle$ . These key-value pairs are also randomly partitioned and summed across different machines. Since long sentence pairs usually extract more phrase pairs, we need to normalize the importance scores based on the sentence length. The algorithm fits well into the *MapReduce* programming model (Dean and Ghemawat, 2008) and we use it as our implementation.

## 2.5 Integration into translation modeling

After sufficient number of iterations, the importance scores of sentence pairs (i.e.,  $u(s_i)$ ) are obtained. Instead of simple filtering, we use the

scores of sentence pairs as the fractional counts to re-estimate the translation probabilities of phrase pairs. Given a phrase pair  $p = \langle \bar{f}, \bar{e} \rangle$ ,  $A(\bar{f})$  and  $B(\bar{e})$  indicate the sets of sentences that  $\bar{f}$  and  $\bar{e}$  appear. Then the translation probability is defined as:

$$P_{\text{CW}}(\bar{f}|\bar{e}) = \frac{\sum_{i \in A(\bar{f}) \cap B(\bar{e})} u(s_i) \times c_i(\bar{f}, \bar{e})}{\sum_{j \in B(\bar{e})} u(s_j) \times c_j(\bar{e})}$$

where  $c_i(\cdot)$  denotes the count of the phrase or phrase pair in  $s_i$ .  $P_{\text{CW}}(\bar{f}|\bar{e})$  and  $P_{\text{CW}}(\bar{e}|\bar{f})$  are named as Corpus Weighting (CW) based translation probability, which are integrated into the log-linear model in addition to the conventional phrase translation probabilities (Koehn et al., 2003).

## 3 Experiments

### 3.1 Setup

We evaluated our bilingual data cleaning approach on large-scale Chinese-to-English machine translation tasks. The bilingual data we used was mainly mined from the web (Jiang et al., 2009)<sup>1</sup>, as well as the United Nations parallel corpus released by LDC and the parallel corpus released by China Workshop on Machine Translation (CWMT), which contain around 30 million sentence pairs in total after removing duplicated ones. The development data and testing data is shown in Table 1.

| Data Set                  | #Sentences | Source    |
|---------------------------|------------|-----------|
| NIST 2003 (dev)           | 919        | open test |
| NIST 2005 (test)          | 1,082      | open test |
| NIST 2006 (test)          | 1,664      | open test |
| NIST 2008 (test)          | 1,357      | open test |
| CWMT 2008 (test)          | 1,006      | open test |
| In-house dataset 1 (test) | 1,002      | web data  |
| In-house dataset 2 (test) | 5,000      | web data  |
| In-house dataset 3 (test) | 2,999      | web data  |

Table 1: Development and testing data used in the experiments.

A phrase-based decoder was implemented based on inversion transduction grammar (Wu, 1997). The performance of this decoder is similar to the state-of-the-art phrase-based decoder in Moses, but the implementation is more straightforward. We use the following feature functions in the log-linear model:

<sup>1</sup>Although supervised data cleaning has been done in the post-processing, the corpus still contains a fair amount of noisy data based on our random sampling.

|                        | dev          | NIST 2005    | NIST 2006    | NIST 2008    | CWMT 2008    | IH 1         | IH 2         | IH 3         |
|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| baseline               | 41.24        | 37.34        | 35.20        | 29.38        | 31.14        | 24.29        | 22.61        | 24.19        |
| (Wuebker et al., 2010) | 41.20        | 37.48        | 35.30        | 29.33        | 31.10        | 24.33        | 22.52        | 24.18        |
| -0.25M                 | 41.28        | 37.62        | 35.31        | 29.70        | 31.40        | 24.52        | 22.69        | 24.64        |
| -0.5M                  | 41.45        | 37.71        | 35.52        | 29.76        | 31.77        | 24.64        | 22.68        | 24.69        |
| -1M                    | 41.28        | 37.41        | 35.28        | 29.65        | 31.73        | 24.23        | 23.06        | 24.20        |
| +CW                    | <b>41.75</b> | <b>38.08</b> | <b>35.84</b> | <b>30.03</b> | <b>31.82</b> | <b>25.23</b> | <b>23.18</b> | <b>24.80</b> |

Table 2: BLEU(%) of Chinese-to-English translation tasks on multiple testing datasets ( $p < 0.05$ ), where ”-numberM” denotes we simply filter *number* million low scored sentence pairs from the bilingual data and use others to extract the phrase table. ”CW” means the corpus weighting feature, which incorporates sentence scores from random walk as fractional counts to re-estimate the phrase translation probabilities.

- phrase translation probabilities and lexical weights in both directions (4 features);
- 5-gram language model with Kneser-Ney smoothing (1 feature);
- lexicalized reordering model (1 feature);
- phrase count and word count (2 features).

The translation model was trained over the word-aligned bilingual corpus conducted by GIZA++ (Och and Ney, 2003) in both directions, and the diag-grow-final heuristic was used to refine the symmetric word alignment. The language model was trained on the LDC English Gigaword Version 4.0 plus the English part of the bilingual corpus. The lexicalized reordering model (Xiong et al., 2006) was trained over the 40% randomly sampled sentence pairs from our parallel data. Case-insensitive BLEU4 (Papineni et al., 2002) was used as the evaluation metric. The parameters of the log-linear model are tuned by optimizing BLEU on the development data using MERT (Och, 2003). Statistical significance test was performed using the bootstrap re-sampling method proposed by Koehn (2004).

### 3.2 Baseline

The experimental results are shown in Table 2. In the baseline system, the phrase pairs that appear only once in the bilingual data are simply discarded because most of them are noisy. In addition, the fix-discount method in (Foster et al., 2006) for phrase table smoothing is also used. This implementation makes the baseline system perform much better and the model size is much smaller. In fact, the basic idea of our ”one count” cutoff is very similar to the idea of ”leaving-one-out” in (Wuebker et al., 2010). The results show

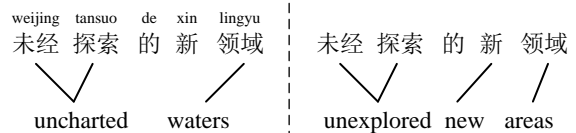


Figure 2: The left one is the non-literal translation in our bilingual corpus. The right one is the literal translation made by human for comparison.

that the ”leaving-one-out” method performs almost the same as our baseline, thereby cannot bring other benefits to the system.

### 3.3 Results

We evaluate the proposed bilingual data cleaning method by incorporating sentence scores into translation modeling. In addition, we also compare with several settings that filtering low-quality sentence pairs from the bilingual data based on the importance scores. The last  $N = \{ 0.25M, 0.5M, 1M \}$  sentence pairs are filtered before the modeling process. Although the simple bilingual data filtering can improve the performance on some datasets, it is difficult to determine the border line and translation performance is fluctuated. One main reason is in the proposed random walk approach, the bilingual sentence pairs with non-literal translations may get lower scores because they appear less frequently compared with those literal translations. Crudely filtering out these data may degrade the translation performance. For example, we have a sentence pair in the bilingual corpus shown in the left part of Figure 2. Although the translation is correct in this situation, translating the Chinese word ”lingyu” to ”waters” appears very few times since the common translations are ”areas” or ”fields”. However, simply filtering out this kind of sentence pairs may lead to some loss of native English expressions, thereby the trans-

lation performance is unstable since both non-parallel sentence pairs and non-literal but parallel sentence pairs are filtered. Therefore, we use the importance score of each sentence pair to estimate the phrase translation probabilities. It consistently brings substantial improvements compared to the baseline, which demonstrates graph-based random walk indeed improves the translation modeling performance for our SMT system.

### 3.4 Discussion

In (Goutte et al., 2012), they evaluated phrase-based SMT systems trained on parallel data with different proportions of synthetic noisy data. They suggested that when collecting larger, noisy parallel data for training phrase-based SMT, cleaning up by trying to detect and remove incorrect alignments can actually degrade performance. Our experimental results confirm their findings on some datasets. Based on our method, sometimes filtering noisy data leads to unexpected results. The reason is two-fold: on the one hand, the non-literal parallel data makes false positive in noisy data detection; on the other hand, large-scale SMT systems is relatively robust and tolerant to noisy data, especially when we remove frequency-1 phrase pairs. Therefore, we propose to integrate the importance scores when re-estimating phrase pair probabilities in this paper. The importance scores can be considered as a kind of contribution constraint, thereby high-quality parallel data contributes more while noisy parallel data contributes less.

## 4 Conclusion and Future Work

In this paper, we develop an effective approach to clean the bilingual data using graph-based random walk. Significant improvements on several datasets are achieved in our experiments. For future work, we will extend our method to explore the relationships of sentence-to-sentence and phrase-to-phrase, which is beyond the existing sentence-to-phrase mutual reinforcement.

### Acknowledgments

We are especially grateful to Yajuan Duan, Hong Sun, Nan Yang and Xilun Chen for the helpful discussions. We also thank the anonymous reviewers for their insightful comments.

## References

- 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.
- Jeffrey Dean and Sanjay Ghemawat. 2008. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.
- George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia, July. Association for Computational Linguistics.
- Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of AMTA 2012*, San Diego, California, October. Association for Machine Translation in the Americas.
- Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu, and Qingsheng Zhu. 2009. Mining bilingual data from the web with adaptively learnt patterns. 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 870–878, Suntec, Singapore, August. Association for Computational Linguistics.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003 Main Papers*, pages 48–54, Edmonton, May-June. Association for Computational Linguistics.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.
- Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.
- 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(4):417–449.



- 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, Sapporo, Japan, July. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.
- Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Sydney, Australia, July. Association for Computational Linguistics.
- Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic, June. Association for Computational Linguistics.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.
- Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484, Uppsala, Sweden, July. Association for Computational Linguistics.
- Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.

# Automatically Predicting Sentence Translation Difficulty

Abhijit Mishra\*, Pushpak Bhattacharyya\*, Michael Carl†

\* Department of Computer Science and Engineering, IIT Bombay, India

{abhijitmishra,pb}@cse.iitb.ac.in

† CRITT, IBC, Copenhagen Business School, Denmark,

mc.ibc@cbs.dk

## Abstract

In this paper we introduce *Translation Difficulty Index* (TDI), a measure of difficulty in text translation. We first define and quantify translation difficulty in terms of TDI. We realize that any measure of TDI based on *direct* input by translators is fraught with subjectivity and ad-hocism. We, rather, rely on *cognitive evidences* from eye tracking. TDI is measured as the sum of *fixation (gaze)* and *saccade (rapid eye movement)* times of the eye. We then establish that TDI is correlated with three properties of the input sentence, *viz.* *length (L)*, *degree of polysemy (DP)* and *structural complexity (SC)*. We train a *Support Vector Regression* (SVR) system to predict TDIs for new sentences using these features as input. The prediction done by our framework is well correlated with the empirical gold standard data, which is a repository of  $\langle L, DP, SC \rangle$  and *TDI* pairs for a set of sentences. The primary use of our work is a way of “binning” sentences (to be translated) in “easy”, “medium” and “hard” categories as per their predicted TDI. This can decide pricing of any translation task, especially useful in a scenario where parallel corpora for *Machine Translation* are built through translation crowdsourcing/outsourcing. This can also provide a way of monitoring progress of second language learners.

## 1 Introduction

Difficulty in translation stems from the fact that most words are polysemous and sentences can be long and have complex structure. While *length of sentence* is commonly used as a translation difficulty indicator, *lexical* and *structural* properties of

a sentence also contribute to translation difficulty. Consider the following example sentences.

1. *The camera-man shot the policeman with a gun. (length-8)*
2. *I was returning from my old office yesterday. (length-8)*

Clearly, sentence 1 is more difficult to process and translate than sentence 2, since it has lexical ambiguity (“*Shoot*” as an act of firing a shot or taking a photograph?) and structural ambiguity (*Shot with a gun* or *policeman with a gun?*). To produce fluent and adequate translations, efforts have to be put to analyze both the lexical and syntactic properties of the sentences.

The most recent work on studying translation difficulty is by Campbell and Hale (1999) who identified several areas of difficulty in lexis and grammar. “Reading” researchers have focused on developing readability formulae, since 1970. The *Flesch-Kincaid Readability test* (Kincaid et al., 1975), the *Fry Readability Formula* (Fry, 1977) and the *Dale-Chall readability formula* (Chall and Dale, 1999) are popular and influential. These formulae use factors such as vocabulary difficulty (or semantic factors) and sentence length (or syntactic factors). In a different setting, Malsburg et al. (2012) correlate eye fixations and scanpaths of readers with sentence processing. While these approaches are successful in quantifying readability, they may not be applicable to translation scenarios. The reason is that, translation is not merely a reading activity. Translation requires co-ordination between source text comprehension and target text production (Dragsted, 2010). To the best of our knowledge, our work on predicting TDI is the first of its kind.

The motivation of the work is as follows. Currently, for domain specific Machine Translation systems, parallel corpora are gathered through translation crowdsourcing/outsourcing. In such

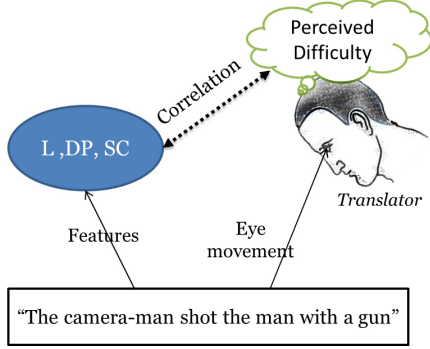


Figure 1: Inherent sentence complexity and perceived difficulty during translation

a scenario, translators are paid on the basis of sentence length, which ignores other factors contributing to translation difficulty, as stated above. Our proposed Translation Difficulty Index (TDI) quantifies the translation difficulty of a sentence considering both lexical and structural properties. This measure can, in turn, be used to cluster sentences according to their difficulty levels (*viz.* easy, medium, hard). Different payment and schemes can be adopted for different such clusters.

TDI can also be useful for training and evaluating second language learners. For example, appropriate examples at particular levels of difficulty can be chosen for giving assignments and monitoring progress.

The rest of the paper is organized in the following way. Section 2 describes TDI as function of translation processing time. Section 3 is on measuring translation processing time through eye tracking. Section 4 gives the correlation of linguistic complexity with observed TDI. In section 5, we describe a technique for predicting TDIs and ranking unseen sentences using *Support Vector Machines*. Section 6 concludes the paper with pointers to future work.

## 2 Quantifying Translation Difficulty

As a first approximation, TDI of a sentence can be the *time taken to translate* the sentence, which can be measured through simple translation experiments. This is based on the assumption that more difficult sentences will require more time to translate. However, “time taken to translate” may not be strongly related to the translation difficulty for two reasons. First, it is difficult to know what fraction of the total translation time is actually spent on the translation-related-thinking. For ex-

ample, translators may spend considerable amount of time typing/writing translations, which is irrelevant to the translation difficulty. Second, the translation time is sensitive to distractions from the environment. So, instead of the “time taken to translate”, we are more interested in the “time for which translation related processing is carried out by the brain”. This can be termed as the *Translation Processing Time* ( $T_p$ ). Mathematically,

$$T_p = T_{p.comp} + T_{p.gen} \quad (1)$$

Where  $T_{p.comp}$  and  $T_{p.gen}$  are the processing times for source text comprehension and target text generation respectively. The empirical TDI, is computed by normalizing  $T_p$  with sentence length.

$$TDI = \frac{T_p}{sentencelength} \quad (2)$$

Measuring  $T_p$  is a difficult task as translators often switch between thinking and writing activities. Here comes the role of *eye tracking*.

## 3 Measuring $T_p$ by eye-tracking

We measure  $T_p$  by analyzing the gaze behavior of translators through eye-tracking. The rationale behind using eye-tracking is that, humans spend time on what they see, and this “time” is correlated with the complexity of the information being processed, as shown in Figure 1. Two fundamental components of eye behavior are (a) *Gaze-fixation* or simply, *Fixation* and (b) *Saccade*. The former is a long stay of the visual gaze on a single location. The latter is a very rapid movement of the eyes between positions of rest. An intuitive feel for these two concepts can be had by considering the example of translating the sentence *The camera-man shot the policeman with a gun* mentioned in the introduction. It is conceivable that the eye will linger long on the word “shot” which is ambiguous and will rapidly move across “shot”, “camera-man” and “gun” to ascertain the clue for disambiguation.

The terms  $T_{p.comp}$  and  $T_{p.gen}$  in (1) can now be looked upon as the sum of fixation and saccadic durations for both source and target sentences respectively.

Modifying 1

$$T_p = \sum_{f \in F_s} dur(f) + \sum_{s \in S_s} dur(s) + \sum_{f \in F_t} dur(f) + \sum_{s \in S_t} dur(s) \quad (3)$$

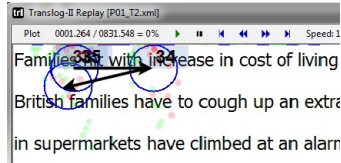


Figure 2: Screenshot of Translog. The circles represent fixations and arrow represent saccades.

Here,  $F_s$  and  $S_s$  correspond to sets of fixations and saccades for source sentence and  $F_t$  and  $S_t$  correspond to those for the target sentence respectively.  $dur$  is a function returning the duration of fixations and saccades.

### 3.1 Computing TDI using eye-tracking database

We obtained TDIs for a set of sentences from the Translation Process Research Database (TPR 1.0)(Carl, 2012). The database contains translation studies for which gaze data is recorded through the Translog software<sup>1</sup>(Carl, 2012). Figure 2 presents a screendump of Translog. Out of the 57 available sessions, we selected 40 translation sessions comprising 80 sentence translations<sup>2</sup>. Each of these 80 sentences was translated from English to three different languages, *viz.* Spanish, Danish and Hindi by at least 2 translators. The translators were young professional linguists or students pursuing PhD in linguistics.

The eye-tracking data is noisy and often exhibits *systematic errors* (Hornof and Halverson, 2002). To correct this, we applied automatic error correction technique (Mishra et al., 2012) followed by manually correcting incorrect gaze-to-word mapping using Translog. Note that, gaze and saccadic durations may also depend on the translator’s reading speed. We tried to rule out this effect by sampling out translations for which the variance in participant’s reading speed is minimum. Variance in reading speed was calculated after taking a samples of source text for each participant and measuring the time taken to read the text.

After preprocessing the data, TDI was computed for each sentence by using (2) and (3).The observed unnormalized TDI score<sup>3</sup> ranges from 0.12 to 0.86. We normalize this to a [0,1] scale

<sup>1</sup><http://www.translog.dk>

<sup>2</sup>20% of the translation sessions were discarded as it was difficult to rectify the gaze logs for these sessions.

<sup>3</sup>Anything beyond the upper bound is hard to translate and can be assigned with the maximum score.

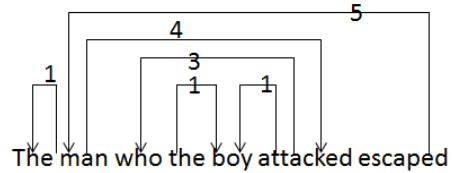


Figure 3: Dependency graph used for computing SC

using MinMax normalization.

If the “time taken to translate” and  $T_p$  were strongly correlated, we would have rather opted “time taken to translate” for the measurement of TDI. The reason is that “time taken to translate” is relatively easy to compute and does not require expensive setup for conducting “eye-tracking” experiments. But our experiments show that there is a weak correlation (coefficient = 0.12) between “time taken to translate” and  $T_p$ . This makes us believe that  $T_p$  is still the best option for TDI measurement.

## 4 Relating TDI to sentence features

Our claim is that translation difficulty is mainly caused by three features: *Length*, *Degree of Polysemy* and *Structural Complexity*.

### 4.1 Length

It is the total number of words occurring in a sentence.

### 4.2 Degree of Polysemy (DP)

The degree of polysemy of a sentence is the sum of senses possessed by each word in the Wordnet normalized by the sentence length. Mathematically,

$$DP_{sentence} = \frac{\sum_{w \in W} Senses(w)}{length(sentence)} \quad (4)$$

Here,  $Senses(w)$  retrieves the total number senses of a word P from the Wordnet.  $W$  is the set of words appearing in the sentence.

### 4.3 Structural Complexity (SC)

Syntactically, words, phrases and clauses are attached to each other in a sentence. If the attachment units lie far from each other, the sentence has higher structural complexity. Lin (1996) defines it as the *total length of dependency links in the dependency structure of the sentence*.

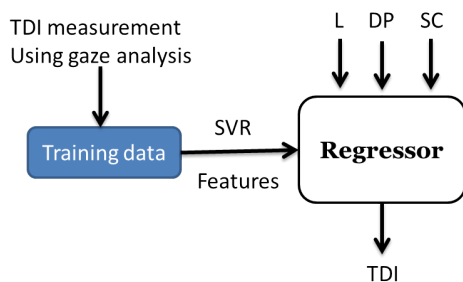


Figure 4: Prediction of TDI using linguistic properties such as Length(L), Degree of Polysemy (DP) and Structural Complexity (SC)

Example: *The man who the boy attacked escaped.*

Figure 3 shows the dependency graph for the example sentence. The weights of the edges correspond how far the two connected words lie from each other in the sentence. Using Lin’s formula, the SC score for the example sentence turns out to be 15.

Lin’s way of computing SC is affected by sentence length since the number of dependency links for a sentence depends on its length. So we normalize SC by the length of the sentence. After normalization, the SC score for the example given becomes  $15/7 = 2.14$

#### 4.4 How are TDI and linguistic features related

To validate that translation difficulty depends on the above mentioned linguistic features, we tried to find out the correlation coefficients between each feature and empirical TDI. We extracted three sets of sample sentences. For each sample, sentence selection was done with a view to varying one feature, keeping the other two constant. The Correlation Coefficients between L, DP and SC and the empirical TDI turned out to be **0.72**, **0.41** and **0.63** respectively. These positive correlation coefficients indicate that all the features contribute to the translation difficulty.

## 5 Predicting TDI

Our system predicts TDI from the linguistic properties of a sentence as shown in Figure 4.

The prediction happens in a supervised setting through regression. Training such a system requires a set sentences annotated with TDIs. In our case, direct annotation of TDI is a difficult and unintuitive task. So, we annotate TDI by observ-

| Kernel(C=3.0)       | MSE (%)      | Correlation |
|---------------------|--------------|-------------|
| Linear              | 20.64        | 0.69        |
| <b>Poly (Deg 2)</b> | <b>12.88</b> | <b>0.81</b> |
| Poly (Deg 3)        | 13.35        | 0.78        |
| Rbf (default)       | 13.32        | 0.73        |

Table 1: Relative MSE and Correlation with observed data for different kernels used for SVR.

ing translator’s behavior (using equations (1) and (2)) instead of asking people to rate sentences with TDI.

We are now prepared to give the regression scenario for predicting TDI.

### 5.1 Preparing the dataset

Our dataset contains 80 sentences for which TDI have been measured (Section 3.1). We divided this data into 10 sets of training and testing datasets in order to carry out a 10-fold evaluation. DP and SC features were computed using Princeton Wordnet<sup>4</sup> and Stanford Dependence Parser<sup>5</sup>.

### 5.2 Applying Support Vector Regression

To predict TDI, Support Vector Regression (SVR) technique (Joachims et al., 1999) was preferred since it facilitates multiple kernel-based methods for regression. We tried using different kernels using default parameters. Error analysis was done by means of Mean Squared Error estimate (MSE). We also measured the Pearson correlation coefficient between the empirical and predicted TDI for our test-sets.

Table 1 indicates Mean Square Error percentages for different kernel methods used for SVR. MSE (%) indicates by what percentage the predicted TDIs differ from the observed TDIs. In our setting, quadratic polynomial kernel with  $c=3.0$  outperforms other kernels. The predicted TDIs are well correlated with the empirical TDIs. This tells us that even if the predicted scores are not as accurate as desired, the system is capable of ranking sentences in correct order. Table 2 presents examples from the test dataset for which the observed TDI ( $TDI_O$ ) and the TDI predicted by polynomial kernel based SVR ( $TDI_P$ ) are shown.

Our larger goal is to group unknown sentences into different categories by the level of transla-

<sup>4</sup><http://www.wordnet.princeton.edu>

<sup>5</sup><http://www.nlp.stanford.edu/software/lex-parser.html>

| Example  | L  | DP | SC  | $TDI_O$ | $TDI_P$ | Error |
|--|----|----|-----|---------|---------|-------|
| 1. American Express recently announced a second round of job cuts. | 10 | 10 | 1.8 | 0.24    | 0.23    | 4%    |
| 2. Sociology is a relatively new academic discipline.              | 7  | 6  | 3.7 | 0.49    | 0.53    | 8%    |

Table 2: Example sentences from the test dataset.

tion difficulty. For that, we tried to manually assign three different class labels to sentences *viz. easy, medium and hard* based on the empirical TDI scores. The ranges of scores chosen for easy, medium and hard categories were [0-0.3], [0.3-0.75] and [0.75-1.0] respectively (by trial and error). Then we trained a *Support Vector Rank* (Joachims, 2006) with default parameters using different kernel methods. The ranking framework achieves a maximum **67.5%** accuracy on the test data. The accuracy should increase by adding more data to the training dataset.

## 6 Conclusion

This paper introduces an approach to quantifying translation difficulty and automatically assigning difficulty levels to unseen sentences. It establishes a relationship between the intrinsic sentential properties, *viz., length (L), degree of polysemy (DP) and structural complexity (SC)*, on one hand and the Translation Difficulty Index (*TDI*), on the other. Future work includes deeper investigation into other linguistic factors such as presence of domain specific terms, target language properties *etc.* and applying more sophisticated cognitive analysis techniques for more reliable TDI score. We would like to make use of *inter-annotator agreement* to decide the boundaries for the translation difficulty categories. Extending the study to different language pairs and studying the applicability of this technique for Machine Translation Quality Estimation are also on the agenda.

## Acknowledgments

We would like to thank the CRITT, CBS group for their help in manual correction of TPR data. In particular, thanks to Barto Mesa and Khristina for helping with Spanish and Danish dataset corrections.

## References

- Campbell, S., and Hale, S. 1999. What makes a text difficult to translate? *Refereed Proceedings of the 23rd Annual ALAA Congress*.
- Carl, M. 2012. Translog-II: A Program for Recording User Activity Data for Empirical Reading and Writing Research. In *Proceedings of the Eight International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA)*
- Carl, M. 2012. The CRITT TPR-DB 1.0: A Database for Empirical Human Translation Process Research. *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP-2012)*.
- Chall, J. S., and Dale, E. 1995. *Readability revisited: the new Dale-Chall readability formula* Cambridge, Mass.: Brookline Books.
- Dragsted, B. 2010. Co-ordination of reading and writing processes in translation. *Contribution to Translation and Cognition, Shreve, G. and Angelone, E.(eds.)Cognitive Science Society*.
- Fry, E. 1977 *Fry's readability graph: Clarification, validity, and extension to level 17* *Journal of Reading*, 21(3), 242-252.
- Hornof, A. J. and Halverson, T. 2002 Cleaning up systematic error in eye-tracking data by using required fixation locations. *Behavior Research Methods, Instruments, and Computers*, 34, 592604.
- Joachims, T., Schlkopf, B., Burges, C and A. Smola (ed.). 1999. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999,
- Joachims, T. 2006 Training Linear SVMs in Linear Time. *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*.
- Kincaid, J. P., Fishburne, R. P., Jr., Rogers, R. L., and Chissom, B. S. 1975. *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel* Millington, Tennessee: Naval Air Station Memphis, pp. 8-75.

- Lin, D. 1996 On the structural complexity of natural language sentences. *Proceeding of the 16th International Conference on Computational Linguistics (COLING)*, pp. 729733.
- Mishra, A., Carl, M, Bhattacharyya, P. 2012 A heuristic-based approach for systematic error correction of gaze data for reading. In Michael Carl, P.B. and Choudhary, K.K., editors, *Proceedings of the First Workshop on Eye-tracking and Natural Language Processing*, Mumbai, India. The COLING 2012 Organizing Committee
- von der Malsburg, T., Vasishth, S., and Kliegl, R. 2012 *Scanpaths in reading are informative about sentence processing*. In Michael Carl, P.B. and Choudhary, K.K., editors, *Proceedings of the First Workshop on Eye-tracking and Natural Language Processing*, Mumbai, India. The COLING 2012 Organizing Committee

# Learning to Prune: Context-Sensitive Pruning for Syntactic MT

**Wenduan Xu**

Computer Laboratory  
University of Cambridge  
wenduan.xu@cl.cam.ac.uk

**Yue Zhang**

Singapore University of  
Technology and Design  
yue\_zhang@sutd.edu.sg

**Philip Williams and Philipp Koehn**

School of Informatics  
University of Edinburgh  
p.j.williams-2@sms.ed.ac.uk  
pkoehn@inf.ed.ac.uk

## Abstract

We present a context-sensitive chart pruning method for CKY-style MT decoding. Source phrases that are unlikely to have aligned target constituents are identified using sequence labellers learned from the parallel corpus, and speed-up is obtained by pruning corresponding chart cells. The proposed method is easy to implement, orthogonal to cube pruning and additive to its pruning power. On a full-scale English-to-German experiment with a string-to-tree model, we obtain a speed-up of more than 60% over a strong baseline, with no loss in BLEU.

## 1 Introduction

Syntactic MT models suffer from decoding efficiency bottlenecks introduced by online  $n$ -gram language model integration and high grammar complexity. Various efforts have been devoted to improving decoding efficiency, including hypergraph rescoring (Heafield et al., 2013; Huang and Chiang, 2007), coarse-to-fine processing (Petrov et al., 2008; Zhang and Gildea, 2008) and grammar transformations (Zhang et al., 2006). For more expressive, linguistically-motivated syntactic MT models (Galley et al., 2004; Galley et al., 2006), the grammar complexity has grown considerably over hierarchical phrase-based models (Chiang, 2007), and decoding still suffers from efficiency issues (DeNero et al., 2009).

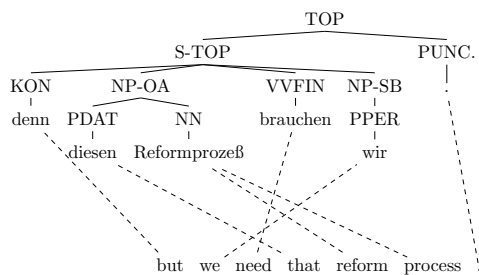
In this paper, we study a chart pruning method for CKY-style MT decoding that is orthogonal to

cube pruning (Chiang, 2007) and additive to its pruning power. The main intuition of our method is to find those source phrases (i.e. any sequence of consecutive words) that are unlikely to have any consistently aligned target counterparts according to the source context and grammar constraints. We show that by using highly-efficient sequence labelling models learned from the bitext used for translation model training, such phrases can be effectively identified prior to MT decoding, and corresponding chart cells can be excluded for decoding without affecting translation quality.

We call our method *context-sensitive pruning* (CSP); it can be viewed as a bilingual adaptation of similar methods in monolingual parsing (Roark and Hollingshead, 2008; Zhang et al., 2010) which improve parsing efficiency by “closing” chart cells using binary classifiers. Our contribution is that we demonstrate such methods can be applied to synchronous-grammar parsing by labelling the source-side alone. This is achieved through a novel training scheme where the labelling models are trained over the word-aligned bitext and gold-standard pruning labels are obtained by projecting target-side constituents to the source words. To our knowledge, this is the first work to apply this technique to MT decoding.

The proposed method is easy to implement and effective in practice. Results on a full-scale English-to-German experiment show that it gives more than 60% speed-up over a strong cube pruning baseline, with no loss in BLEU. While we use a string-to-tree model in this paper, the approach can be adapted to other syntax-based models.





|       |       |   |  |
|-------|-------|---|--|
| $r_1$ | KON   | → | $\langle$ but, denn $\rangle$  |
| $r_2$ | NP-SB | → | $\langle$ we, wir $\rangle$  |
| $r_3$ | NP-OA | → | $\langle$ that reform process, diesen Reformprozeß $\rangle$                   |
| $r_4$ | TOP   | → | $\langle$ $X_1 \dots$ , S-TOP $_1 \dots$ $\rangle$                             |
| $r_5$ | S-TOP | → | $\langle$ but $X_1$ need $X_2$ , denn NP-OA $_2$ brauchen NP-SB $_1$ $\rangle$ |

Figure 1: A selection of grammar rules extractable from an example word-aligned sentence pair.

## 2 The Baseline String-to-Tree Model

Our baseline translation model uses the rule extraction algorithm of Chiang (2007) adapted to a string-to-tree grammar. After extracting phrasal pairs using the standard approach of Koehn et al. (2003), all pairs whose target phrases are not exhaustively dominated by a constituent of the parse tree are removed and each remaining pair,  $\langle \bar{f}, \bar{e} \rangle$ , together with its constituent label,  $C$ , forms a lexical grammar rule:  $C \rightarrow \langle \bar{f}, \bar{e} \rangle$ . The rules  $r_1$ ,  $r_2$ , and  $r_3$  in Figure 1 are lexical rules. Non-lexical rules are generated by eliminating one or more pairs of terminal substrings from an existing rule and substituting non-terminals. This process produces the example rules  $r_4$  and  $r_5$ .

Our decoding algorithm is a variant of CKY and is similar to other algorithms tailored for specific syntactic translation grammars (DeNero et al., 2009; Hopkins and Langmead, 2010). By taking the source-side of each rule, projecting onto it the non-terminal labels from the target-side, and weighting the grammar according to the model’s local scoring features, decoding is a straightforward extension of monolingual weighted chart parsing. Non-local features, such as  $n$ -gram language model scores, are incorporated through cube pruning (Chiang, 2007).

## 3 Chart Pruning

### 3.1 Motivations

The abstract rules and large non-terminal sets of many syntactic MT grammars cause translation

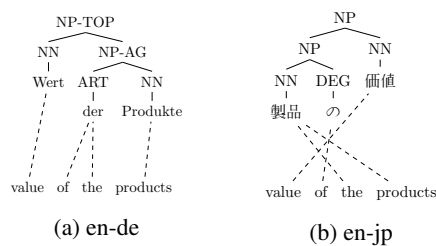


Figure 2: Two example alignments. In (a) “the products” does not have a consistent alignment on the target side, while it does in (b).

overgeneration at the span level and render decoding inefficient. Prior work on monolingual syntactic parsing has demonstrated that by excluding chart cells that are likely to violate constituent constraints, decoding efficiency can be improved with no loss in accuracy (Roark and Hollingshead, 2008). We consider a similar mechanism for syntactic MT decoding by prohibiting subtranslation generation for chart cells violating synchronous-grammar constraints.

A motivating example is shown in Figure 2a, where a segment of an English-German sentence pair from the training data, along with its word alignment and target-side parse tree is depicted. The English phrases “value of” and “the products” do not have corresponding German translations in this example. Although the grammar may have rules to translate these two phrases, they can be safely pruned for this particular sentence pair.

In contrast to chart pruning for monolingual parsing, our pruning decisions are based on the source context, its target translation and the mapping between the two. This distinction is important since the syntactic correspondence between different language pairs is different. Suppose that we were to translate the same English sentence into Japanese (Figure 2a); unlike the English to German example, the English phrase “the products” will be a valid phrase that has a Japanese translation under a target constituent, since it is syntactically aligned to “製品” (Figure 2b).

The key question to consider is how to inject target syntax and word alignment information into our labelling models, so that pruning decisions can be based on the source alone, we address this in the following two sections.

### 3.2 Pruning by Labelling

We use binary tags to indicate whether a source word can start or end a multi-word phrase that has



Figure 3: The pruning effects of two types of binary tags. The shaded cells are pruned and two types of tags are assigned independently.

a consistently aligned target constituent. We call these two types the *b*-tag and the *e*-tag, respectively, and use the set of values  $\{0, 1\}$  for both.

Under this scheme, a *b*-tag value of 1 indicates that a source word can be the start of a source phrase that has a consistently aligned target phrase; similarly an *e*-tag of 0 indicates that a word cannot end a source phrase. If either the *b*-tag or the *e*-tag of an input phrase is 0, the corresponding chart cells will be pruned. The pruning effects of the two types of tags are illustrated in Figure 3. In general, 0-valued *b*-tags prune a whole column of chart cells and 0-valued *e*-tags prune a whole diagonal of cells; and the chart cells on the first row and the top-most cell are always kept so that complete translations can always be found.

We build a separate labeller for each tag type using gold-standard *b*- and *e*-tags, respectively. We train the labellers with maximum-entropy models (Curran and Clark, 2003; Ratnaparkhi, 1996), using features similar to those used for supertagging for CCG parsing (Clark and Curran, 2004). In each case, features for a pruning tag consist of word and POS uni-grams extracted from the 5-word window with the current word in the middle, POS trigrams ending with the current word, as well as two previous tags as a bigram and two separate uni-grams. Our pruning labellers are highly efficient, run in linear time and add little overhead to decoding. During testing, in order to prevent over-pruning, a probability cutoff value  $\theta$  is used. A tag value of 0 is assigned to a word only if its marginal probability is greater than  $\theta$ .

### 3.3 Gold-standard Pruning Tags

Gold-standard tags are extracted from the word-aligned bitext used for translation model training, respecting rule extraction constraints, which is crucial for the success of our method.

For each training sentence pair, gold-standard *b*-tags and *e*-tags are assigned separately to the

---

#### Algorithm 1 Gold-standard Labelling Algorithm

**Input** forward alignment  $A_{e \sim f}$ , backward alignment  $\hat{A}_{f \sim e}$  and 1-best parse tree  $\tau$  for  $f$

**Output** Tag sequences  $\mathbf{b}$  and  $\mathbf{e}$  for  $e$

```

1: procedure TAG( $e, f, \tau, A, \hat{A}$ )
2:    $l \leftarrow |e|$ 
3:   for  $i \leftarrow 0$  to  $l - 1$  do
4:      $\mathbf{b}[i] \leftarrow 0, \mathbf{e}[i] \leftarrow 0$ 
5:   for  $f[i', j']$  in  $\tau$  do
6:      $\mathbf{s} \leftarrow \{\hat{A}[k] \mid k \in [i', j']\}$ 
7:     if  $|\mathbf{s}| \leq 1$  then continue
8:      $i \leftarrow \min(\mathbf{s}), j \leftarrow \max(\mathbf{s})$ 
9:     if CONSISTENT( $i, j, i', j'$ ) then
10:       $\mathbf{b}[i'] \leftarrow 1, \mathbf{e}[j'] \leftarrow 1$ 

11: procedure CONSISTENT( $i, j, i', j'$ )
12:    $\mathbf{t} \leftarrow \{A[k] \mid k \in [i, j]\}$ 
13:   return  $\min(\mathbf{t}) \geq i'$  and  $\max(\mathbf{t}) \leq j'$ 

```

---

source words. First, we initialize both tags of each source word to 0s. Then, we iterate through all target constituent spans, and for each span, we find its corresponding source phrase, as determined by the word alignment. If a constituent exists for the phrase pair, the *b*-tag of the *first* word and the *e*-tag of the *last* word in the source phrase are set to 1s, respectively. Pseudocode is shown in Algorithm 1.

Note that our definition of the gold-standard allows source-side labels to integrate bilingual information. On line 6, the target-side syntax is projected to the source; on line 9, consistency is checked against word alignment.

Consider again the alignment in Figure 2a. Taking the target constituent span covering “der Produkte” as an example, the source phrase under a consistent word alignment is “of the products”. Thus, the *b*-tag of “of” and the *e*-tag of “products” are set to 1s. After considering all target constituent spans, the complete *b*- and *e*-tag sequences for the source-side phrase in Figure 2a are  $[1, 1, 0, 0]$  and  $[0, 0, 1, 1]$ , respectively. Note that, since we never prune single-word spans, we ignore source phrases under consistent one-to-one or one-to-many alignments.

From the gold standard data, we found 73.69% of the 54M words do not begin a multi-word aligned phrase and 77.71% do not end a multi-word aligned phrase; the 1-best accuracies of the two labellers tested on a held-out 20K sentences are 82.50% and 88.78% respectively.

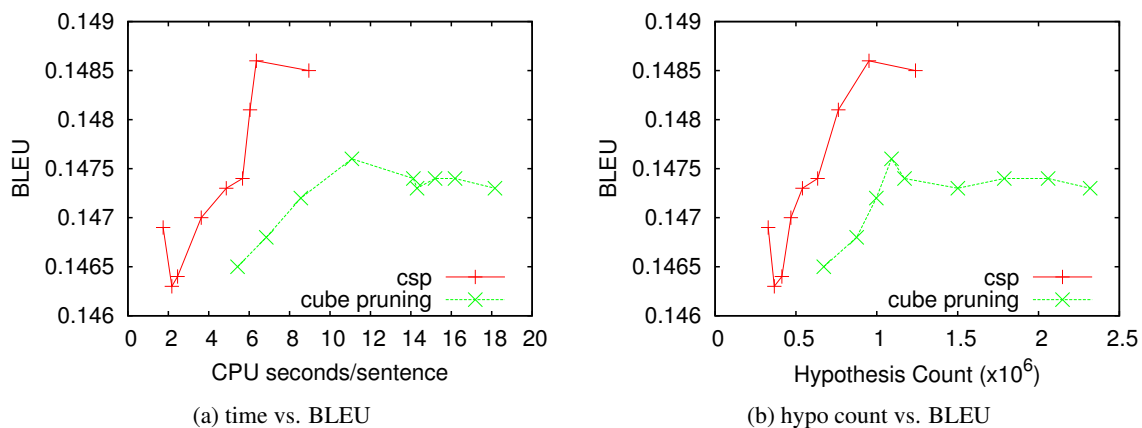


Figure 4: Translation quality comparison with the cube pruning baseline.

## 4 Experiments

### 4.1 Setup

A Moses (Koehn et al., 2007) string-to-tree system is used as our baseline. The training corpus consists of the English-German sections of the Europarl (Koehn, 2005) and the News Commentary corpus. Discarding pairs without target-side parses, the final training data has 2M sentence pairs, with 54M and 52M words on the English and German sides, respectively. Word-alignments are obtained by running GIZA++ (Och and Ney, 2000) in both directions and refined with “grow-diag-final-and” (Koehn et al., 2003). For all experiments, a 5-gram language model with Kneser-Ney smoothing (Chen and Goodman, 1996) built with the SRILM Toolkit (Stolcke and others, 2002) is used.

The development and test sets are the 2008 WMT newstest (2,051 sentences) and 2009 WMT newstest (2,525 sentences) respectively. Feature weights are tuned with MERT (Och, 2003) on the development set and output is evaluated using case-sensitive BLEU (Papineni et al., 2002). For both rule extraction and decoding, up to seven terminal/non-terminal symbols on the source-side are allowed. For decoding, the maximum span-length is restricted to 15, and the grammar is pre-filtered to match the entire test set for both the baseline system and the chart pruning decoder.

We use two labellers to perform *b*- and *e*-tag labelling independently prior to decoding. Training of the labelling models is able to complete in under 2.5 hours and the whole test set is labelled in under 2 seconds. A standard perceptron POS tagger (Collins, 2002) trained on Wall Street Journal sections 2-21 of the Penn Treebank is used to as-

sign POS tags for both our training and test data.

### 4.2 Results

Figures 4a and 4b compare CSP with the cube pruning baseline in terms of BLEU. Decoding speed is measured by the average decoding time and average number of hypotheses generated per sentence. We first run the baseline decoder under various beam settings ( $b = 100 - 2500$ ) until no further increase in BLEU is observed. We then run the CSP decoder with a range of  $\theta$  values ( $\theta = 0.91 - 0.99$ ), at the default beam size of 1000 of the baseline decoder. The CSP decoder, which considers far fewer chart cells and generates significantly fewer subtranslations, consistently outperforms the slower baseline. It ultimately achieves a BLEU score of 14.86 at a probability cutoff value of 0.98, slightly higher than the highest score of the baseline.

At all levels of comparable translation quality, our decoder is faster than the baseline. On average, the speed-up gained is 63.58% as measured by average decoding time, and comparing on a point-by-point basis, our decoder always runs over 60% faster. At the  $\theta$  value of 0.98, it yields a speed-up of 57.30%, compared with a beam size of 400 for the baseline, where both achieved the highest BLEU.

Figures 5a and 5b demonstrate the pruning power of CSP ( $\theta = 0.95$ ) in comparison with the baseline (beam size = 300); across all the cutoff values and beam sizes, the CSP decoder considers 54.92% fewer translation hypotheses on average and the minimal reduction achieved is 46.56%.

Figure 6 shows the percentage of spans of different lengths pruned by CSP ( $\theta = 0.98$ ). As ex-

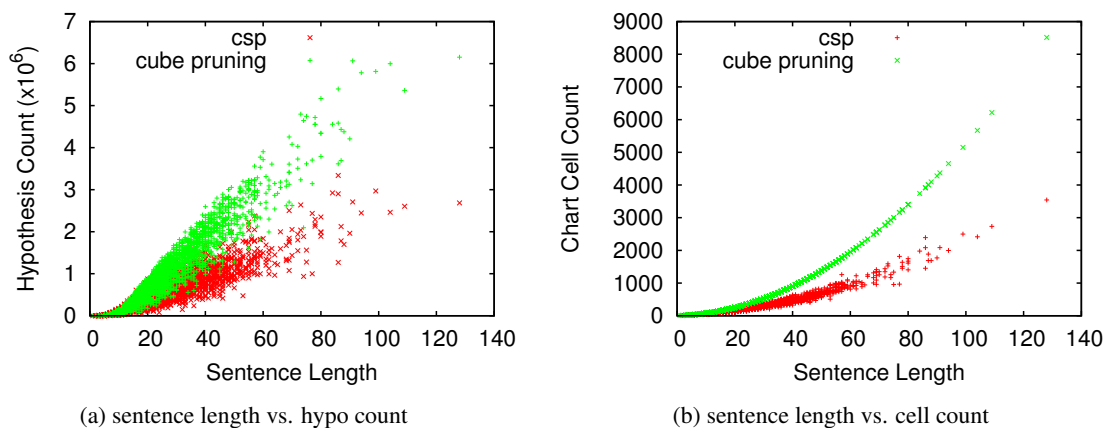


Figure 5: Search space comparison with the cube pruning baseline.

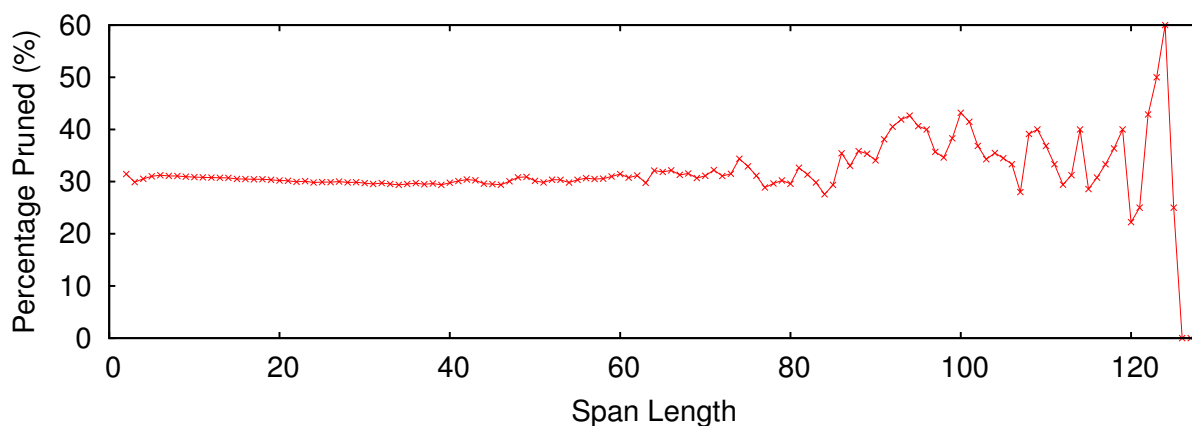


Figure 6: Percentage of spans of different lengths pruned at  $\theta = 0.98$ .

pected, longer spans are pruned more often, as they are more likely to be at the intersections of cells pruned by the two types of pruning labels, thus can be pruned by either type.

We also find CSP does not improve search quality and it leads to slightly lower model scores, which shows that some higher scored translation hypotheses are pruned. This, however, is perfectly desirable. Since our pruning decisions are based on independent labellers using contextual information, with the objective of eliminating unlikely subtranslations and rule applications. It may even offset defects of the translation model (i.e. high-scored bad translations). The fact that the output BLEU did not decrease supports this reasoning.

Finally, it is worth noting that our string-to-tree model does not force complete target parses to be built during decoding, which is not required in our pruning method either. We do not use any other heuristics (other than keeping singleton and the top-most cells) to make complete translation always possible. The hypothesis here is that good

labelling models should not affect the derivation of complete target translations.

## 5 Conclusion

We presented a novel sequence labelling based, context-sensitive pruning method for a string-to-tree MT model. Our method achieves more than 60% speed-up over a state-of-the-art baseline on a full-scale translation task. In future work, we plan to adapt our method to models with different rule extraction algorithms, such as Hiero and forest-based translation (Mi and Huang, 2008).

## Acknowledgements

We thank the anonymous reviewers for comments. The first author is fully supported by the Carnegie Trust and receives additional support from the Cambridge Trusts. Yue Zhang is supported by SUTD under the grant SRG ISTD 2012-038. Philip Williams and Philipp Koehn are supported under EU-FP7-287658 (EU BRIDGE).

## References

- S.F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. ACL*, pages 310–318.
- David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.
- S. Clark and J.R. Curran. 2004. The importance of supertagging for wide-coverage ccg parsing. In *Proc. COLING*, page 282.
- Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*, pages 1–8.
- J.R. Curran and S. Clark. 2003. Investigating gis and smoothing for maximum entropy taggers. In *Proc. EACL*, pages 91–98.
- John DeNero, Mohit Bansal, Adam Pauls, and Dan Klein. 2009. Efficient parsing for transducer grammars. In *Proc. NAACL-HLT*, pages 227–235.
- M. Galley, M. Hopkins, K. Knight, and D. Marcu. 2004. What’s in a translation rule. In *Proc. HLT-NAACL*, pages 273–280.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. COLING and ACL*, pages 961–968.
- Kenneth Heafield, Philipp Koehn, and Alon Lavie. 2013. Grouping language model boundary words to speed k-best extraction from hypergraphs. In *Proc. NAACL*.
- Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proc. EMNLP*, pages 646–655, October.
- L. Huang and D. Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. ACL*, volume 45, page 144.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. NAACL-HLT*, pages 48–54.
- 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 *Proc. ACL Demo Sessions*, pages 177–180.
- P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, volume 5.
- H. Mi and L. Huang. 2008. Forest-based translation rule extraction. In *Proc. EMNLP*, pages 206–214.
- F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proc. ACL*, pages 440–447, Hongkong, China, October.
- F.J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.
- K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.
- S. Petrov, A. Haghghi, and D. Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proc. ACL*, pages 108–116.
- A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. EMNLP*, volume 1, pages 133–142.
- Brian Roark and Kristy Hollingshead. 2008. Classifying chart cells for quadratic complexity context-free inference. In *Proc. COLING*, pages 745–751.
- Brian Roark and Kristy Hollingshead. 2009. Linear complexity context-free parsing pipelines via chart constraints. In *Proc. NAACL*, pages 647–655.
- A. Stolcke et al. 2002. Srlm-an extensible language modeling toolkit. In *Proc. ICSLP*, volume 2, pages 901–904.
- Hao Zhang and Daniel Gildea. 2008. Efficient multi-pass decoding for synchronous context free grammars. In *Proc. ACL*.
- Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proc. NAACL*, pages 256–263.
- Y. Zhang, B.G. Ahn, S. Clark, C. Van Wyk, J.R. Curran, and L. Rimell. 2010. Chart pruning for fast lexicalised-grammar parsing. In *Proc. COLING*, pages 1471–1479.

# A Novel Graph-based Compact Representation of Word Alignment

Qun Liu<sup>†‡</sup>

Zhaopeng Tu<sup>‡</sup>

Shouxun Lin<sup>‡</sup>

<sup>†</sup>Centre for Next Generation Localisation  
Dublin City University  
qliu@computing.dcu.ie

<sup>‡</sup>Key Lab. of Intelligent Info. Processing  
Institute of Computing Technology, CAS  
{tuzhaopeng, sxlin}@ict.ac.cn

## Abstract

In this paper, we propose a novel compact representation called *weighted bipartite hypergraph* to exploit the fertility model, which plays a critical role in word alignment. However, estimating the probabilities of rules extracted from hypergraphs is an NP-complete problem, which is computationally infeasible. Therefore, we propose a divide-and-conquer strategy by decomposing a hypergraph into a set of independent subhypergraphs. The experiments show that our approach outperforms both 1-best and  $n$ -best alignments.

## 1 Introduction

Word alignment is the task of identifying translational relations between words in parallel corpora, in which a word at one language is usually translated into several words at the other language (*fertility model*) (Brown et al., 1993). Given that many-to-many links are common in natural languages (Moore, 2005), it is necessary to pay attention to the relations among alignment links.

In this paper, we have proposed a novel graph-based compact representation of word alignment, which takes into account the joint distribution of alignment links. We first transform each alignment to a bigraph that can be decomposed into a set of subgraphs, where all interrelated links are in the same subgraph (§ 2.1). Then we employ a weighted partite hypergraph to encode multiple bigraphs (§ 2.2).

The main challenge of this research is to efficiently calculate the fractional counts for rules extracted from hypergraphs. This is equivalent to the decision version of set covering problem, which is NP-complete. Observing that most alignments are not connected, we propose a divide-and-conquer strategy by decomposing a hypergraph into a set

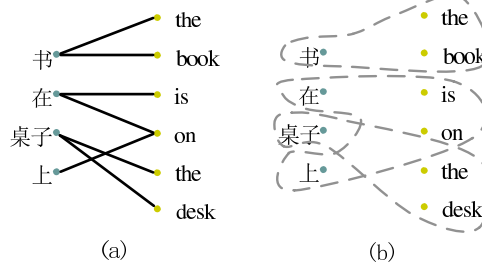


Figure 1: A bigraph constructed from an alignment (a), and its disjoint MCSs (b).

of independent subhypergraphs, which is computationally feasible in practice (§ 3.2). Experimental results show that our approach significantly improves translation performance by up to 1.3 BLEU points over 1-best alignments (§ 4.3).

## 2 Graph-based Compact Representation

### 2.1 Word Alignment as a Bigraph

Each alignment of a sentence pair can be transformed to a bigraph, in which the two disjoint vertex sets  $S$  and  $T$  are the source and target words respectively, and the edges are word-by-word links. For example, Figure 1(a) shows the corresponding bigraph of an alignment.

The bigraph usually is not connected. A graph is called connected if there is a path between every pair of distinct vertices. In an alignment, words in a specific portion at the source side (i.e. a verb phrase) usually align to those in the corresponding portion (i.e. the verb phrase at the target side), and would never align to other words; and vice versa. Therefore, there is no edge that connects the words in the portion to those outside the portion.

Therefore, a bigraph can be decomposed into a unique set of *minimum connected subgraphs* (MCSs), where each subgraph is connected and does not contain any other MCSs. For example, the bigraph in Figure 1(a) can be decomposed into

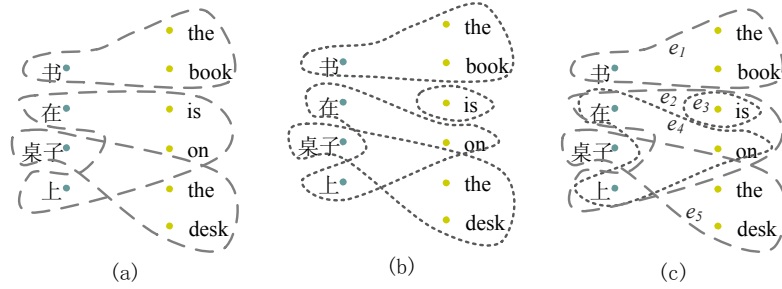


Figure 2: (a) One alignment of a sentence pair; (b) another alignment of the same sentence pair; (c) the resulting hypergraph that takes the two alignments as samples.

the MCSs in Figure 1(b). We can see that all interrelated links are in the same MCS. These MCSs work as fundamental units in our approach to take advantage of the relations among the links. Hereinafter, we use bigraph to denote the alignment of a sentence pair.

## 2.2 Weighted Bipartite Hypergraph

We believe that offering more alternatives to extracting translation rules could help improve translation quality. We propose a new structure called *weighted bipartite hypergraph* that compactly encodes multiple alignments.

We use an example to illustrate our idea. Figures 2(a) and 2(b) show two bigraphs of the same sentence pair. Intuitively, we can encode the union set of subgraphs in a bipartite hypergraph, in which each MCS serves as a hyperedge, as in Figure 2(c). Accordingly, we can calculate how well a hyperedge is by calculating its relative frequency, which is the probability sum of bigraphs in which the corresponding MCS occurs divided by the probability sum of all possible bigraphs. Suppose that the probabilities of the two bigraphs in Figures 2(a) and 2(b) are 0.7 and 0.3, respectively. Then the weight of  $e_1$  is 1.0 and  $e_2$  is 0.7. Therefore, each hyperedge is associated with a weight to indicate how well it is.

Formally, a *weighted bipartite hypergraph*  $H$  is a triple  $\langle S, T, E \rangle$  where  $S$  and  $T$  are two sets of vertices on the source and target sides, and  $E$  are hyperedges associated with weights. Currently, we estimate the weights of hyperedges from an  $n$ -best list by calculating relative frequencies:

$$w(e_i) = \frac{\sum_{BG \in \mathcal{N}} p(BG) \times \delta(BG, g_i)}{\sum_{BG \in \mathcal{N}} p(BG)}$$

Here  $\mathcal{N}$  is an  $n$ -best bigraph (i.e., alignment) list,

$p(BG)$  is the probability of a bigraph  $BG$  in the  $n$ -best list,  $g_i$  is the MCS that corresponds to  $e_i$ , and  $\delta(BG, g_i)$  is an indicator function which equals 1 when  $g_i$  occurs in  $BG$ , and 0 otherwise.

It is worthy mentioning that a hypergraph encodes much more alignments than the input  $n$ -best list. For example, we can construct a new alignment by using hyperedges from different bigraphs that cover all vertices.

## 3 Graph-based Rule Extraction

In this section we describe how to extract translation rules from a hypergraph (§ 3.1) and how to estimate their probabilities (§ 3.2).

### 3.1 Extraction Algorithm

We extract translation rules from a hypergraph for the hierarchical phrase-based system (Chiang, 2007). Chiang (2007) describes a rule extraction algorithm that involves two steps: (1) extract phrases from 1-best alignments; (2) obtain variable rules by replacing sub-phrase pairs with non-terminals. Our extraction algorithm differs at the first step, in which we extract phrases from hypergraphs instead of 1-best alignments. Rather than restricting ourselves by the alignment consistency in the traditional algorithm, we extract all possible candidate target phrases for each source phrase. To maintain a reasonable rule table size, we filter out less promising candidates that have a *fractional count* lower than a threshold.

### 3.2 Calculating Fractional Counts

The *fractional count* of a phrase pair is the probability sum of the alignments with which the phrase pair is consistent (§3.2.2), divided by the probability sum of all alignments encoded in a hypergraph (§3.2.1) (Liu et al., 2009).

Intuitively, our approach faces two challenges:

1. How to calculate the probability sum of all alignments encoded in a hypergraph (§3.2.1)?
2. How to efficiently calculate the probability sum of all consistent alignments for each phrase pair (§3.2.2)?

### 3.2.1 Enumerating All Alignments

In theory, a hypergraph can encode all possible alignments if there are enough hyperedges. However, since a hypergraph is constructed from an  $n$ -best list, it can only represent partial space of all alignments ( $p(A|H) < 1$ ) because of the limiting size of hyperedges learned from the list. Therefore, we need to enumerate all possible alignments in a hypergraph to obtain the probability sum  $p(A|H)$ .

Specifically, generating an alignment from a hypergraph can be modelled as finding a *complete hyperedge matching*, which is a set of hyperedges without common vertices that matches all vertices. The probability of the alignment is the product of hyperedge weights. Thus, enumerating all possible alignments in a hypergraph is reformulated as finding all *complete hypergraph matchings*, which is an NP-complete problem (Valiant, 1979).

Similar to the bigraph, a hypergraph is also usually not connected. To make the enumeration practically tractable, we propose a *divide-and-conquer* strategy by decomposing a hypergraph  $H$  into a set of independent subhypergraphs  $\{h_1, h_2, \dots, h_n\}$ . Intuitively, the probability of an alignment is the product of hyperedge weights. According to the divide-and-conquer strategy, the probability sum of all alignments  $A$  encoded in a hypergraph  $H$  is:

$$p(A|H) = \prod_{h_i \in H} p(A_i|h_i)$$

Here  $p(A_i|h_i)$  is the probability sum of all sub-alignments  $A_i$  encoded in the subhypergraph  $h_i$ .

### 3.2.2 Enumerating Consistent Alignments

Since a hypergraph encodes many alignments, it is unrealistic to enumerate all consistent alignments explicitly for each phrase pair.

Recall that a hypergraph can be decomposed to a list of independent subhypergraphs, and an alignment is a combination of the sub-alignments from the decompositions. We observe that a phrase pair is absolutely consistent with the sub-alignments from some subhypergraphs, while possibly consistent with the others. As an example,

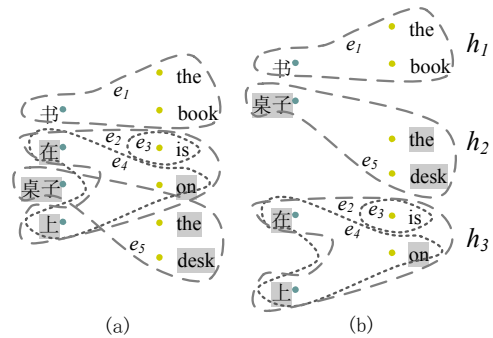


Figure 3: A hypergraph with a candidate phrase in the grey shadow (a), and its independent subhypergraphs  $\{h_1, h_2, h_3\}$ .

consider the phrase pair in the grey shadow in Figure 3(a), it is consistent with all sub-alignments from both  $h_1$  and  $h_2$  because they are outside and inside the phrase pair respectively, while not consistent with the sub-alignment that contains hyperedge  $e_2$  from  $h_3$  because it contains an alignment link that crosses the phrase pair.

Therefore, to calculate the probability sum of all consistent alignments, we only need to consider the *overlap subhypergraphs*, which have at least one hyperedge that crosses the phrase pair. Given a overlap subhypergraph, the probability sum of consistent sub-alignments is calculated by subtracting the probability sum of the sub-alignments that contain crossed hyperedges, from the probability sum of all sub-alignments encoded in a hypergraph.

Given a phrase pair  $P$ , let  $OS$  and  $NS$  denotes the sets of overlap and non-overlap subhypergraphs respectively ( $NS = H - OS$ ). Then

$$p(A|H, P) = \prod_{h_i \in OS} p(A_i|h_i, P) \prod_{h_j \in NS} p(A_j|h_j)$$

Here the phrase pair is absolutely consistent with the sub-alignments from non-overlap subhypergraphs ( $NS$ ), and we have  $p(A|h, P) = p(A|h)$ . Then the fractional count of a phrase pair is:

$$c(P|H) = \frac{p(A|H, P)}{p(A|H)} = \frac{\prod_{h_i \in OS} p(A|h_i, P)}{\prod_{h_i \in OS} p(A|h_i)}$$

After we get the fractional counts of translation rules, we can estimate their *relative frequencies* (Och and Ney, 2004). We follow (Liu et al., 2009; Tu et al., 2011) to learn lexical tables from  $n$ -best lists and then calculate the lexical weights.



| Rules from... | Rules | MT03  | MT04  | MT05  | Avg.  |
|---------------|-------|-------|-------|-------|-------|
| 1-best        | 257M  | 33.45 | 35.25 | 33.63 | 34.11 |
| 10-best       | 427M  | 34.10 | 35.71 | 34.04 | 34.62 |
| Hypergraph    | 426M  | 34.71 | 36.24 | 34.41 | 35.12 |

Table 1: Evaluation of translation quality.

## 4 Experiments

### 4.1 Setup

We carry out our experiments on Chinese-English translation tasks using a reimplement of the hierarchical phrase-based system (Chiang, 2007). Our training data contains 1.5 million sentence pairs from LDC dataset.<sup>1</sup> We train a 4-gram language model on the Xinhua portion of the GIGAWORD corpus using the SRI Language Toolkit (Stolcke, 2002) with modified Kneser-Ney Smoothing (Kneser and Ney, 1995). We use minimum error rate training (Och, 2003) to optimize the feature weights on the MT02 testset, and test on the MT03/04/05 testsets. For evaluation, case-insensitive NIST BLEU (Papineni et al., 2002) is used to measure translation performance.

We first follow Venugopal et al. (2008) to produce  $n$ -best lists via GIZA++. We produce 10-best lists in two translation directions, and use “grow-diag-final-and” strategy (Koehn et al., 2003) to generate the final  $n$ -best lists by selecting the top  $n$  alignments. We re-estimated the probability of each alignment in the  $n$ -best list using re-normalization (Venugopal et al., 2008). Finally we construct weighted alignment hypergraphs from these  $n$ -best lists.<sup>2</sup> When extracting rules from hypergraphs, we set the pruning threshold  $t = 0.5$ .

### 4.2 Tractability of Divide-and-Conquer Strategy

Figure 4 shows the distribution of vertices (hyperedges) number of the subhypergraphs. We can see that most of the subhypergraphs have just less than two vertices and hyperedges.<sup>3</sup> Specifically, each subhypergraph has 2.0 vertices and 1.4 hy-

<sup>1</sup>The corpus includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

<sup>2</sup>Here we only use 10-best lists, because the alignments beyond top 10 have very small probabilities, thus have negligible influence on the hypergraphs.

<sup>3</sup>It’s interesting that there are few subhypergraphs that have exactly 2 hyperedges. In this case, the only two hyperedges fully cover the vertices and they differ at the word-by-word links, which is uncommon in  $n$ -best lists.

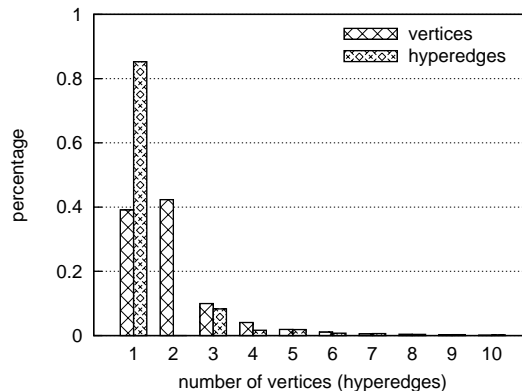


Figure 4: The distribution of vertices (hyperedges) number of the subhypergraphs.

peredges on average. This suggests that the divide-and-conquer strategy makes the extraction computationally tractable, because it greatly reduces the number of vertices and hyperedges. For computational tractability, we only allow a subhypergraph has at most 5 hyperedges.<sup>4</sup>

### 4.3 Translation Performance

Table 1 shows the rule table size and translation quality. Using  $n$ -best lists slightly improves the BLEU score over 1-best alignments, but at the cost of a larger rule table. This is in accord with intuition, because all possible translation rules would be extracted from different alignments in  $n$ -best lists without pruning. This larger rule table indeed leads to a high rule coverage, but in the meanwhile, introduces translation errors because of the low-quality rules (i.e., rules extracted only from low-quality alignments in  $n$ -best lists). By contrast, our approach not only significantly improves the translation performance over 1-best alignments, but also outperforms  $n$ -best lists with a similar-scale rule table. The absolute improvements of 1.0 BLEU points on average over 1-best alignments are statistically significant at  $p < 0.01$  using *sign-test* (Collins et al., 2005).

<sup>4</sup>If a subhypergraph has more than 5 hyperedges, we forcibly partition it into small subhypergraphs by iteratively removing lowest-probability hyperedges.

| Rules from . . . | Shared |       | Non-shared |       | All   |       |
|------------------|--------|-------|------------|-------|-------|-------|
|                  | Rules  | BLEU  | Rules      | BLEU  | Rules | BLEU  |
| 10-best          | 1.83M  | 32.75 | 2.81M      | 30.71 | 4.64M | 34.62 |
| Hypergraph       | 1.83M  | 33.24 | 2.89M      | 31.12 | 4.72M | 35.12 |

Table 2: Comparison of rule tables learned from  $n$ -best lists and hypergraphs. “All” denotes the full rule table, “Shared” denotes the intersection of two tables, and “Non-shared” denotes the complement. Note that the probabilities of “Shared” rules are different for the two approaches.

Why our approach outperforms  $n$ -best lists? In theory, the rule table extracted from  $n$ -best lists is a subset of that from hypergraphs. In practice, however, this is not true because we pruned the rules that have fractional counts lower than a threshold. Therefore, the question arises as to how many rules are shared by  $n$ -best and hypergraph-based extractions. We try to answer this question by comparing the different rule tables (filtered on the test sets) learned from  $n$ -best lists and hypergraphs. Table 2 gives some statistics. “All” denotes the full rule table, “Shared” denotes the intersection of two tables, and “Non-shared” denotes the complement. Note that the probabilities of “Shared” rules are different for the two approaches. We can see that both the “Shared” and “Non-shared” rules learned from hypergraphs outperform  $n$ -best lists, indicating: (1) our approach has a better estimation of rule probabilities because we estimate the probabilities from a much larger alignment space that can not be represented by  $n$ -best lists, (2) our approach can extract good rules that cannot be extracted from any single alignments in the  $n$ -best lists.

## 5 Related Work

Our research builds on previous work in the field of graph models and compact representations. Graph models have been used before in word alignment: the search space of word alignment can be structured as a graph and the search problem can be reformulated as finding the optimal path through this graph (e.g., (Och and Ney, 2004; Liu et al., 2010)). In addition, Kumar and Byrne (2002) define a graph distance as a loss function for minimum Bayes-risk word alignment, Riesa and Marcu (2010) open up the word alignment task to advances in hypergraph algorithms currently used in parsing. As opposed to the search problem, we propose a graph-based compact representation that encodes multiple alignments for machine translation.

Previous research has demonstrated that compact representations can produce improved results by offering more alternatives, e.g., using forests over 1-best trees (Mi and Huang, 2008; Tu et al., 2010; Tu et al., 2012a), word lattices over 1-best segmentations (Dyer et al., 2008), and weighted alignment matrices over 1-best word alignments (Liu et al., 2009; Tu et al., 2011; Tu et al., 2012b). Liu et al., (2009) estimate the link probabilities from  $n$ -best lists, while Gispert et al., (2010) learn the alignment posterior probabilities directly from IBM models. However, both of them ignore the relations among alignment links. By contrast, our approach takes into account the joint distribution of alignment links and explores the fertility model past the link level.

## 6 Conclusion

We have presented a novel compact representation of word alignment, named weighted bipartite hypergraph, to exploit the relations among alignment links. Since estimating the probabilities of rules extracted from hypergraphs is an NP-complete problem, we propose a computationally tractable divide-and-conquer strategy by decomposing a hypergraph into a set of independent subhypergraphs. Experimental results show that our approach outperforms both 1-best and  $n$ -best alignments.

## Acknowledgement

The authors are supported by 863 State Key Project No. 2011AA01A207, National Key Technology R&D Program No. 2012BAH39B03 and National Natural Science Foundation of China (Contracts 61202216). 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 thank Junhui Li, Yifan He and the anonymous reviewers for their insightful comments.

## References

- Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of Association for Computational Linguistics*, pages 531–540.
- Adrià de Gispert, Juan Pino, and William Byrne. 2010. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 545–554.
- Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184.
- Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54.
- Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 140–147.
- Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1017–1026.
- Yang Liu, Qun Liu, and Shouxun Lin. 2010. Discriminative word alignment by linear modeling. *Computational Linguistics*, 36(3):303–339.
- Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214.
- Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 81–88, October.
- Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.
- 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.
- Jason Riesa and Daniel Marcu. 2010. Hierarchical search for word alignment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 157–166.
- Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of Seventh International Conference on Spoken Language Processing*, volume 3, pages 901–904. Citeseer.
- Zhaopeng Tu, Yang Liu, Young-Sook Hwang, Qun Liu, and Shouxun Lin. 2010. Dependency forest for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1092–1100.
- Zhaopeng Tu, Yang Liu, Qun Liu, and Shouxun Lin. 2011. Extracting hierarchical rules from a weighted alignment matrix. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1294–1303.
- Zhaopeng Tu, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2012a. Dependency forest for sentiment analysis. In *Springer-Verlag Berlin Heidelberg*, pages 69–77.
- Zhaopeng Tu, Yang Liu, Yifan He, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012b. Combining multiple alignments to improve machine translation. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1249–1260.
- Leslie G Valiant. 1979. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201.
- Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: n-best alignments and parses in mt training. In *Proceedings of AMTA*, pages 192–201.

# Stem Translation with Affix-Based Rule Selection for Agglutinative Languages

Zhiyang Wang<sup>†</sup>, Yajuan Lü<sup>†</sup>, Meng Sun<sup>†</sup>, Qun Liu<sup>††</sup>

<sup>†</sup>Key Laboratory of Intelligent Information Processing  
Institute of Computing Technology, Chinese Academy of Sciences  
P.O. Box 2704, Beijing 100190, China  
{wangzhiyang, lvyajuan, sunmeng, liuqun}@ict.ac.cn  
<sup>††</sup>Centre for Next Generation Localisation  
Faculty of Engineering and Computing, Dublin City University  
qliu@computing.dcu.ie

## Abstract

Current translation models are mainly designed for languages with limited morphology, which are not readily applicable to agglutinative languages as the difference in the way lexical forms are generated. In this paper, we propose a novel approach for translating agglutinative languages by treating stems and affixes differently. We employ stem as the atomic translation unit to alleviate data sparseness. In addition, we associate each stem-granularity translation rule with a distribution of related affixes, and select desirable rules according to the similarity of their affix distributions with given spans to be translated. Experimental results show that our approach significantly improves the translation performance on tasks of translating from three Turkic languages to Chinese.

## 1 Introduction

Currently, most methods on statistical machine translation (SMT) are developed for translation of languages with limited morphology (e.g., English, Chinese). They assumed that word was the atomic translation unit (ATU), always ignoring the internal morphological structure of word. This assumption can be traced back to the original IBM word-based models (Brown et al., 1993) and several significantly improved models, including phrase-based (Och and Ney, 2004; Koehn et al., 2003), hierarchical (Chiang, 2005) and syntactic (Quirk et al., 2005; Galley et al., 2006; Liu et al., 2006) models. These improved models worked well for translating languages like English with large scale parallel corpora available.

Different from languages with limited morphology, words of agglutinative languages are formed mainly by concatenation of stems and affixes. Generally, a stem can attach with several affixes, thus leading to tens of hundreds of possible inflected variants of lexicons for a single stem. Modeling each lexical form as a separate word will generate high out-of-vocabulary rate for SMT. Theoretically, ways like morphological analysis and increasing bilingual corpora could alleviate the problem of data sparsity, but most agglutinative languages are less-studied and suffer from the problem of resource-scarceness. Therefore, previous research mainly focused on the different inflected variants of the same stem and made various transformation of input by morphological analysis, such as (Lee, 2004; Goldwater and McClosky, 2005; Yang and Kirchhoff, 2006; Habash and Sadat, 2006; Bisazza and Federico, 2009; Wang et al., 2011). These work still assume that the atomic translation unit is word, stem or morpheme, without considering the difference between stems and affixes.

In agglutinative languages, stem is the base part of word not including inflectional affixes. Affix, especially inflectional affix, indicates different grammatical categories such as tense, person, number and case, etc., which is useful for translation rule disambiguation. Therefore, we employ stem as the atomic translation unit and use affix information to guide translation rule selection. Stem-granularity translation rules have much larger coverage and can lower the OOV rate. Affix based rule selection takes advantage of auxiliary syntactic roles of affixes to make a better rule selection. In this way, we can achieve a balance between rule coverage and matching accuracy, and ultimately improve the translation performance.

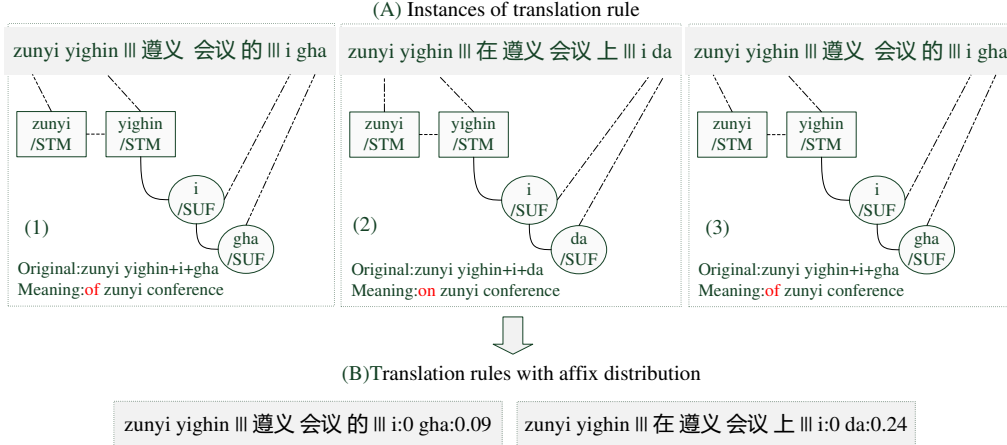


Figure 1: Translation rule extraction from Uyghur to Chinese. Here tag “/STM” represents stem and “/SUF” means suffix.

## 2 Affix Based Rule Selection Model

Figure 1 (B) shows two translation rules along with affix distributions. Here a translation rule contains three parts: the source part (on stem level), the target part, and the related affix distribution (represented as a vector). We can see that, although the source part of the two translation rules are identical, their affix distributions are quite different. Affix “gha” in the first rule indicates that something is affiliated to a subject, similar to “of” in English. And “da” in second rule implies location information. Therefore, given a span “zunyi/STM yighin/STM+i/SUF+da/SUF+...” to be translated, we hope to encourage our model to select the second translation rule. We can achieve this by calculating similarity between the affix distributions of the translation rule and the span.

The affix distribution can be obtained by keeping the related affixes for each rule instance during translation rule extraction ((A) in Figure 1). After extracting and scoring stem-granularity rules in a traditional way, we extract stem-granularity rules again by keeping affix information and compute the affix distribution with tf-idf (Salton and Buckley, 1987). Finally, the affix distribution will be added to the previous stem-granularity rules.

### 2.1 Affix Distribution Estimation

Formally, translation rule instances with the same source part can be treated as a *document collection*<sup>1</sup>, so each rule instance in the collection is

<sup>1</sup>We employ concepts from text classification to illustrate how to estimate affix distribution.

some kind of *document*. Our goal is to classify the source parts into the target parts on the *document collection* level with the help of affix distribution. Accordingly, we employ vector space model (VSM) to represent affix distribution of each rule instance. In this model, the feature weights are represented by the classic tf-idf (Salton and Buckley, 1987):

$$\text{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad \text{idf}_{i,j} = \log \frac{|\mathbf{D}|}{|\mathbf{j} : \mathbf{a}_i \in \mathbf{r}_j|} \quad (1)$$

$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \times \text{idf}_{i,j}$$

where  $\text{tfidf}_{i,j}$  is the weight of affix  $a_i$  in translation rule instance  $r_j$ .  $n_{i,j}$  indicates the number of occurrence of affix  $a_i$  in  $r_j$ .  $|\mathbf{D}|$  is the number of rule instance with the same source part, and  $|\mathbf{j} : \mathbf{a}_i \in \mathbf{r}_j|$  is the number of rule instance which contains affix  $a_i$  within  $|\mathbf{D}|$ .

Let’s take the suffix “gha” from ( $A_1$ ) in Figure 1 as an example. We assume that there are only three instances of translation rules extracted from parallel corpus ((A) in Figure 1). We can see that “gha” only appear once in ( $A_1$ ) and also appear once in whole instances. Therefore,  $\text{tf}_{\text{gha},(A_1)}$  is 0.5 and  $\text{idf}_{\text{gha},(A_1)}$  is  $\log(3/2)$ .  $\text{tfidf}_{\text{gha},(A_1)}$  is the product of  $\text{tf}_{\text{gha},(A_1)}$  and  $\text{idf}_{\text{gha},(A_1)}$  which is 0.09.

Given a set of  $\mathbf{N}$  translation rule instances with the same source and target part, we define the centroid vector  $\mathbf{d}_r$  according to the centroid-based classification algorithm (Han and Karypis, 2000),

$$\mathbf{d}_r = \frac{1}{\mathbf{N}} \sum_{i \in \mathbf{N}} \mathbf{d}_i \quad (2)$$

| Data set     | #Sent. | #Type |      |       | #Token |      |       |
|--------------|--------|-------|------|-------|--------|------|-------|
|              |        | word  | stem | morph | word   | stem | morph |
| UY-CH-Train. | 50K    | 69K   | 39K  | 42K   | 1.2M   | 1.2M | 1.6M  |
| UY-CH-Dev.   | 0.7K*4 | 5.9K  | 4.1K | 4.6K  | 18K    | 18K  | 23.5K |
| UY-CH-Test.  | 0.7K*1 | 4.7K  | 3.3K | 3.8K  | 14K    | 14K  | 17.8K |
| KA-CH-Train. | 50K    | 62K   | 40K  | 42K   | 1.1M   | 1.1M | 1.3M  |
| KA-CH-Dev.   | 0.7K*4 | 5.3K  | 4.2K | 4.5K  | 15K    | 15K  | 18K   |
| KA-CH-Test.  | 0.2K*1 | 2.6K  | 2.0K | 2.3K  | 8.6K   | 8.6K | 10.8K |
| KI-CH-Train. | 50K    | 53K   | 27K  | 31K   | 1.2M   | 1.2M | 1.5M  |
| KI-CH-Dev.   | 0.5K*4 | 4.1K  | 3.1K | 3.5K  | 12K    | 12K  | 15K   |
| KI-CH-Test.  | 0.2K*4 | 2.2K  | 1.8K | 2.1K  | 4.7K   | 4.7K | 5.8K  |

Table 1: Statistics of data sets. \* $N$  means the number of reference, *morph* is short to morpheme. UY, KA, KI, CH represent Uyghur, Kazakh, Kirghiz and Chinese respectively.

$\mathbf{d}_r$  is the final affix distribution.

By comparing the similarity of affix distributions, we are able to decide whether a translation rule is suitable for a span to be translated. In this work, similarity is measured using the cosine distance similarity metric, given by

$$\text{sim}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{\|\mathbf{d}_1\| \times \|\mathbf{d}_2\|} \quad (3)$$

where  $\mathbf{d}_i$  corresponds to a vector indicating affix distribution, and “ $\cdot$ ” denotes the inner product of the two vectors.

Therefore, for a specific span to be translated, we first analyze it to get the corresponding stem sequence and related affix distribution represented as a vector. Then the stem sequence is used to search the translation rule table. If the source part is matched, the similarity will be calculated for each candidate translation rule by cosine similarity (as in equation 3). Therefore, in addition to the traditional translation features on stem level, our model also adds the affix similarity score as a dynamic feature into the log-linear model (Och and Ney, 2002).

### 3 Related Work

Most previous work on agglutinative language translation mainly focus on Turkish and Finnish. Bisazza and Federico (2009) and Mermer and Saraclar (2011) optimized morphological analysis as a pre-processing step to improve the translation between Turkish and English. Yeniterzi and Oflazer (2010) mapped the syntax of the English side to the morphology of the Turkish side with the factored model (Koehn and Hoang, 2007). Yang

and Kirchhoff (2006) backed off surface form to stem when translating OOV words of Finnish. Luong and Kan (2010) and Luong et al. (2010) focused on Finnish-English translation through improving word alignment and enhancing phrase table. These works still assumed that the atomic translation unit is word, stem or morpheme, without considering the difference between stems and affixes.

There are also some work that employed the context information to make a better choice of translation rules (Carpuat and Wu, 2007; Chan et al., 2007; He et al., 2008; Cui et al., 2010). all the work employed rich context information, such as POS, syntactic, etc., and experiments were mostly done on less inflectional languages (i.e. Chinese, English) and resourceful languages (i.e. Arabic).

## 4 Experiments

In this work, we conduct our experiments on three different agglutinative languages, including Uyghur, Kazakh and Kirghiz. All of them are derived from Altaic language family, belonging to Turkic languages, and mostly spoken by people in Central Asia. There are about 24 million people take these languages as mother tongue. All of the tasks are derived from the evaluation of China Workshop of Machine Translation (CWMT)<sup>2</sup>. Table 1 shows the statistics of data sets.

For the language model, we use the SRI Language Modeling Toolkit (Stolcke, 2002) to train a 5-gram model with the target side of training corpus. And phrase-based Moses<sup>3</sup> is used as our

<sup>2</sup><http://mt.xmu.edu.cn/cwmt2011/en/index.html>.

<sup>3</sup><http://www.statmt.org/moses/>

|              | UY-CH                       | KA-CH                        | KI-CH                  |
|--------------|-----------------------------|------------------------------|------------------------|
| <b>word</b>  | 31.74 <sub>+0.0</sub>       | 28.64 <sub>+0.0</sub>        | 35.05 <sub>+0.0</sub>  |
| <b>stem</b>  | <b>33.74<sub>+2.0</sub></b> | <b>30.14<sub>+1.5</sub></b>  | 35.52 <sub>+0.47</sub> |
| <b>morph</b> | 32.69 <sub>+0.95</sub>      | 29.21 <sub>+0.57</sub>       | 34.97 <sub>-0.08</sub> |
| <b>affix</b> | <b>34.34<sub>+2.6</sub></b> | <b>30.19<sub>+2.27</sub></b> | 35.96 <sub>+0.91</sub> |

Table 2: Translation results from Turkic languages to Chinese. **word**: ATU is surface form, **stem**: ATU is represented stem, **morph**: ATU denotes morpheme, **affix**: stem translation with affix distribution similarity. BLEU scores in **bold** means significantly better than the baseline according to (Koehn, 2004) for p-value less than 0.01.

baseline SMT system. The decoding weights are optimized with MERT (Och, 2003) to maximum word-level BLEU scores (Papineni et al., 2002).

#### 4.1 Using Unsupervised Morphological Analyzer

As most agglutinative languages are resource-poor, we employ unsupervised learning method to obtain the morphological structure. Following the approach in (Virpioja et al., 2007), we employ the Morfessor<sup>4</sup> Categories-MAP algorithm (Creutz and Lagus, 2005). It applies a hierarchical model with three categories (prefix, stem, and suffix) in an unsupervised way. From Table 1 we can see that vocabulary sizes of the three languages are reduced obviously after unsupervised morphological analysis.

Table 2 shows the translation results. All the three translation tasks achieve obvious improvements with the proposed model, which always performs better than only employ **word**, **stem** and **morph**. For the Uyghur to Chinese translation (UY-CH) task in Table 2, performances after unsupervised morphological analysis are always better than the baseline. And we gain up to +2.6 BLEU points improvements with **affix** compared to the baseline. For the Kazakh to Chinese translation (KA-CH) task, the improvements are also significant. We achieve +2.27 and +0.77 improvements compared to the baseline and **stem**, respectively. As for the Kirghiz to Chinese translation (KI-CH) task, improvements seem relative small compared to the other two language pairs. However, it also gains +0.91 BLEU points over the baseline.

<sup>4</sup><http://www.cis.hut.fi/projects/morpho/>

|              | UY     | Unsup | Sup  |
|--------------|--------|-------|------|
| <b>stem</b>  | #Type  | 39K   | 21K  |
|              | #Token | 1.2M  | 1.2M |
| <b>affix</b> | #Type  | 3.0K  | 0.3K |
|              | #Token | 0.4M  | 0.7M |

Table 3: Statistics of training corpus after unsupervised(Unsup) and supervised(Sup) morphological analysis.

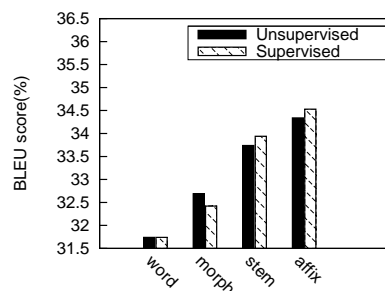


Figure 2: Uyghur to Chinese translation results after unsupervised and supervised analysis.

#### 4.2 Using Supervised Morphological Analyzer

Taking it further, we also want to see the effect of supervised analysis on our model. A generative statistical model of morphological analysis for Uyghur was developed according to (Mairehaba et al., 2012). Table 3 shows the difference of statistics of training corpus after supervised and unsupervised analysis. Supervised method generates fewer type of stems and affixes than the unsupervised approach. As we can see from Figure 2, except for the **morph** method, **stem** and **affix** based approaches perform better after supervised analysis. The results show that our approach can obtain even better translation performance if better morphological analyzers are available. Supervised morphological analysis generates more meaningful morphemes, which lead to better disambiguation of translation rules.

## 5 Conclusions and Future Work

In this paper we propose a novel framework for agglutinative language translation by treating stem and affix differently. We employ the stem sequence as the main part for training and decoding. Besides, we associate each stem-granularity translation rule with an affix distribution, which could be used to make better translation decisions by calculating the affix distribution similarity be-

tween the rule and the instance to be translated. We conduct our model on three different language pairs, all of which substantially improved the translation performance. The procedure is totally language-independent, and we expect that other language pairs could benefit from our approach.

## Acknowledgments

The authors were supported by 863 State Key Project (No. 2011AA01A207), and National Key Technology R&D Program (No. 2012BAH39B03), Key Project of Knowledge Innovation Program of Chinese Academy of Sciences (No. KGZD-EW-501). 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

- Arianna Bisazza and Marcello Federico. 2009. Morphological pre-processing for Turkish to English statistical machine translation. In *Proceedings of IWSLT*, pages 129–135.
- 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.
- Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of EMNLP-CoNLL*, pages 61–72.
- Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of ACL*, pages 33–40.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270.
- Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of AKRR*, pages 106–113.
- Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2010. A joint rule selection model for hierarchical phrase-based translation. In *Proceedings of ACL, Short Papers*, pages 6–11.
- Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING/ACL*, pages 961–968.
- Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of HLT-EMNLP*, pages 676–683.
- Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of NAACL, Short Papers*, pages 49–52.
- Eui-Hong Sam Han and George Karypis. 2000. Centroid-based document classification: analysis experimental results. In *Proceedings of PKDD*, pages 424–431.
- Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Proceedings of COLING*, pages 321–328.
- Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of EMNLP-CoNLL*, pages 868–876.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.
- Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *Proceedings of HLT-NAACL, Short Papers*, pages 57–60.
- Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of COLING-ACL*, pages 609–616.
- Minh-Thang Luong and Min-Yen Kan. 2010. Enhancing morphological alignment for translating highly inflected languages. In *Proceedings of COLING*, pages 743–751.
- Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. 2010. A hybrid morpheme-word representation for machine translation of morphologically rich languages. In *Proceedings of EMNLP*, pages 148–157.
- Aili Mairehaba, Wenbin Jiang, Zhiyang Wang, Yibulayin Tuergen, and Qun Liu. 2012. Directed graph model of Uyghur morphological analysis. *Journal of Software*, 23(12):3115–3129.
- Coskun Mermer and Murat Saraclar. 2011. Unsupervised Turkish morphological segmentation for statistical machine translation. In *Workshop of MT and Morphologically-rich Languages*.



- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, pages 295–302.
- Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, pages 417–449.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal SMT. In *Proceedings of ACL*, pages 271–279.
- Gerard Salton and Chris Buckley. 1987. Term weighting approaches in automatic text retrieval. Technical report.
- Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, pages 311–318.
- Sami Virpioja, Jaakko J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of MT SUMMIT*, pages 491–498.
- Zhiyang Wang, Yajuan Lü, and Qun Liu. 2011. Multi-granularity word alignment and decoding for agglutinative language translation. In *Proceedings of MT SUMMIT*, pages 360–367.
- Mei Yang and Katrin Kirchhoff. 2006. Phrase-based backoff models for machine translation of highly inflected languages. In *Proceedings of EACL*, pages 1017–1020.
- Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *Proceedings of ACL*, pages 454–464.

# A Novel Translation Framework Based on Rhetorical Structure Theory

Mei Tu      Yu Zhou      Chengqing Zong

National Laboratory of Pattern Recognition, Institute of Automation,  
Chinese Academy of Sciences

{mtu, yzhou, cqzong}@nlpr.ia.ac.cn

## Abstract

Rhetorical structure theory (RST) is widely used for discourse understanding, which represents a discourse as a hierarchically semantic structure. In this paper, we propose a novel translation framework with the help of RST. In our framework, the translation process mainly includes three steps: 1) **Source RST-tree acquisition**: a source sentence is parsed into an RST tree; 2) **Rule extraction**: translation rules are extracted from the source tree and the target string via bilingual word alignment; 3) **RST-based translation**: the source RST-tree is translated with translation rules. Experiments on Chinese-to-English show that our RST-based approach achieves improvements of 2.3/0.77/1.43 BLEU points on NIST04/NIST05/CWMT2008 respectively.

## 1 Introduction

For statistical machine translation (SMT), a crucial issue is how to build a translation model to extract as much accurate and generative translation knowledge as possible. The existing SMT models have made much progress. However, they still suffer from the bad performance of unnatural or even unreadable translation, especially when the sentences become complicated. We think the deep reason is that those models only extract translation information on lexical or syntactic level, but fail to give an overall understanding of source sentences on semantic level of discourse. In order to solve such problem, (Gong et al., 2011; Xiao et al., 2011; Wong and Kit, 2012) build discourse-based translation models to ensure the lexical coherence or consistency. Although some lexicons can be translated better by their models, the overall structure still remains unnatural. Marcu et al. (2000) design a discourse structure transferring module, but leave much work to do, especially on how to integrate this module into SMT and how to automatically

analyze the structures. Those reasons urge us to seek a new translation framework under the idea of “translation with overall understanding”.

Rhetorical structure theory (RST) (Mann and Thompson, 1988) provides us with a good perspective and inspiration to build such a framework. Generally, an RST tree can explicitly show the minimal spans with semantic functional integrity, which are called elementary discourse units (*edus*) (Marcu et al., 2000), and it also depicts the hierarchical relations among *edus*. Furthermore, since different languages’ *edus* are usually equivalent on semantic level, it is intuitive to create a new framework based on RST by directly mapping the source *edus* to target ones.

Taking the Chinese-to-English translation as an example, our translation framework works as the following steps:

- 1) **Source RST-tree acquisition**: a source sentence is parsed into an RST-tree;
- 2) **Rule extraction**: translation rules are extracted from the source tree and the target string via bilingual word alignment;
- 3) **RST-based translation**: the source RST-tree is translated into target sentence with extracted translation rules.

Experiments on Chinese-to-English sentence-level discourses demonstrate that this method achieves significant improvements.

## 2 Chinese RST Parser

### 2.1 Annotation of Chinese RST Tree

Similar to (Soricut and Marcu, 2003), a node of RST tree is represented as a tuple  $R-[s, m, e]$ , which means the relation  $R$  controls two semantic spans  $U_1$  and  $U_2$ ,  $U_1$  starts from word position  $s$  and stops at word position  $m$ .  $U_2$  starts from  $m+1$  and ends with  $e$ . Under the guidance of definition of RST, Yue (2008) defined 12 groups<sup>1</sup> of

<sup>1</sup>They are *Parallel, Alternative, Condition, Reason, Elaboration, Means, Preparation, Enablement, Antithesis, Background, Evidences, Others*.

Example 1:

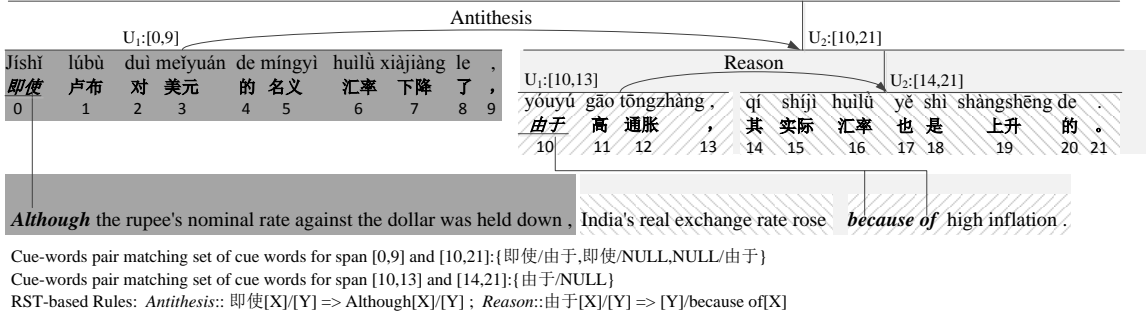


Figure 1: An example of Chinese RST tree and its word alignment of the corresponding English string.

rhetoical relations for Chinese particularly, upon which our Chinese RST parser is developed.

Figure 1 illustrates an example of Chinese RST tree and its alignment to the English string. There are two levels in this tree. The *Antithesis* relation controls  $U_1$  from 0 to 9 and  $U_2$  from 10 to 21. Thus it is written as *Antithesis*-[0,9,21]. Different shadow blocks denote the alignments of different *edus*. Links between source and target words are alignments of cue words. Cue words are viewed as the strongest clues for rhetorical relation recognition and always found at the beginning of text (Reitter, 2003), such as “即使(although), 由于(because of)”. With the cue words included, the relations are much easier to be analyzed. So we focus on the explicit relations with cue words in this paper as our first try.

## 2.2 Bayesian Method for Chinese RST Parser

For Chinese RST parser, there are two tasks. One is the segmentation of *edu* and the other is the relation tagging between two semantic spans.

| Feature    | Meaning                                     |
|------------|---|
| $F_1(F_6)$ | left(right) child is a syntactic sub-tree?  |
| $F_2(F_5)$ | left(right) child ends with a punctuation?  |
| $F_3(F_4)$ | cue words of left (right) child.            |
| $F_7$      | left and right children are sibling nodes?  |
| $F_8(F_9)$ | syntactic head symbol of left(right) child. |

Table 1: 9 features used in our Bayesian model

Inspired by the features used in English RST parser (Soricut and Marcu, 2003; Reitter, 2003; Duverle and Prendinger, 2009; Hernault et al., 2010a), we design a Bayesian model to build a joint parser for segmentation and tagging simultaneously. In this model, 9 features in Table 1 are used. In the table, punctuations include comma, semicolons, period and question mark. We view explicit connectives as cue words in this paper.

Figure 2 illustrates the conditional independences of 9 features which are denoted with  $F_1 \sim F_9$ .

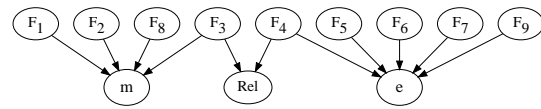


Figure 2: The graph for conditional independences of 9 features.

The segmentation and parsing conditional probabilities are computed as follows:

$$P(m|F_1^9) = P(m|F_1^3, F_8) \quad (1)$$

$$P(e|F_1^9) = P(e|F_4^7, F_9) \quad (2)$$

$$P(Rel|F_1^9) = P(Rel|F_3^4) \quad (3)$$

where  $F_n$  represents the  $n^{th}$  feature,  $F_n^l$  means features from  $n$  to  $l$ . *Rel* is short for relation. (1) and (2) describe the conditional probabilities of  $m$  and  $e$ . When using Formula (3) to predict the relation, we search all the cue-words pair, as shown in Figure 1, to get the best match. When training, we use maximum likelihood estimation to get all the associated probabilities. For decoding, the pseudo codes are given as below.

```

1: Nodes={[]}
2: Parser(0,End)
3: Parser(s,e): // recursive parser function
4: if s > e or e is -1: return -1;
5: m = GetMaxM(s,e) //compute m through Formula(1);if no cue words found, then m=-1;
6: e' = GetMaxE(s,m,e) //compute e' through F (2);
7: if m or e' equals to -1: return -1;
8: Rel=GetRelation(s,m,e') //compute relation by F (3)
9: push [Rel,s,m,e'] into Nodes
10: Parser(s,m)
11: Parser(m+1,e')
12: Parser(e'+1,e)
13: Rel=GetRelation(s,e',e)
14: push [Rel,s,e',e] into Nodes
15: return e

```

For example in Figure 1, for the first iteration,  $s=0$  and  $m$  will be chosen from  $\{1-20\}$ . We get  $m=9$  through Formula (1). Then, similar with  $m$ , we get  $e=21$  through Formula (2). Finally, the relation is figured out by Formula (3). Thus, a node is generated. A complete RST tree constructs until the end of the iterative process for this sentence. This method can run fast due to the simple greedy algorithm. It is plausible in our cases, because we only have a small scale of manually-annotated Chinese RST corpus, which prefers simple rather than complicated models.

### 3 Translation Model

#### 3.1 Rule Extraction

As shown in Figure 1, the RST tree-to-string alignment provides us with two types of translation rules. One is common phrase-based rules, which are just like those in phrase-based model (Koehn et al., 2003). The other is RST tree-to-string rule, and it's defined as,

$$\begin{aligned} relation :: U_1(\alpha, X)/U_2(\gamma, Y) \\ \Rightarrow U_1(tr(\alpha), tr(X)) \sim U_2(tr(\gamma), tr(Y)) \end{aligned}$$

where the terminal characters  $\alpha$  and  $\gamma$  represent the cue words which are optimum match for maximizing Formula (3). While the non-terminals  $X$  and  $Y$  represent the rest of the sequence. Function  $tr(\cdot)$  means the translation of  $\cdot$ . The operator  $\sim$  is an operator to indicate that the order of  $tr(U_1)$  and  $tr(U_2)$  is monotone or reverse. During rules' extraction, if the mean position of all the words in  $tr(U_1)$  precedes that in  $tr(U_2)$ ,  $\sim$  is monotone. Otherwise,  $\sim$  is reverse.

For example in Figure 1, the *Reason* relation controls  $U_1:[10,13]$  and  $U_2:[14,21]$ . Because the mean position of  $tr(U_2)$  is before that of  $tr(U_1)$ , the reverse order is selected. We list the RST-based rules for Example 1 in Figure 1.

#### 3.2 Probabilities Estimation

For the phrase-based translation rules, we use four common probabilities and the probabilities' estimation is the same with those in (Koehn et al., 2003). While the probabilities of RST-based translation rules are given as follows,

(1)  $P(r_e|r_f, Rel) = \frac{Count(r_e, r_f, relation)}{Count(r_f, relation)}$ : where  $r_e$  is the target side of the rule, ignorance of the order, i.e.  $U_1(tr(\alpha), tr(X)) \sim U_2(tr(\gamma), tr(Y))$  with two directions,  $r_f$  is the source side, i.e.  $U_1(\alpha, X)/U_2(\gamma, Y)$ , and *Rel* means the relation type.

(2)  $P(\tau|r_e, r_f, Rel) = \frac{Count(\tau, r_e, r_f, relation)}{Count(r_e, r_f, relation)}$ :  $\tau \in \{monotone, reverse\}$ . It is the conditional probability of re-ordering.

### 4 Decoding

The decoding procedure of a discourse can be derived from the original decoding formula  $e_1^I = \text{argmax}_{e_1^I} P(e_1^I|f_1^I)$ . Given the rhetorical structure of a source sentence and the corresponding rule-table, the translating process is to find an optimal path to get the highest score under structure constrains, which is,

$$\begin{aligned} \text{argmax}_{e_s} \{P(e_s|f_t)\} \\ = \text{argmax}_{e_s} \left\{ \prod_{f_n \in f_t} P(e_{u1}, e_{u2}, \tau|f_n) \right\} \end{aligned}$$

where  $f_t$  is a source RST tree combined by a set of node  $f_n$ .  $e_s$  is the target string combined by series of  $e_n$  (translations of  $f_n$ ).  $f_n$  consists of  $U_1$  and  $U_2$ .  $e_{u1}$  and  $e_{u2}$  are translations of  $U_1$  and  $U_2$  respectively. This global optimization problem is approximately simplified to local optimization to reduce the complexity,

$$\prod_{f_n \in f_t} \text{argmax}_{e_n} \{P(e_{u1}, e_{u2}, \tau|f_n)\}$$

In our paper, we have the following two ways to factorize the above formula,

#### Decoder 1:

$$\begin{aligned} P(e_{u1}, e_{u2}, \tau|f_n) \\ = P(e_{cp}, e_x, e_y, \tau|f_{cp}, f_x, f_y) \\ = P(e_{cp}|f_{cp})P(\tau|e_{cp}, f_{cp})P(e_x|f_x)P(e_y|f_y) \\ = P(r_e|r_f, Rel)P(\tau|r_e, r_f, Rel)P(e_x|f_x)P(e_y|f_y) \end{aligned}$$

where  $e_x$ ,  $e_y$  are the translation of non-terminal parts.  $f_{cp}$  and  $e_{cp}$  are cue-words pair of source and target sides. The first and second factors are just the probabilities introduced in Section 3.2. After approximately simplified to local optimization, the final formulae are re-written as,

$$\text{argmax}_{\tau} \{P(r_e|r_f, Rel)P(\tau|r_e, r_f, Rel)\} \quad (4)$$

$$\text{argmax}_{e_x} \{P(e_x|f_x)\} \quad (5)$$

$$\text{argmax}_{e_y} \{P(e_y|f_y)\} \quad (6)$$

Taking the source sentence with its RST tree in Figure 1 for instance, we adopt a bottom-up manner to do translation recursively. Suppose the best rules selected by (4) are just those written in the figure, Then span [11,13] and [14,21] are firstly translated by (5) and (6). Their translations are then re-packaged by the rule of *Reason*-[10,13,21]. Iteratively, the translations of span [1,9] and [10,21] are re-packaged by the rule of *Antithesis*-[0,9,21] to form the final translation.

**Decoder 2 :** Suppose that the translating process of two spans  $U_1$  and  $U_2$  are independent of each other, we rewrite  $P(e_{u1}, e_{u2}, \tau|f_n)$  as follows,

$$\begin{aligned} &P(e_{u1}, e_{u2}, \tau|f_n) \\ &= P(e_{u1}, e_{u2}, \tau|f_{u1}, f_{u2}) \\ &= P(e_{u1}|f_{u1})P(e_{u2}|f_{u2})P(\tau|r_f, Rel) \\ &= P(e_{u1}|f_{u1})P(e_{u2}|f_{u2}) \sum_{r_e} P(\tau|r_e, r_f, Rel)P(r_e|r_f, Rel) \end{aligned}$$

after approximately simplified to local optimization, the final formulae are re-written as below,

$$\operatorname{argmax}_{e_{u1}} \{Pr(e_{u1}|f_{u1})\} \quad (7)$$

$$\operatorname{argmax}_{e_{u2}} \{Pr(e_{u2}|f_{u2})\} \quad (8)$$

$$\operatorname{argmax}_r \left\{ \sum_e Pr(\tau|r_e, r_f, Rel)Pr(r_e|r_f, Rel) \right\} \quad (9)$$

We also adopt the bottom-up manner similar to Decoder 1. In Figure 1,  $U_1$  and  $U_2$  of *Reason* node are firstly translated. Their translations are then re-ordered. Then the translations of two spans of *Antithesis* node are re-ordered and constructed into the final translation. In Decoder 2, the minimal translation-unit is *edu*. While in Decoder 1, an *edu* is further split into cue-word part and the rest part to obtain the respective translation.

In our decoders, language model(LM) is used for translating *edus* in Formula(5),(6),(7),(8), but not for reordering the upper spans because with the bottom-to-up combination, the spans become longer and harder to be judged by a traditional language model. So we only use RST rules to guide the reordering. But LM will be properly considered in our future work.

## 5 Experiment

### 5.1 Setup

In order to do Chinese RST parser, we annotated over 1,000 complicated sentences on CTB (Xue et al., 2005), among which 1,107 sentences are used for training, and 500 sentences are used for testing. Berkeley parser<sup>2</sup> is used for getting the syntactic trees.

The translation experiment is conducted on Chinese-to-English direction. The bilingual training data is from the LDC corpus<sup>3</sup>. The training corpus contains 2.1M sentence pairs. We obtain the word alignment with the grow-diag-final-and strategy by GIZA++<sup>4</sup>. A 5-gram language model is trained on the Xinhua portion of the English

<sup>2</sup> <http://code.google.com/p/berkeleyparser/>

<sup>3</sup> LDC category number : LDC2000T50, LDC2002E18, LDC2003E07, LDC2004T07, LDC2005T06, LDC2002L27, LDC2005T10 and LDC2005T34

<sup>4</sup> <http://code.google.com/p/giza-pp/>

Gigaword corpus. For tuning and testing, we use NIST03 evaluation data as the development set, and extract the relatively long and complicated sentences from NIST04, NIST05 and CWMT08<sup>5</sup> evaluation data as the test set. The number and average word-length of sentences are 511/36, 320/34, 590/38 respectively. We use case-insensitive BLEU-4 with the shortest length penalty for evaluation.

To create the baseline system, we use the toolkit Moses<sup>6</sup> to build a phrase-based translation system. Meanwhile, considering that Xiong et al. (2009) have presented good results by dividing long and complicated sentences into sub-sentences only by punctuations during decoding, we re-implement their method for comparison.

### 5.2 Results of Chinese RST Parser

Table 2 shows the results of RST parsing. On average, our RS trees are 2 layers deep. The parsing errors mostly result from the segmentation errors, which are mainly caused by syntactic parsing errors. On the other hand, the polysemous cue words, such as “而(but, and, thus)” may lead ambiguity for relation recognition, because they can be clues for different relations.

| Task         | Precision | Recall | F1   |
|--------------|-----------|--------|------|
| Segmentation | 0.74      | 0.83   | 0.78 |
| Labeling     | 0.71      | 0.78   | 0.75 |

Table 2: Segmentation and labeling result.

### 5.3 Results of Translation

Table 3 presents the translation comparison results. In this table, XD represents the method in (Xiong et al., 2009). D1 stands for Decoder-1, and D2 for Decoder-2. Values with boldface are the highest scores in comparison. D2 performs best on the test data with 2.3/0.77/1.43/1.16 points. Compared with XD, our results also outperform by 0.52 points on the whole test data.

Observing and comparing the translation results, we find that our translation results are more readable by maintaining the semantic integrality of the *edus* and by giving more appreciate reorganization of the translated *edus*.

| Testing Set | Baseline | XD    | D1           | D2           |
|-------------|----------|-------|--------------|--------------|
| NIST04      | 29.39    | 31.52 | 31.34        | <b>31.69</b> |
| NIST05      | 29.86    | 29.80 | 30.28        | <b>30.63</b> |
| CWMT08      | 24.31    | 25.24 | <b>25.74</b> | <b>25.74</b> |
| ALL         | 27.85    | 28.49 | 28.66        | <b>29.01</b> |

Table 3: Comparison with related models.

<sup>5</sup> China Workshop on Machine Translation 2008

<sup>6</sup> [www.statmt.org/moses/index.php?n=Main.HomePage](http://www.statmt.org/moses/index.php?n=Main.HomePage)

## 6 Conclusion and Future Work

In this paper, we present an RST-based translation framework for modeling semantic structures in translation model, so as to maintain the semantically functional integrity and hierarchical relations of *edus* during translating. With respect to the existing models, we think our translation framework works more similarly to what human does, and we believe that this research is a crucial step towards discourse-oriented translation.

In the next step, we will study on the implicit discourse relations for Chinese and further modify the RST-based framework. Besides, we will try to combine other current translation models such as syntactic model and hierarchical model into our framework. Furthermore, the more accurate evaluation metric for discourse-oriented translation will be further studied.

### Acknowledgments

The research work has been funded by the Hi-Tech Research and Development Program (“863” Program) of China under Grant No. 2011AA01A207, 2012AA011101, and 2012AA011102 and also supported by the Key Project of Knowledge Innovation Program of Chinese Academy of Sciences under Grant No. KGZD-EW-501.

### References

- David A Duverle and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4<sup>th</sup> International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 665–673. Association for Computational Linguistics.
- Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 909–919. Association for Computational Linguistics.
- Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010a. A sequential model for discourse segmentation. *Computational Linguistics and Intelligent Text Processing*, pages 315–326.
- Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010b. Hilda: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Volume 1*, pages 48–54. Association for Computational Linguistics.
- William C Mann and Sandra A Thompson. 1986. Rhetorical structure theory: Description and construction of text structures. *Technical report, DTIC Document*.
- William C Mann and Sandra A Thompson. 1987. Rhetorical structure theory: A framework for the analysis of texts. *Technical report, DTIC Document*.
- William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.
- Daniel Marcu, Lynn Carlson, and Maki Watanabe. 2000. The automatic translation of discourse structures. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 9–17. Morgan Kaufmann Publishers Inc.
- David Reitter. 2003. Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. *Language*, 18:52.
- Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.
- Billy TM Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, page 1060–1068. Association for Computational Linguistics.
- Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in machine translation. In *Machine Translation Summit*, volume 13, pages 131–138.
- Hao Xiong, Wenwen Xu, Haitao Mi, Yang Liu, and Qun Liu. 2009. Sub-sentence division for tree-based machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 137–140. Association for Computational Linguistics.
- Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207.
- Ming Yue. 2008. Rhetorical structure annotation of Chinese news commentaries. *Journal of Chinese Information Processing*, 4:002.

# Improving machine translation by training against an automatic semantic frame based evaluation metric

Chi-kiu Lo and Karteek Addanki and Markus Saers and Dekai Wu  
HKUST

Human Language Technology Center  
Department of Computer Science and Engineering  
Hong Kong University of Science and Technology

{jackielo|vskaddanki|masaers|dekai}@cs.ust.hk

## Abstract

We present the first ever results showing that tuning a machine translation system against a semantic frame based objective function, MEANT, produces more robustly adequate translations than tuning against BLEU or TER as measured across commonly used metrics and human subjective evaluation. Moreover, for informal web forum data, human evaluators preferred MEANT-tuned systems over BLEU- or TER-tuned systems by a significantly wider margin than that for formal newswire—even though automatic semantic parsing might be expected to fare worse on informal language. We argue that by preserving the meaning of the translations as captured by semantic frames right in the training process, an MT system is constrained to make more accurate choices of both lexical and reordering rules. As a result, MT systems tuned against semantic frame based MT evaluation metrics produce output that is more adequate. Tuning a machine translation system against a semantic frame based objective function is independent of the translation model paradigm, so, any translation model can benefit from the semantic knowledge incorporated to improve translation adequacy through our approach.

## 1 Introduction

We present the first ever results of tuning a statistical machine translation (SMT) system against a semantic frame based objective function in order to produce a more adequate output. We compare the performance of our system with that of two baseline SMT systems tuned against BLEU and TER, the commonly used n-gram and edit distance

based metrics. Our system performs better than the baseline across seven commonly used evaluation metrics and subjective human evaluation on adequacy. Surprisingly, tuning against a semantic MT evaluation metric also significantly outperforms the baseline on the domain of informal web forum data wherein automatic semantic parsing might be expected to fare worse. These results strongly indicate that using a semantic frame based objective function for tuning would drive development of MT towards direction of higher utility.

Glaring errors caused by semantic role confusion that plague the state-of-the-art MT systems are a consequence of using fast and cheap lexical n-gram based objective functions like BLEU to drive their development. Despite enforcing fluency it has been established that these metrics do not enforce translation utility adequately and often fail to preserve meaning closely (Callison-Burch *et al.*, 2006; Koehn and Monz, 2006).

We argue that instead of BLEU, a metric that focuses on getting the meaning right should be used as an objective function for tuning SMT so as to drive continuing progress towards higher utility. MEANT (Lo *et al.*, 2012), is an automatic semantic MT evaluation metric that measures similarity between the MT output and the reference translation via semantic frames. It correlates better with human adequacy judgment than other automatic MT evaluation metrics. Since a high MEANT score is contingent on correct lexical choices as well as syntactic and semantic structures, we believe that tuning against MEANT would improve both translation adequacy and fluency.

Incorporating semantic structures into SMT by tuning against a semantic frame based evaluation metric is independent of the MT paradigm. Therefore, systems from different MT paradigms (such as hierarchical, phrase based, transduction grammar based) can benefit from the semantic information incorporated through our approach.

## 2 Related Work

Relatively little work has been done towards biasing the translation decisions of an SMT system to produce adequate translations that correctly preserve *who did what to whom, when, where and why* (Pradhan *et al.*, 2004). This is because the development of SMT systems was predominantly driven by tuning against n-gram based evaluation metrics such as BLEU or edit distance based metrics such as TER which do not sufficiently bias SMT system's decisions to produce adequate translations. Although there has been a recent surge of work aimed towards incorporating semantics into the SMT pipeline, none attempt to tune against a semantic objective function. Below, we describe some of the attempts to incorporate semantic information into the SMT and present a brief survey on evaluation metrics that focus on rewarding semantically valid translations.

**Utilizing semantics in SMT** In the past few years, there has been a surge of work aimed at incorporating semantics into various stages of the SMT. Wu and Fung (2009) propose a two-pass model that reorders the MT output to match the SRL of the input, which is too late to affect the translation decisions made by the MT system during decoding. In contrast, training against a semantic objective function attempts to improve the decoding search strategy by incorporating a bias towards meaningful translations into the model instead of postprocessing its results.

Komachi *et al.* (2006) and Wu *et al.* (2011) preprocess the input sentence to match the verb frame alternations in the output side. Liu and Gildea (2010) and Aziz *et al.* (2011) use input side SRL to train a tree-to-string SMT system. Xiong *et al.* (2012) trained a discriminative model to predict the position of the semantic roles in the output. All these approaches are orthogonal to the present question of whether to train toward a semantic objective function. Any of the above models could potentially benefit from tuning with semantic metrics.

**MT evaluation metrics** As mentioned previously, tuning against n-gram based metrics such as BLEU (Papineni *et al.*, 2002), NIST (Dodgington, 2002), METEOR (Banerjee and Lavie, 2005) does not sufficiently drive SMT into making decisions to produce adequate translations that correctly preserve *who did what to whom,*

*when, where and why*". In fact, a number of large scale meta-evaluations (Callison-Burch *et al.*, 2006; Koehn and Monz, 2006) report cases where BLEU strongly disagrees with human judgments of translation accuracy. Tuning against edit distance based metrics such as CDER (Leusch *et al.*, 2006), WER (Nießen *et al.*, 2000), and TER (Snover *et al.*, 2006) also fails to sufficiently bias SMT systems towards producing translations that preserve semantic information.

We argue that an SMT system tuned against an adequacy-oriented metric that correlates well with human adequacy judgement produces more adequate translations. For this purpose, we choose MEANT, an automatic semantic MT evaluation metric that focuses on getting the meaning right by comparing the semantic structures of the MT output and the reference. We briefly describe some of the alternative semantic metrics below to justify our choice.

ULC (Giménez and Màrquez, 2007, 2008) is an aggregated metric that incorporates several semantic similarity features and shows improved correlation with human judgement on translation quality (Callison-Burch *et al.*, 2007; Giménez and Màrquez, 2007; Callison-Burch *et al.*, 2008; Giménez and Màrquez, 2008) but no work has been done towards tuning an MT system against ULC perhaps due to its expensive running time. Lambert *et al.* (2006) did tune on QUEEN, a simplified version of ULC that discards the semantic features and is based on pure lexical features. Although tuning on QUEEN produced slightly more preferable translations than solely tuning on BLEU, the metric does not make use of any semantic features and thus fails to exploit any potential gains from tuning to semantic objectives.

Although TINE (Rios *et al.*, 2011) is a recall-oriented automatic evaluation metric which aims to preserve the basic event structure, no work has been done towards tuning an SMT system against it. TINE performs comparably to BLEU and worse than METEOR on correlation with human adequacy judgment.

In contrast to TINE, MEANT (Lo *et al.*, 2012), which is the weighted f-score over the matched semantic role labels of the automatically aligned semantic frames and role fillers, outperforms BLEU, NIST, METEOR, WER, CDER and TER. This makes it more suitable for tuning SMT systems to produce much adequate translations.



| newswire    | BLEU         | NIST        | METEOR no_syn | METEOR       | WER          | CDER         | TER          | MEANT         |
|-------------|--------------|-------------|---------------|--------------|--------------|--------------|--------------|---------------|
| BLEU-tuned  | 29.85        | 8.84        | 52.10         | 55.42        | 67.88        | 55.67        | <b>58.40</b> | 0.1667        |
| TER-tuned   | 25.37        | 6.56        | 48.26         | 51.24        | 66.18        | 52.58        | 56.96        | 0.1578        |
| MEANT-tuned | <b>25.91</b> | <b>7.81</b> | <b>50.15</b>  | <b>53.60</b> | <b>67.76</b> | <b>54.56</b> | 58.61        | <b>0.1676</b> |

Table 1: Translation quality of MT system tuned against MEANT, BLEU and TER on newswire data

| forum       | BLEU        | NIST        | METEOR no_syn | METEOR       | WER          | CDER         | TER          | MEANT         |
|-------------|-------------|-------------|---------------|--------------|--------------|--------------|--------------|---------------|
| BLEU-tuned  | 9.58        | 4.10        | 31.77         | 34.63        | 80.09        | 64.54        | 76.12        | 0.1711        |
| TER-tuned   | 6.94        | 2.21        | 28.55         | 30.85        | 76.15        | 57.96        | 74.73        | 0.1539        |
| MEANT-tuned | <b>7.92</b> | <b>3.11</b> | <b>30.40</b>  | <b>33.08</b> | <b>77.32</b> | <b>61.01</b> | <b>74.64</b> | <b>0.1727</b> |

Table 2: Translation quality of MT system tuned against MEANT, BLEU and TER on forum data

### 3 Tuning SMT against MEANT

We now show that using MEANT as an objective function to drive minimum error rate training (MERT) of state-of-the-art MT systems improves MT utility not only on formal newswire text, but even on informal forum text, where automatic semantic parsing is difficult.

Toward improving translation utility of state-of-the-art MT systems, we chose to use a strong and competitive system in the DARPA BOLT program as our baseline. The baseline system is a Moses hierarchical model trained on a collection of LDC newswire and a small portion of Chinese-English parallel web forum data, together with a 5-gram language model. For the newswire experiment, we used a collection of NIST 02-06 test sets as our development set and NIST 08 test set for evaluation. The development and test sets contain 6,331 and 1,357 sentences respectively with four references. For the forum data experiment, the development and test sets were a held-out subset of the BOLT phase 1 training data. The development and test sets contain 2,000 sentences and 1,697 sentences with one reference.

We use ZMERT (Zaidan, 2009) to tune the baseline because it is a widely used, highly competitive, robust, and reliable implementation of MERT that is also fully configurable and extensible with regard to incorporating new evaluation metrics. In this experiment, we use a MEANT implementation along the lines described in Lo *et al.* (2012).

In each experiment, we tune two contrastive conventional 100-best MERT tuned baseline systems on both newswire and forum data genres; one tuned against BLEU, an n-gram based evaluation metric and the other using TER, an edit distance based metric. As semantic role labeling is expensive we only tuned using 10-best list for MEANT-tuned system. Tuning against BLEU and TER took

around 1.5 hours and 5 hours per iteration respectively whereas tuning against MEANT took about 1.6 hours per iteration.

### 4 Results

Of course, tuning against any metric would maximize the performance of the SMT system on that particular metric, but would be overfitting. For example, something would be seriously wrong if tuning against BLEU did not yield the best BLEU scores. A far more worthwhile goal would be to bias the SMT system to produce adequate translations while achieving the best scores across all the metrics. With this as our objective, we present the results of comparing MEANT-tuned systems against the baselines as evaluated on commonly used automatic metrics and human adequacy judgement.

**Cross-evaluation using automatic metrics** Tables 1 and 2 show that MEANT-tuned systems achieve the best scores across all other metrics in both newswire and forum data genres, when avoiding comparison of the overfit metrics too similar to the one the system was tuned on (the cells shaded in grey in the table: NIST and METEOR are n-gram based metrics, similar to BLEU while WER and CDER are edit distance based metrics, similar to TER). In the newswire domain, however, our system achieves marginally lower TER score than BLEU-tuned system.

Figure 1 shows an example where the MEANT-tuned system produced a more adequate translation that accurately preserves the semantic structure of the input sentence than the two baseline systems. The MEANT scores for the MT output from the BLEU-, TER- and MEANT-tuned systems are 0.0635, 0.1131 and 0.2426 respectively. Both the MEANT score and the human evaluators rank the MT output from the MEANT-tuned sys-

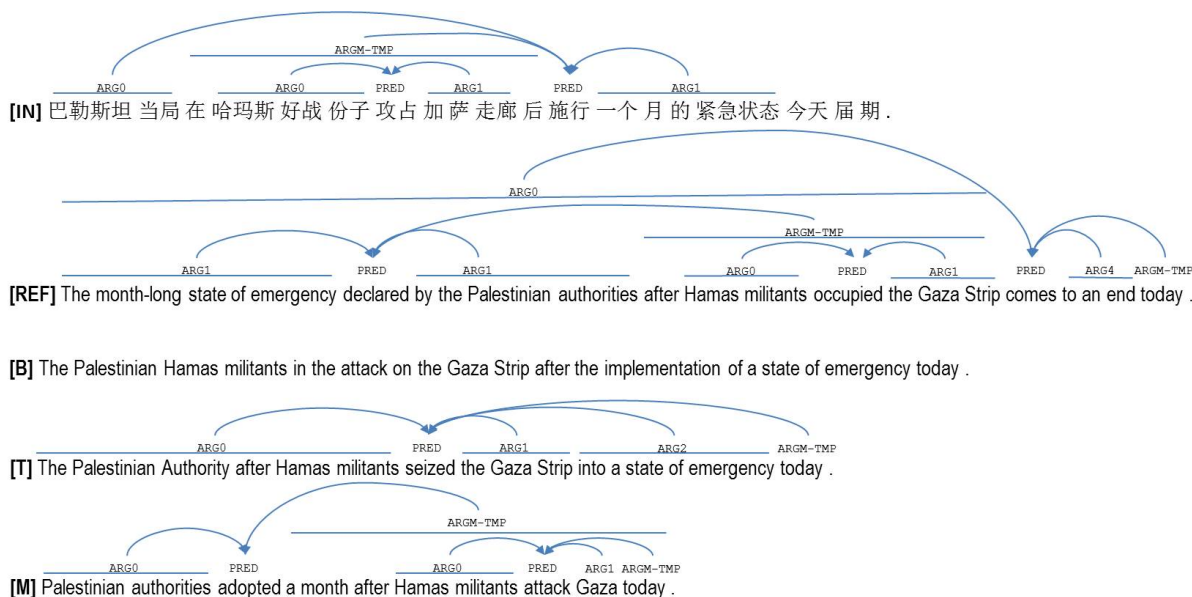


Figure 1: Examples of machine translation output and the corresponding semantic parses from the [B] BLEU-, [T] TER- and [M] MEANT-tuned systems together with [IN] the input sentence and [REF] the reference translation. Note that the MT output of the BLEU-tuned system has no semantic parse output by the automatic shallow semantic parser.

tem as the most adequate translation. In this example, the MEANT-tuned system has translated the two predicates “占领” and “施行” in the input sentence into the correct form of the predicates “attack” and “adopted” in the MT output, whereas the BLEU-tuned system has translated both of them incorrectly (translates the predicates into nouns) and the TER-tuned system has correctly translated only the first predicate (into “seized”) and dropped the second predicate. Moreover, for the frame “占领” in the input sentence, the MEANT-tuned system has correctly translated the ARG0 “哈马斯好战份子” into “Hamas militants” and the ARG1 “加萨走廊” into “Gaza”. However, the TER-tuned system has dropped the predicate “施行” so that the corresponding arguments “The Palestinian Authority” and “into a state of emergency” have all been incorrectly associated with the predicate “占领/seized”. This example shows that the translation adequacy of SMT has been improved by tuning against MEANT because the MEANT-tuned system is more accurately preserving the semantic structure of the input sentence.

Our results show that MEANT-tuned system maintains a balance between lexical choices and word order because it performs well on n-gram based metrics that reward lexical matching and edit distance metrics that penalize incorrect word

order. This is not surprising as a high MEANT score relies on a high degree of semantic structure matching, which is contingent upon correct lexical choices as well as syntactic and semantic structures.

**Human subjective evaluation** In line with our original objective of biasing SMT systems towards producing adequate translations, we conduct a human evaluation to judge the translation utility of the outputs produced by MEANT-, BLEU- and TER-tuned systems. Following the manual evaluation protocol of Lambert *et al.* (2006), we randomly draw 150 sentences from the test set in each domain to form the manual evaluation set. Table 3 shows the MEANT scores of the two manual evaluation sets. In both evaluation sets, like in the test sets, the output from the MEANT-tuned system score slightly higher in MEANT than that from the BLEU-tuned system and significantly higher than that from the TER-tuned system. The output of each tuned MT system along the input sentence and the reference were presented to human evaluators. Each evaluation set is ranked by two evaluators for measuring inter-evaluator agreement.

Table 4 indicates that output of the MEANT-tuned system is ranked adequate more frequently compared to BLEU- and TER-tuned baselines for both newswire and web forum genres. The inter-

|             | newswire | forum  |
|-------------|----------|--------|
| BLEU-tuned  | 0.1564   | 0.1663 |
| TER-tuned   | 0.1203   | 0.1453 |
| MEANT-tuned | 0.1633   | 0.1737 |

Table 3: MEANT scores of each system in the 150-sentence manual evaluation set.

|                 | newswire |        | forum  |        |
|-----------------|----------|--------|--------|--------|
|                 | Eval 1   | Eval 2 | Eval 1 | Eval 2 |
| BLEU-tuned (B)  | 37       | 42     | 47     | 42     |
| TER-tuned (T)   | 22       | 24     | 28     | 23     |
| MEANT-tuned (M) | 55       | 56     | 59     | 68     |
| B=T             | 14       | 12     | 0      | 0      |
| M=B             | 5        | 4      | 8      | 9      |
| M=T             | 4        | 4      | 4      | 4      |
| M=B=T           | 13       | 9      | 4      | 4      |

Table 4: No. of sentences ranked the most adequate by human evaluators for each system.

| $H_1$                    | newswire | forum |
|--------------------------|----------|-------|
| MEANT-tuned > BLEU-tuned | 80%      | 95%   |
| MEANT-tuned > TER-tuned  | 99%      | 99%   |

Table 5: Significance level of accepting the alternative hypothesis.

evaluator agreement is 84% and 70% for newswire and forum data genres respectively.

We performed the right-tailed two proportion significance test on human evaluation of the SMT system outputs for both the genres. Table 5 shows that the MEANT-tuned system generates more adequate translations than the TER-tuned system at the 99% significance level for both newswire and web forum genres. The MEANT-tuned system is ranked more adequate than the BLEU-tuned system at the 95% significance level on the web forum genre and for the newswire genre the hypothesis is accepted at a significance level of 80%. The high inter-evaluator agreement and the significance tests confirm that MEANT-tuned system is better at producing adequate translations compared to BLEU- or TER-tuned systems.

**Informal vs. formal text** The results of table 4 and 5 also show that—surprisingly—the human evaluators preferred MEANT-tuned system output over BLEU-tuned and TER-tuned system output by a far wider margin on the informal forum text compared to the formal newswire text. The MEANT-tuned system is better than both baselines at the 80% significance level for the formal text genre. For the informal text genre, it performs the two baselines at the 95% significance level. Although one might expect an semantic

frame dependent metric such as MEANT to perform poorly on the domain of informal text, surprisingly, it nonetheless significantly outperforms the baselines at the task of generating adequate output. This indicates that the design of the MEANT evaluation metric is robust enough to tune an SMT system towards adequate output on informal text domains despite the shortcomings of automatic shallow semantic parsing.

## 5 Conclusion

We presented the first ever results to demonstrate that tuning an SMT system against MEANT produces much adequate translation than tuning against BLEU or TER, as measured across all other commonly used metrics and human subjective evaluation. We also observed that tuning against MEANT succeeds in producing adequate output significantly more frequently even on the informal text such as web forum data. By preserving the meaning of the translations as captured by semantic frames right in the training process, an MT system is constrained to make more accurate choices of both lexical and reordering rules. The performance of our system as measured across all commonly used metrics indicate that tuning against a semantic MT evaluation metric does produce output which is adequate and fluent.

We believe that tuning on MEANT would prove equally useful for MT systems based on any paradigm, especially where the model does not incorporate semantic information to improve the adequacy of the translations produced and using MEANT as an objective function to tune SMT would drive sustainable development of MT towards the direction of higher utility.

## Acknowledgment

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

- Wilker Aziz, Miguel Rios, and Lucia Specia. Shallow semantic trees for SMT. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT2011)*, 2011.
- Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005.
- Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of BLEU in Machine Translation Research. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 249–256, 2006.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (Meta-) evaluation of Machine Translation. In *Proceedings of the 2nd Workshop on Statistical Machine Translation*, pages 136–158, 2007.
- Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further Meta-evaluation of Machine Translation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation*, pages 70–106, 2008.
- George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 138–145, San Diego, California, 2002.
- Jesús Giménez and Lluís Màrquez. Linguistic features for automatic evaluation of heterogeneous MT systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264, Prague, Czech Republic, June 2007.
- Jesús Giménez and Lluís Màrquez. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 195–198, Columbus, Ohio, June 2008.
- Philipp Koehn and Christof Monz. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings of the Workshop on Statistical Machine Translation (WMT-06)*, pages 102–121, 2006.
- Mamoru Komachi, Yuji Matsumoto, and Masaaki Nagata. Phrase reordering for statistical machine translation based on predicate-argument structure. In *Proceedings of the 3rd International Workshop on Spoken Language Translation (IWSLT 2006)*, 2006.
- Patrik Lambert, Jesús Giménez, Marta R Costajussá, Enrique Amigó, Rafael E Banchs, Lluís Màrquez, and JAR Fonollosa. Machine Translation system development based on human likeness. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 246–249. IEEE, 2006.
- Gregor Leusch, Nicola Ueffing, and Hermann Ney. CDer: Efficient MT Evaluation Using Block Movements. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, 2006.
- Ding Liu and Daniel Gildea. Semantic role features for machine translation. In *Proceedings of the 23rd international conference on Computational Linguistics (COLING-10)*, 2010.
- Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. Fully Automatic Semantic MT Evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT2012)*, 2012.
- Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. A Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*, 2000.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July 2002.
- Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. Shallow Semantic Parsing Using Support Vector Machines. In *Proceedings of the 2004 Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-04)*, 2004.
- Miguel Rios, Wilker Aziz, and Lucia Specia. Tine: A metric to assess mt adequacy. In *Proceed-*

- ings of the Sixth Workshop on Statistical Machine Translation*, pages 116–122. Association for Computational Linguistics, 2011.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA-06)*, pages 223–231, Cambridge, Massachusetts, August 2006.
- Dekai Wu and Pascale Fung. Semantic Roles for SMT: A Hybrid Two-Pass Model. In *Proceedings of the 2009 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT-09)*, pages 13–16, 2009.
- Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. Extracting preordering rules from predicate-argument structures. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP-11)*, 2011.
- Deyi Xiong, Min Zhang, and Haizhou Li. Modeling the Translation of Predicate-Argument Structure for SMT. In *Proceedings of the Joint conference of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-12)*, 2012.
- Omar F. Zaidan. 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, 2009.

# Bilingual Lexical Cohesion Trigger Model for Document-Level Machine Translation

Guosheng Ben<sup>†</sup> Deyi Xiong<sup>‡\*</sup> Zhiyang Teng<sup>†</sup> Yajuan Lü<sup>†</sup> Qun Liu<sup>§†</sup>

<sup>†</sup>Key Laboratory of Intelligent Information Processing  
Institute of Computing Technology, Chinese Academy of Sciences  
{benguosheng, tengzhiyang, lvyajuan, liuqun}@ict.ac.cn

<sup>‡</sup>School of Computer Science and Technology, Soochow University  
{dyxiong}@suda.edu.cn

<sup>§</sup>Centre for Next Generation Localisation, Dublin City University  
{qliu}@computing.dcu.ie

## Abstract

In this paper, we propose a bilingual lexical cohesion trigger model to capture lexical cohesion for document-level machine translation. We integrate the model into hierarchical phrase-based machine translation and achieve an absolute improvement of 0.85 BLEU points on average over the baseline on NIST Chinese-English test sets.

## 1 Introduction

Current statistical machine translation (SMT) systems are mostly sentence-based. The major drawback of such a sentence-based translation fashion is the neglect of inter-sentential dependencies. As a linguistic means to establish inter-sentential links, lexical cohesion ties sentences together into a meaningfully interwoven structure through words with the same or related meanings (Wong and Kit, 2012).

This paper studies lexical cohesion devices and incorporate them into document-level machine translation. We propose a **bilingual lexical cohesion trigger model** to capture lexical cohesion for document-level SMT. We consider a lexical cohesion item in the source language and its corresponding counterpart in the target language as a trigger pair, in which we treat the source language lexical cohesion item as the trigger and its target language counterpart as the triggered item. Then we use mutual information to measure the strength of the dependency between the trigger and triggered item.

We integrate this model into a hierarchical phrase-based SMT system. Experiment results

show that it is able to achieve substantial improvements over the baseline.

The remainder of this paper proceeds as follows: Section 2 introduces the related work and highlights the differences between previous methods and our model. Section 3 elaborates the proposed bilingual lexical cohesion trigger model, including the details of identifying lexical cohesion devices, measuring dependency strength of bilingual lexical cohesion triggers and integrating the model into SMT. Section 4 presents experiments to validate the effectiveness of our model. Finally, Section 5 concludes with future work.

## 2 Related Work

As a linguistic means to establish inter-sentential links, cohesion has been explored in the literature of both linguistics and computational linguistics. Cohesion is defined as relations of meaning that exist within the text and divided into grammatical cohesion that refers to the syntactic links between text items and lexical cohesion that is achieved through word choices in a text by Halliday and Hasan (1976). In order to improve the quality of machine translation output, cohesion has served as a high level quality criterion in post-editing (Vasconcellos, 1989). As a part of COMTIS project, grammatical cohesion is integrated into machine translation models to capture inter-sentential links (Cartoni et al., 2011). Wong and Kit (2012) incorporate lexical cohesion to machine translation evaluation metrics to evaluate document-level machine translation quality. Xiong et al. (2013) integrate various target-side lexical cohesion devices into document-level machine translation. Lexical cohesion is also partially explored in the cache-based translation models of Gong et al. (2011) and translation consistency constraints of Xiao et al.

\*Corresponding author

(2011).

All previous methods on lexical cohesion for document-level machine translation as mentioned above have one thing in common, which is that they do not use any source language information. Our work is mostly related to the mutual information trigger based lexical cohesion model proposed by Xiong et al. (2013). However, we significantly extend their model to a bilingual lexical cohesion trigger model that captures both source and target-side lexical cohesion items to improve target word selection in document-level machine translation.

### 3 Bilingual Lexical Cohesion Trigger Model

#### 3.1 Identification of Lexical Cohesion Devices

Lexical cohesion can be divided into reiteration and collocation (Wong and Kit, 2012). Reiteration is a form of lexical cohesion which involves the repetition of a lexical item. Collocation is a pair of lexical items that have semantic relations, such as synonym, near-synonym, superordinate, subordinate, antonym, meronym and so on. In the collocation, we focus on the synonym/near-synonym and super-subordinate semantic relations<sup>1</sup>. We define lexical cohesion devices as content words that have lexical cohesion relations, namely the reiteration, synonym/near-synonym and super-subordinate.

Reiteration is common in texts. Take the following two sentences extracted from a document for example (Halliday and Hasan, 1976).

1. There is a boy climbing the old *elm*.
2. That *elm* is not very safe.

We see that word *elm* in the first sentence is repeated in the second sentence. Such reiteration devices are easy to identify in texts. Synonym/near-synonym is a semantic relationship set. We can use WordNet (Fellbaum, 1998) to identify them. WordNet is a lexical resource that clusters words with the same sense into a semantic group called synset. Synsets in WordNet are organized according to their semantic relations. Let  $s(w)$  denote a function that defines all synonym words of  $w$  grouped in the same synset in WordNet. We can use the function to compute all synonyms and near-synonyms for word  $w$ . In order to represent conveniently,  $s_0$  denotes the set of synonyms in

<sup>1</sup>Other collocations are not used frequently, such as antonyms. So we do not consider them in our study.

$s(w)$ . Near-synonym set  $s_1$  is defined as the union of all synsets that are defined by the function  $s(w)$  where  $w \in s_0$ . It can be formulated as follows.

$$s_1 = \bigcup_{w \in s_0} s(w) \quad (1)$$

$$s_2 = \bigcup_{w \in s_1} s(w) \quad (2)$$

$$s_3 = \bigcup_{w \in s_2} s(w) \quad (3)$$

Similarly  $s_m$  can be defined recursively as follows.

$$s_m = \bigcup_{w \in s_{m-1}} s(w) \quad (4)$$

Obviously, We can find synonyms and near-synonyms for word  $w$  according to formula (4).

Superordinate and subordinate are formed by words with an is-a semantic relation in WordNet. As the super-subordinate relation is also encoded in WordNet, we can define a function that is similar to  $s(w)$  identify hypernyms and hyponyms.

We use *rep*, *syn* and *hyp* to represent the lexical cohesion device reiteration, synonym/near-synonym and super-subordinate respectively hereafter for convenience.

#### 3.2 Bilingual Lexical Cohesion Trigger Model

In a bilingual text, lexical cohesion is present in the source and target language in a synchronous fashion. We use a trigger model capture such a bilingual lexical cohesion relation. We define  $xRy$  ( $R \in \{rep, syn, hyp\}$ ) as a trigger pair where  $x$  is the trigger in the source language and  $y$  the triggered item in the target language. In order to capture these synchronous relations between lexical cohesion items in the source language and their counterparts in the target language, we use word alignments. First, we identify a monolingual lexical cohesion relation in the target language in the form of  $tRy$  where  $t$  is the trigger,  $y$  the triggered item that occurs in a sentence succeeding the sentence of  $t$ , and  $R \in \{rep, syn, hyp\}$ . Second, we find word  $x$  in the source language that is aligned to  $t$  in the target language. We may find multiple words  $x_1^k$  in the source language that are aligned to  $t$ . We use all of them  $x_i R t (1 \leq i \leq k)$  to define bilingual lexical cohesion relations. In this way, we can create bilingual lexical cohesion relations  $xRy$  ( $R \in \{rep, syn, hyp\}$ ):  $x$  being the trigger and  $y$  the triggered item.

The possibility that  $y$  will occur given  $x$  is equal to the chance that  $x$  triggers  $y$ . Therefore we measure the strength of dependency between the trigger and triggered item according to pointwise mutual information (PMI) (Church and Hanks, 1990; Xiong et al., 2011).

The PMI for the trigger pair  $xRy$  where  $x$  is the trigger,  $y$  the triggered item that occurs in a target sentence succeeding the target sentence that aligns to the source sentence of  $x$ , and  $R \in \{rep, syn, hyp\}$  is calculated as follows.

$$PMI(xRy) = \log\left(\frac{p(x, y, R)}{p(x, R)p(y, R)}\right) \quad (5)$$

The joint probability  $p(x, y, R)$  is:

$$p(x, y, R) = \frac{C(x, y, R)}{\sum_{x, y} C(x, y, R)} \quad (6)$$

where  $C(x, y, R)$  is the number of aligned bilingual documents where both  $x$  and  $y$  occur with the relation  $R$  in different sentences, and  $\sum_{x, y} C(x, y, R)$  is the number of bilingual documents where this relation  $R$  occurs. The marginal probabilities of  $p(x, R)$  and  $p(y, R)$  can be calculated as follows.

$$p(x, R) = \sum_y C(x, y, R) \quad (7)$$

$$p(y, R) = \sum_x C(x, y, R) \quad (8)$$

Given a target sentence  $y_1^m$ , our bilingual lexical cohesion trigger model is defined as follows.

$$MI_R(y_1^m) = \prod_{y_i} \exp(PMI(\cdot Ry_i)) \quad (9)$$

where  $y_i$  are content words in the sentence  $y_1^m$  and  $PMI(\cdot Ry_i)$  is the maximum PMI value among all trigger words  $x_1^q$  from source sentences that have been recently translated, where trigger words  $x_1^q$  have an  $R$  relation with word  $y_i$ .

$$PMI(\cdot Ry_i) = \max_{1 \leq j \leq q} PMI(x_j Ry_i) \quad (10)$$

Three models  $MI_{rep}(y_1^m)$ ,  $MI_{syn}(y_1^m)$ ,  $MI_{hyp}(y_1^m)$  for the reiteration device, the synonym/near-synonym device and the super-subordinate device can be formulated as above. They are integrated into the log-linear model of SMT as three different features.

### 3.3 Decoding

We incorporate our bilingual lexical cohesion trigger model into a hierarchical phrase-based system (Chiang, 2007). We add three features as follows.

- $MI_{rep}(y_1^m)$
- $MI_{syn}(y_1^m)$
- $MI_{hyp}(y_1^m)$

In order to quickly calculate the score of each feature, we calculate PMI for each trigger pair before decoding. We translate document one by one. During translation, we maintain a cache to store source language sentences of recently translated target sentences and three sets  $S_{rep}$ ,  $S_{syn}$ ,  $S_{hyp}$  to store source language words that have the relation of  $\{rep, syn, hyp\}$  with content words generated in target language. During decoding, we update scores according to formula (9). When one sentence is translated, we store the corresponding source sentence into the cache. When the whole document is translated, we clear the cache for the next document.

## 4 Experiments

### 4.1 Setup

Our experiments were conducted on the NIST Chinese-English translation tasks with large-scale training data. The bilingual training data contains 3.8M sentence pairs with 96.9M Chinese words and 109.5M English words from LDC<sup>2</sup>. The monolingual data for training data English language model includes the Xinhua portion of the Gigaword corpus. The development set is the NIST MT Evaluation test set of 2005 (MT05), which contains 100 documents. We used the sets of MT06 and MT08 as test sets. The numbers of documents in MT06, MT08 are 79 and 109 respectively. For the bilingual lexical cohesion trigger model, we collected data with document boundaries explicitly provided. The corpora are selected from our bilingual training data and the whole Hong Kong parallel text corpus<sup>3</sup>, which contains 103,236 documents with 2.80M sentences.

<sup>2</sup>The corpora include LDC2002E18, LDC2003E07, LDC2003E14, LDC2004E12, LDC2004T07, LDC2004T08 (Only Hong Kong News), LDC2005T06 and LDC2005T10.

<sup>3</sup>They are LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08 (Hong Kong Hansards/Laws/News).



We obtain the word alignments by running GIZA++ (Och and Ney, 2003) in both directions and applying “grow-diag-final-and” refinement (Koehn et al., 2003). We apply SRI Language Modeling Toolkit (Stolcke, 2002) to train a 4-gram language model with Kneser-Ney smoothing. Case-insensitive NIST BLEU (Papineni et al., 2002) was used to measure translation performance. We used minimum error rate training MERT (Och, 2003) for tuning the feature weights.

#### 4.2 Distribution of Lexical Cohesion Devices in the Target Language

| Cohesion Device | Percentage(%) |
|-----------------|---------------|
| <i>rep</i>      | 30.85         |
| <i>syn</i>      | 17.58         |
| <i>hyp</i>      | 18.04         |

Table 1: Distributions of lexical cohesion devices in the target language.

In this section we want to study how these lexical cohesion devices distribute in the training data before conducting our experiments on the bilingual lexical cohesion model. Here we study the distribution of lexical cohesion in the target language (English). Table 1 shows the distribution of percentages that are counted based on the content words in the training data. From Table 1, we can see that the reiteration cohesion device is nearly a third of all content words (30.85%), synonym/near-synonym and super-subordinate devices account for 17.58% and 18.04%. Obviously, lexical cohesion devices are frequently used in real-world texts. Therefore capturing lexical cohesion devices is very useful for document-level machine translation.

#### 4.3 Results

| System             | MT06         | MT08         | Avg          |
|--------------------|--------------|--------------|--------------|
| Base               | 30.43        | 23.32        | 26.88        |
| <i>rep</i>         | 31.24        | 23.70        | 27.47        |
| <i>syn</i>         | 30.92        | 23.71        | 27.32        |
| <i>hyp</i>         | 30.97        | 23.48        | 27.23        |
| <i>rep+syn+hyp</i> | <b>31.47</b> | <b>23.98</b> | <b>27.73</b> |

Table 2: BLEU scores with various lexical cohesion devices on the test sets MT06 and MT08. “Base” is the traditional hierarchical system, “Avg” is the average BLEU score on the two test sets.

Results are shown in Table 2. From the table, we can see that integrating a single lexical cohesion device into SMT, the model gains an improvement of up to 0.81 BLEU points on the MT06 test set. Combining all three features *rep+syn+hyp* together, the model gains an improvement of up to 1.04 BLEU points on MT06 test set, and an average improvement of 0.85 BLEU points on the two test sets of MT06 and MT08. These stable improvements strongly suggest that our bilingual lexical cohesion trigger model is able to substantially improve the translation quality.

## 5 Conclusions

In this paper we have presented a bilingual lexical cohesion trigger model to incorporate three classes of lexical cohesion devices, namely the reiteration, synonym/near-synonym and super-subordinate devices into a hierarchical phrase-based system. Our experimental results show that our model achieves a substantial improvement over the baseline. This displays the advantage of exploiting bilingual lexical cohesion.

Grammatical and lexical cohesion have often been studied together in discourse analysis. In the future, we plan to extend our model to capture both grammatical and lexical cohesion in document-level machine translation.

## Acknowledgments

This work was supported by 863 State Key Project (No.2011AA01A207) and National Key Technology R&D Program(No.2012BAH39B03). Qun Liu was also 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.

## References

- Bruno Cartoni, Andrea Gesmundo, James Henderson, Cristina Grisot, Paola Merlo, Thomas Meyer, Jacques Moeschler, Sandrine Zufferey, Andrei Popescu-Belis, et al. 2011. Improving mt coherence through text-level processing of input texts: the comtis project. [http://webcast.in2p3.fr/videos-the\\_comtis\\_project](http://webcast.in2p3.fr/videos-the_comtis_project).
- David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.
- Kenneth Ward Church and Patrick Hanks. 1990. Word

- association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.
- Christine Fellbaum. 1998. Wordnet: An electronic lexical database.
- Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- M.A.K Halliday and Ruqayia Hasan. 1976. Cohesion in english. *English language series*, 9.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. 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. 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, Sapporo, Japan, July. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904.
- Muriel Vasconcellos. 1989. Cohesion and coherence in the presentation of machine translation products. *Georgetown University Round Table on Languages and Linguistics*, pages 89–105.
- Billy T. M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068, Jeju Island, Korea, July. Association for Computational Linguistics.
- Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in machine translation. In *Machine Translation Summit*, volume 13, pages 131–138.
- Deyi Xiong, Min Zhang, and Haizhou Li. 2011. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1288–1297, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Deyi Xiong, Guosheng Ben, Min Zhang, Yajuan Lv, and Qun Liu. 2013. Modeling lexical cohesion for document-level machine translation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, Beijing, China.

# Generalized Reordering Rules for Improved SMT

**Fei Huang**

IBM T. J. Watson Research Center  
huangfe@us.ibm.com

**Cezar Pendus**

IBM T. J. Watson Research Center  
cpendus@us.ibm.com

## Abstract

We present a simple yet effective approach to syntactic reordering for Statistical Machine Translation (SMT). Instead of solely relying on the top-1 best-matching rule for source sentence preordering, we generalize fully lexicalized rules into partially lexicalized and unlexicalized rules to broaden the rule coverage. Furthermore, we consider multiple permutations of all the matching rules, and select the final reordering path based on the weighed sum of reordering probabilities of these rules. Our experiments in English-Chinese and English-Japanese translations demonstrate the effectiveness of the proposed approach: we observe consistent and significant improvement in translation quality across multiple test sets in both language pairs judged by both humans and automatic metric.

## 1 Introduction

Languages are structured data. The proper handling of linguistic structures (such as word order) has been one of the most important yet most challenging tasks in statistical machine translation (SMT). It is important because it has significant impact on human judgment of Machine Translation (MT) quality: an MT output without structure is just like a bag of words. It is also very challenging due to the lack of effective methods to model the structural difference between source and target languages.

A lot of research has been conducted in this area. Approaches include distance-based penalty function (Koehn et. al. 2003) and lexicalized distortion models such as (Tillman 2004), (Al-Onaizan and Papineni 2006). Because these models are relatively easy to compute, they are widely used in phrase-based SMT systems. Hierarchical phrase-based system (Hiero,

Chiang, 2005) utilizes long range reordering information without syntax. Other models use more syntactic information (string-to-tree, tree-to-string, tree-to-tree, string-to-dependency etc.) to capture the structural difference between language pairs, including (Yamada and Knight, 2001), (Zollmann and Venugopal, 2006), (Liu et. al. 2006), and (Shen et. al. 2008). These models demonstrate better handling of sentence structures, while the computation is more expensive compared with the distortion-based models.

In the middle of the spectrum, (Xia and McCord 2004), (Collins et. al 2005), (Wang et. al. 2007), and (Visweswariah et. al. 2010) combined the benefits of the above two strategies: their approaches reorder an input sentence based on a set of reordering rules defined over the source sentence's syntax parse tree. As a result, the re-ordered source sentence resembles the word order of its target translation. The reordering rules are either hand-crafted or automatically learned from the training data (source parse trees and bitext word alignments). These rules can be unlexicalized (only including the constituent labels) or fully lexicalized (including both the constituent labels and their head words). The unlexicalized reordering rules are more general and can be applied broadly, but sometimes they are not discriminative enough. In the following English-Chinese reordering rules,

$$\begin{aligned} 0.44 \quad NP \ PP \rightarrow 0 \ 1 \\ 0.56 \quad NP \ PP \rightarrow 1 \ 0 \end{aligned}$$

the NP and PP nodes are reordered with close to random probabilities. When the constituents are attached with their headwords, the reordering probability is much higher than that of the unlexicalized rules.

$$\begin{aligned} 0.20 \quad NP: \textit{testimony} \ PP: \textit{by} \rightarrow 0 \ 1 \\ 0.80 \quad NP: \textit{testimony} \ PP: \textit{by} \rightarrow 1 \ 0 \end{aligned}$$

Unfortunately, the application of lexicalized reordering rules is constrained by data sparseness: it is unlikely to train the  $NP: \langle \textit{noun} \rangle$

*PP*:<*prep*> reordering rules for every noun-preposition combination. Even for the learnt lexicalized rules, their counts are also relatively small, thus the reordering probabilities may not be estimated reliably, which could lead to incorrect reordering decisions.

To alleviate this problem, we generalize fully lexicalized rules into partially lexicalized rules, which are further generalized into unlexicalized rules. Such generalization allows partial match when the fully lexicalized rules can not be found, thus achieving broader rule coverage.

Given a node of a source parse tree, we find all the matching rules and consider all their possible reorder permutations. Each permutation has a reordering score, which is the weighted sum of reordering probabilities of all the matching rules. We reorder the child nodes based on the permutation with the highest reordering score. Finally we translate the reordered sentence in a phrase-based SMT system. Our experiments in English to Chinese (**EnZh**) and English to Japanese (**EnJa**) translation demonstrate the effectiveness of the proposed approach: we observe consistent improvements across multiple test sets in multiple language pairs and significant gain in human judgment of the MT quality.

This paper is organized as follows: in section 2 we briefly introduce the syntax-based reordering technique. In section 3, we describe our approach. In section 4, we show the experiment results, which is followed by conclusion in section 5.

## 2 Baseline Syntax-based Reordering

In the general syntax-based reordering, reordering is achieved by permuting the children of any interior node in the source parse tree. Although there are cases where reordering is needed across multiple constituents, this still is a simple and effective technique.

Formally, the reordering rule is a triple  $\{p, lhs, rhs\}$ , where  $p$  is the reordering probability,  $lhs$  is the left hand side of the rule, i.e., the constituent label sequence of a parse tree node, and  $rhs$  is the reordering permutation derived either from hand-crafted rules as in (Collins et. al 2005) and (Wang et. al. 2007), or from training data as in (Visweswariah et. al. 2010).

The training data includes bilingual sentence pairs with word alignments, as well as the source sentences' parse trees. The children's relative

order of each node is decided according to their average alignment position in the target sentence. Such relative order is a permutation of the integer sequence  $[0, 1, \dots, N-1]$ , where  $N$  is the number of children of the given parse node. The counts of each permutation of each parse label sequence will be collected from the training data and converted to probabilities as shown in the examples in Section 1. Finally, only the permutation with the highest probability is selected to reorder the matching parse node. The SMT system is re-trained on reordered training data to translate reordered input sentences.

Following the above approach, only the reordering rule  $[0.56 NP PP \rightarrow 1 0]$  is kept in the above example. In other words, all the *NP PP* phrases will be reordered, even though the reordering is only slightly preferred in all the training data.

## 3 Generalized Syntactic Reordering

As shown in the previous examples, reordering depends not only on the constituents' parse labels, but also on the headwords of the constituents. Such fully lexicalized rules suffer from data sparseness: there is either no matching lexicalized rule for a given parse node or the matching rule's reordering probability is unreliable. We address the above issues with rule generalization, then consider all the permutations from multi-level rule matching.

### 3.1 Rule Generalization

Lexicalized rules are applied only when both the constituent labels and headwords match. When only the labels match, these reordering rules are not used. To increase the rule coverage, we generalize the fully lexicalized rules into partially lexicalized and unlexicalized rules.

We notice that many lexicalized rules share similar reordering permutations, thus it is possible to merge them to form a partially lexicalized rule, where lexicalization only appears at selected constituent's headword. Although it is possible to have multiple lexicalizations in a partially lexicalized rule (which will exponentially increase the total number of rules), we observe that most of the time reordering is triggered by a single constituent. Therefore we keep one lexicalization in the partially lexicalized rules. For example, the following lexicalized rule:

*VB:appeal PP-MNR:by PP-DIR:to --> 1 2 0*

will be converted into the following 3 partially lexicalized rules:

*VB:appeal PP-MNR PP-DIR* --> 1 2 0  
*VB PP-MNR:by PP-DIR* --> 1 2 0  
*VB PP-MNR PP-DIR:to* --> 1 2 0

The count of each rule will be the *sum* of the fully lexicalized rules which can derive the given partially lexicalized rule. In the above preordering rules, “MNR” and “DIR” are functional labels, indicating the semantic labels (“manner”, “direction”) of the parse node.

We could go even further, converting the partially lexicalized rules into unlexicalized rules. This is similar to the baseline syntax reordering model, although we will keep all their possible permutations and counts for rule matching, as shown below.

5 *VB PP-MNR PP-DIR* --> 2 0 1  
22 *VB PP-MNR PP-DIR* --> 2 1 0  
21 *VB PP-MNR PP-DIR* --> 0 1 2  
41 *VB PP-MNR PP-DIR* --> 1 2 0  
35 *VB PP-MNR PP-DIR* --> 1 0 2

Note that to reduce the noise from parsing and word alignment errors, we only keep the reordering rules that appear at least 5 times. Then we convert the counts into probabilities:

$$p_i(rhs | lhs_i) = \frac{C_i(rhs, lhs_i)}{\sum C_i(*, lhs_i)}$$

where  $i \in \{f, p, u\}$  represents the fully, partially and un-lexicalized rules, and  $C_i(rhs, lhs_i)$  is the count of rule ( $lhs_i \rightarrow rhs$ ) in type  $i$  rules.

When we convert the most specific *fully lexicalized* rules to the more general *partially lexicalized* rules and then to the most general *unlexicalized* rules, we increase the rule coverage while keep their discriminative power at different levels as much as possible.

### 3.2 Multiple Permutation Multi-level Rule Matching

When applying the three types of reordering rules to reorder a parse tree node, we find all the matching rules and consider all possible permutations. As multiple levels of rules can lead to the same permutation with different probabilities, we take the weighted sum of probabilities from all matching rules (with the same *rhs*). Therefore, the permutation decision is not based on any particular rule, but the combination of all the rules matching different

levels of context. As opposed to the general syntax-based reordering approaches, this strategy achieves a desired balance between broad rule coverage and specific rule match: when a fully lexicalized rule matches, it has strong influence on the permutation decision given the richer context. If such specific rule is unavailable or has low probability, more general (partial and unlexicalized) rules will have higher weights. For each permutation we compute the weighted reordering probability, then select the permutation that has the highest score.

Formally, given a parse tree node  $T$ , let  $lhs_f$  be the label:head\_word sequence of the fully lexicalized rules matching  $T$ . Similarly,  $lhs_p$  and  $lhs_u$  are the sequences of the matching partially lexicalized and unlexicalized rules, respectively, and let  $rhs$  be their possible permutations. The top-score permutation is computed as:

$$rhs^* = \arg \max_{rhs} \sum_{i \in \{f, p, u\}} w_i p_i(rhs | lhs_i)$$

where  $w_i$ 's are the weights of different kind of rules and  $p_i$  is reordering probability of each rule. The weights are chosen empirically based on the performance on a held-out tuning set. In our experiments,  $w_f=1.0$ ,  $w_p=0.5$ , and  $w_u=0.2$ , where higher weights are assigned to more specific rules.

For each parse tree node, we identify the top permutation choice and reorder its children accordingly. The source parse tree is traversed breadth-first.

## 4 Experiments

We applied the generalized syntax-based reordering on both English-Chinese (EnZh) and English-Japanese (EnJa) translations. Our English parser is IBM's maximum entropy constituent parser (Ratnaparkhi 1999) trained on Penn Treebank. Experiments in (Visweswariah et. al. 2010) indicated that minimal difference was observed using Berkeley's parser or IBM's parser for reordering.

Our EnZh training data consists of 20 million sentence pairs (~250M words), half of which are from LDC released bilingual corpora and the other half are from technical domains (e.g., software manual). We first trained automatic word alignments (HMM alignments in both directions and a MaxEnt alignment (Ittycheriah and Roukos, 2005)), then parsed the English sentences with the IBM parser. We extracted different reordering rules from the word alignments and the English parse trees. After

frequency-based pruning, we obtained 12M lexicalized rules, 13M partially lexicalized rules and 600K unlexicalized rules. Using these rules, we applied preordering on the English sentences and then built an SMT system with the reordered training data. Our decoder is a phrase-based decoder (Tillman 2006), where various features are combined within the log-linear framework. These features include source-to-target phrase translation score based on relative frequency, source-to-target and target-to-source word-to-word translation scores, a 5-gram language model score, distortion model scores and word count.

|                       | <b>Tech1</b> | <b>Tech2</b> | <b>MT08</b>  |
|-----------------------|--------------|--------------|--------------|
| <b># of sentences</b> | 582          | 600          | 1859         |
| <b>PBMT</b>           | 33.08        | 31.35        | 36.81        |
| <b>UnLex</b>          | 33.37        | 31.38        | 36.39        |
| <b>FullLex</b>        | 34.12        | 31.62        | 37.14        |
| <b>PartLex</b>        | 34.13        | 32.58        | 37.60        |
| <b>MPML</b>           | <b>34.34</b> | <b>32.64</b> | <b>38.02</b> |

Table 1: MT experiment comparison using different syntax-based reordering techniques on English-Chinese test sets.

We selected one tuning set from software manual domain (**Tech1**), and used PRO tuning (Hopkins and May 2011) to select decoder feature weights. Our test sets include one from the online technical support domain (**Tech2**) and one from the news domain: the NIST MT08 English-Chinese evaluation test data. The translation quality is measured by BLEU score (Papineni et. al., 2001). Table 1 shows the BLEU score of the baseline phrase-based system (**PBMT**)

that uses lexicalized reordering at decoding time rather than *preordering*. Next, Table 1 shows the translation results with several preordered systems that use unlexicalized (**UnLex**), fully lexicalized (**FullLex**) and partially lexicalized (**PartLex**) rules, respectively. The lexicalized reordering model is still applicable for preordered systems so that some preordering errors can be recovered at run time.

First we observed that the **UnLex** preordering model on average does not improve over the typical phrase-based MT baseline due to its limited discriminative power. When the preordering decision is conditioned on the head word, the **FullLex** model shows some gains (~0.3 pt) thanks to the richer matching context, while the **PartLex** model improves further over the **FullLex** model because of its broader

coverage. Combining all three with multi-permutation, multi-level rule matching (**MPML**) brings the most gains, with consistent (~1.3 Bleu points) improvement over the baseline system on all the test sets. Note that the Bleu scores on the news domain (**MT08**) are higher than those on the tech domain. This is because the Tech1 and Tech2 have one reference translation while MT08 has 4 reference translations.

In addition to the automatic MT evaluation, we also used human judgment of quality of the MT translation on a set of randomly selected 125 sentences from the baseline and improved reordering systems. The human judgment score is 2.82 for the **UnLex** system output, and 3.04 for the improved **MPML** reordering output. The 0.2 point improvement on the 0-5 scale is considered significant.

|                       | <b>Tech1</b> | <b>Tech2</b> | <b>News</b>  |
|-----------------------|--------------|--------------|--------------|
| <b># of sentences</b> | 1000         | 600          | 600          |
| <b>PBMT</b>           | 56.45        | 35.45        | 21.70        |
| <b>UnLex</b>          | 59.22        | 38.36        | 23.08        |
| <b>FullLex</b>        | 57.55        | 36.56        | 22.23        |
| <b>PartLex</b>        | 59.80        | 38.47        | 23.13        |
| <b>MPML</b>           | <b>59.94</b> | <b>38.62</b> | <b>23.31</b> |

Table 2: MT experiment comparison using generalized syntax-based reordering techniques on English-Japanese test sets.

We also apply the same generalized reordering technique on English-Japanese (**EnJa**) translation. As there is very limited publicly available English-Japanese parallel data, most our training data (20M sentence pairs) is from the in-house software manual domain. We use the same English parser and phrase-based decoder as in EnZh experiment. Table 2 shows the translation results on technical and news domain test sets. All the test sets have single reference translation.

First, we observe that the improvement from preordering is larger than that in EnZh MT (1.6-3 pts vs. 1 pt). This is because the word order difference between English and Japanese is larger than that between English and Chinese (Japanese is a SOV language while both English and Chinese are SVO languages). Without preordering, correct word orders are difficult to obtain given the typical skip-window beam search in the **PBMT**. Also, as in EnZh, the **PartLex** model outperforms the **UnLex** model, both of which being significantly better than the **FullLex** model due to the limited rule coverage in the later model: only 50% preordering rules

are applied in the **FullLex** model. **Tech1** test set is a very close match to the training data thus its BLEU score is much higher.

## 5 Conclusion and Future Work

To summarize, we made the following improvements:

1. We generalized fully lexicalized reordering rules to partially lexicalized and unlexicalized rules for broader rule coverage and reduced data sparseness.
2. We allowed multiple permutation, multi-level rule matching to select the best reordering path.

Experiment results show consistent and significant improvements on multiple English-Chinese and English-Japanese test sets judged by both automatic and human judgments.

In future work we would like to explore new methods to prune the phrase table without degrading MT performance and to make rule extraction and reordering more robust to parsing errors.

## Acknowledgement

The authors appreciate helpful comments from anonymous reviewers as well as fruitful discussions with Karthik Visweswariah and Salim Roukos.

## References

- Yaser Al-Onaizan , Kishore Papineni, Distortion models for statistical machine translation, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, p.529-536, July 17-18, 2006, Sydney, Australia
- David Chiang, A hierarchical phrase-based model for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.263-270, June 25-30, 2005, Ann Arbor, Michigan
- Michael Collins , Philipp Koehn , Ivona Kucerov, Clause restructuring for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.531-540, June 25-30, 2005, Ann Arbor, Michigan
- Mark Hopkins, Jonathan May, Tuning as ranking, In Proceedings of the Conference on Empirical Methods in Natural Language Processing 2011, pp. 1352-1362. Association for Computational Linguistics.
- Abraham Ittycheriah , Salim Roukos, A maximum entropy word aligner for Arabic-English machine translation, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.89-96, October 06-08, 2005, Vancouver, British Columbia, Canada
- Philipp Koehn , Franz Josef Och , Daniel Marcu, Statistical phrase-based translation, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, p.48-54, May 27-June 01, 2003, Edmonton, Canada
- Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In Proceedings of COLING/ACL 2006, pages 609-616, Sydney, Australia, July.
- Libin Shen, Jinxi Xu and Ralph Weischedel 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL). Columbus, OH, USA, June 15 - 20, 2008.
- Christoph Tillmann, A unigram orientation model for statistical machine translation, Proceedings of HLT-NAACL 2004: Short Papers, p.101-104, May 02-07, 2004, Boston, Massachusetts
- Christoph Tillmann. 2006. Efficient Dynamic Programming Search Algorithms for Phrase-based SMT. In Proc. of the Workshop CHPSLP at HLT'06.
- Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In Proceedings of EMNLP-CoNLL.
- Karthik Visweswariah , Jiri Navratil , Jeffrey Sorensen , Vijil Chenthamarakshan , Nanda Kambhatla, Syntax based reordering with automatically derived rules for improved statistical machine translation, Proceedings of the 23rd International Conference on Computational Linguistics, p.1119-1127, August 23-27, 2010, Beijing, China
- Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. Machine Learning, 34(1-3).
- Fei Xia , Michael McCord, Improving a statistical MT system with automatically learned rewrite patterns, Proceedings of the 20th international conference on Computational Linguistics, p.508-es, August 23-27, 2004, Geneva, Switzerland

Kenji Yamada , Kevin Knight, A syntax-based statistical translation model, Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, p.523-530, July 06-11, 2001, Toulouse, France

Andreas Zollmann , Ashish Venugopal, Syntax augmented machine translation via chart parsing, Proceedings of the Workshop on Statistical Machine Translation, June 08-09, 2006, New York City, New York  
Alfred. V. Aho and Jeffrey D. Ullman. 1972. The Theory of Parsing, Translation and Compiling, volume 1. Prentice-Hall, Englewood Cliffs, NJ.



# A Tightly-coupled Unsupervised Clustering and Bilingual Alignment Model for Transliteration

Tingting Li<sup>1</sup>, Tiejun Zhao<sup>1</sup>, Andrew Finch<sup>2</sup>, Chunyue Zhang<sup>1</sup>

<sup>1</sup>Harbin Institute of Technology, Harbin, China

<sup>2</sup>NICT, Japan

<sup>1</sup>{ttli, tjzhao, cyzhang}@mtlab.hit.edu.cn

<sup>2</sup>andrew.finch@nict.go.jp

## Abstract

Machine Transliteration is an essential task for many NLP applications. However, names and loan words typically originate from various languages, obey different transliteration rules, and therefore may benefit from being modeled independently. Recently, transliteration models based on Bayesian learning have overcome issues with over-fitting allowing for many-to-many alignment in the training of transliteration models. We propose a novel coupled Dirichlet process mixture model (cDPMM) that simultaneously clusters and bilingually aligns transliteration data within a single unified model. The unified model decomposes into two classes of non-parametric Bayesian component models: a Dirichlet process mixture model for clustering, and a set of multinomial Dirichlet process models that perform bilingual alignment independently for each cluster. The experimental results show that our method considerably outperforms conventional alignment models.

## 1 Introduction

Machine transliteration methods can be categorized into phonetic-based models (Knight et al., 1998), spelling-based models (Brill et al., 2000), and hybrid models which utilize both phonetic and spelling information (Oh et al., 2005; Oh et al., 2006). Among them, statistical spelling-based models which directly align characters in the training corpus have become popular because they are language-independent, do not require phonetic knowledge, and are capable of achieving state-of-the-art performance (Zhang et al., 2012b). A major problem with real-word transliteration corpora is that they are usually not clean, may contain name pairs with various linguistic origins and

this can hinder the performance of spelling-based models because names from different origins obey different pronunciation rules, for example:

“Kim Jong-il/金正恩” (Korea),

“Kana Gaski/金崎” (Japan),

“Haw King/霍金” (England),

“Jin yong/金庸” (China).

The same Chinese character “金” should be aligned to different romanized character sequences: “Kim”, “Kana”, “King”, “Jin”. To address this issue, many name classification methods have been proposed, such as the supervised language model-based approach of (Li et al., 2007), and the unsupervised approach of (Huang et al., 2005) that used a bottom-up clustering algorithm. (Li et al., 2007) proposed a supervised transliteration model which classifies names based on their origins and genders using a language model; it switches between transliteration models based on the input. (Hagiwara et al., 2011) tackled the issue by using an unsupervised method based on the EM algorithm to perform a soft classification.

Recently, non-parametric Bayesian models (Finch et al., 2010; Huang et al., 2011; Hagiwara et al., 2012) have attracted much attention in the transliteration field. In comparison to many of the previous alignment models (Li et al., 2004; Jiampojarn et al., 2007; Berg-Kirkpatrick et al., 2011), the non-parametric Bayesian models allow unconstrained monotonic many-to-many alignment and are able to overcome the inherent over-fitting problem.

Until now most of the previous work (Li et al., 2007; Hagiwara et al., 2011) is either affected by the multi-origins factor, or has issues with over-fitting. (Hagiwara et al., 2012) took these two factors into consideration, but their approach still operates within an EM framework and model order selection by hand is necessary prior to training.

We propose a simple, elegant, fully-unsupervised solution based on a single generative model able to both cluster and align simultaneously. The coupled Dirichlet Process Mixture Model (cDPMM) integrates a Dirichlet process mixture model (DPMM) (Antoniak, 1974) and a Bayesian Bilingual Alignment Model (BBAM) (Finch et al., 2010). The two component models work synergistically to support one another: the clustering model sorts the data into classes so that self-consistent alignment models can be built using data of the same type, and at the same time the alignment probabilities from the alignment models drive the clustering process.

In summary, the key advantages of our model are as follows:

- it is based on a single, unified generative model;
- it is fully unsupervised;
- it is an infinite mixture model, and does not require model order selection – it is effectively capable of discovering an appropriate number of clusters from the data;
- it is able to handle data from multiple origins;
- it can perform many-to-many alignment without over-fitting.

## 2 Model Description

In this section we describe the methodology and realization of the proposed cDPMM in detail.

### 2.1 Terminology

In this paper, we concentrate on the alignment process for transliteration. The proposed cDPMM segments a bilingual corpus of transliteration pairs into bilingual character sequence-pairs. We will call these sequence-pairs Transliteration Units (TUs). We denote the source and target of a TU as  $s_1^m = \langle s_1, \dots, s_m \rangle$  and  $t_1^n = \langle t_1, \dots, t_n \rangle$  respectively, where  $s_i$  ( $t_i$ ) is a single character in source (target) language. We use the same notation  $(\mathbf{s}, \mathbf{t}) = (\langle s_1, \dots, s_m \rangle, \langle t_1, \dots, t_n \rangle)$  to denote a transliteration pair, which we can write as  $x = (s_1^m, t_1^n)$  for simplicity. Finally, we express the training set itself as a set of sequence pairs:  $D = \{x_i\}_{i=1}^I$ . Our aim is to obtain a bilingual alignment  $\langle (s_1, t_1), \dots, (s_l, t_l) \rangle$  for each transliteration pair  $x_i$ , where each  $(s_j, t_j)$  is a segment of the whole pair (a TU) and  $l$  is the number of segments used to segment  $x_i$ .

### 2.2 Methodology

Our cDPMM integrates two Dirichlet process models: the DPMM clustering model, and the BBAM alignment model which is a multinomial Dirichlet process.

A *Dirichlet process mixture model*, models the data as a mixture of distributions – one for each cluster. It is an infinite mixture model, and the number of components is not fixed prior to training. Equation 1 expresses the DPMM hierarchically.

$$\begin{aligned} G_c | \alpha_c, G_{0c} &\sim DP(\alpha_c, G_{0c}) \\ \theta_k | G_c &\sim G_c \\ x_i | \theta_k &\sim f(x_i | \theta_k) \end{aligned} \quad (1)$$

where  $G_{0c}$  is the base measure and  $\alpha_c > 0$  is the concentration parameter for the distribution  $G_c$ .  $x_i$  is a name pair in training data, and  $\theta_k$  represents the parameters of a candidate cluster  $k$  for  $x_i$ . Specifically  $\theta_k$  contains the probabilities of all the TUs in cluster  $k$ .  $f(x_i | \theta_k)$  (defined in Equation 7) is the probability that mixture component  $k$  parameterized by  $\theta_k$  will generate  $x_i$ .

The alignment component of our cDPMM is a *multinomial Dirichlet process* and is defined as follows:

$$\begin{aligned} G_a | \alpha_a, G_{0a} &\sim DP(\alpha_a, G_{0a}) \\ (\mathbf{s}_j, \mathbf{t}_j) | G_a &\sim G_a \end{aligned} \quad (2)$$

The subscripts ‘c’ and ‘a’ in Equations 1 and 2 indicate whether the terms belong to the clustering or alignment model respectively.

The generative story for the cDPMM is simple: first generate an infinite number of clusters, choose one, then generate a transliteration pair using the parameters that describe the cluster. The basic sampling unit of the cDPMM for the clustering process is a transliteration pair, but the basic sampling unit for BBAM is a TU. In order to integrate the two processes in a single model we treat a transliteration pair as a sequence of TUs generated by a BBAM model. The BBAM generates a sequence (a transliteration pair) based on the joint source-channel model (Li et al., 2004). We use a blocked version of a Gibbs sampler to train each BBAM (see (Mochihashi et al., 2009) for details of this process).

### 2.3 The Alignment Model

This model is a multinomial DP model. Under the Chinese restaurant process (CRP) (Aldous, 1985)

interpretation, each unique TU corresponds to a dish served at a table, and the number of customers in each table represents the count of a particular TU in the model.

The probability of generating the  $j^{\text{th}}$  TU  $(s_j, t_j)$  is,

$$P((s_j, t_j)|(s_{-j}, t_{-j})) = \frac{N((s_j, t_j)) + \alpha_a G_{0a}((s_j, t_j))}{N + \alpha_a} \quad (3)$$

where  $N$  is the total number of TUs generated so far, and  $N((s_j, t_j))$  is the count of  $(s_j, t_j)$ .  $(s_{-j}, t_{-j})$  are all the TUs generated so far except  $(s_j, t_j)$ . The base measure  $G_{0a}$  is a joint spelling model:

$$\begin{aligned} G_{0a}((s, t)) &= P(|s|)P(s||s|)P(|t|)P(t||t|) \\ &= \frac{\lambda_s^{|s|}}{|s|!} e^{-\lambda_s} \mathbf{v}_s^{-|s|} \times \frac{\lambda_t^{|t|}}{|t|!} e^{-\lambda_t} \mathbf{v}_t^{-|t|} \end{aligned} \quad (4)$$

where  $|s|$  ( $|t|$ ) is the length of the source (target) sequence,  $\mathbf{v}_s$  ( $\mathbf{v}_t$ ) is the vocabulary (alphabet) size of the source (target) language, and  $\lambda_s$  ( $\lambda_t$ ) is the expected length of source (target) side.

## 2.4 The Clustering Model

This model is a DPMM. Under the CRP interpretation, a transliteration pair corresponds to a customer, the dish served on each table corresponds to an origin of names.

We use  $z = (z_1, \dots, z_I)$ ,  $z_i \in \{1, \dots, K\}$  to indicate the cluster of each transliteration pair  $x_i$  in the training set and  $\theta = (\theta_1, \dots, \theta_K)$  to represent the parameters of the component associated with each cluster.

In our model, each mixture component is a multinomial DP model, and since  $\theta_k$  contains the probabilities of all the TUs in cluster  $k$ , the number of parameters in each  $\theta_k$  is uncertain and changes with the transliteration pairs that belong to the cluster. For a new cluster (the  $K + 1^{\text{th}}$  cluster), we use Equation 4 to calculate the probability of each TU. The cluster membership probability of a transliteration pair  $x_i$  is calculated as follows,

$$P(z_i = k|D, \theta, z_{-i}) \propto \frac{n_k}{n - 1 + \alpha_c} P(x_i|z, \theta_k) \quad (5)$$

$$P(z_i = K + 1|D, \theta, z_{-i}) \propto \frac{\alpha_c}{n - 1 + \alpha_c} P(x_i|z, \theta_{K+1}) \quad (6)$$

where  $n_k$  is the number of transliteration pairs in the existing cluster  $k \in \{1, \dots, K\}$  (cluster  $K + 1$  is a newly created cluster),  $z_i$  is the cluster indicator for  $x_i$ , and  $z_{-i}$  is the sequence of observed clusters up to  $x_i$ . As mentioned earlier, basic sampling units are inconsistent for the clustering and alignment model, therefore to couple the models the BBAM generates transliteration pairs as a sequence of TUs, these pairs are then used directly in the DPMM.

Let  $\gamma = \langle (s_1, t_1), \dots, (s_l, t_l) \rangle$  be a derivation of a transliteration pair  $x_i$ . To make the model integration process explicit, we use function  $f$  to calculate the probability  $P(x_i|z, \theta_k)$ , where  $f$  is defined as follows,

$$f(x_i|\theta_k) = \begin{cases} \sum_{\gamma \in R} \prod_{(s,t) \in \gamma} P(s, t|\theta_k) & k \in \{1, \dots, K\} \\ \sum_{\gamma \in R} \prod_{(s,t) \in \gamma} G_{0c}(s, t) & k = K + 1 \end{cases} \quad (7)$$

where  $R$  denotes the set of all derivations of  $x_i$ ,  $G_{0c}$  is the same as Equation 4.

The cluster membership  $z_i$  is sampled together with the derivation  $\gamma$  in a single step according to  $P(z_i = k|D, \theta, z_{-i})$  and  $f(x_i|\theta_k)$ . Following the method of (Mochihashi et al., 2009), first  $f(x_i|\theta_k)$  is calculated by forward filtering, and then a sample  $\gamma$  is taken by backward sampling.

## 3 Experiments

### 3.1 Corpora

To empirically validate our approach, we investigate the effectiveness of our model by conducting English-Chinese name transliteration generation on three corpora containing name pairs of varying degrees of mixed origin. The first two corpora were drawn from the ‘‘Names of The World’s Peoples’’ dictionary published by Xin Hua Publishing House. The first corpus was constructed with names only originating from English language (EO), and the second with names originating from English, Chinese, Japanese evenly (ECJ-O). The third corpus was created by extracting name pairs from LDC (Linguistic Data Consortium) Named Entity List, which contains names from all over the world (Multi-O). We divided the datasets into training, development and test sets for each corpus with a ratio of 10:1:1. The details of the division are displayed in Table 2.

| cDPMM Alignment   | BBAM Alignment   |
|---|--|
| mun 蒙 din 丁 ger 格(0, English)<br>ding 丁 guo 果(2, Chinese)<br>tei 丁 be 部(3, Japanese)                      | mun 蒙 din 丁 ger 格<br>din 丁 g _ guo 果<br>t _  丁 e _ ibe 部                 |
| fan 范 chun 纯 yi 一(2, Chinese)<br>hong 洪 il 一 sik 植(5, Korea)<br>sei 静 ichi 一 ro 郎(4, Japanese)            | fan 范 chun 纯 y _ i 一<br>hong 洪 i 一 i _ si 植 k _  <br>sei 静 ch _ i 一 ro 郎 |
| dom 东 b 布 ro 罗 w 夫 s 斯 ki 基(0, Russian)<br>he 何 dong 东 chang 昌(2, Chinese)<br>b 布 ran 兰 don 东(0, English) | do 东 mb 布 ro 罗 w 夫 s 斯 ki 基<br>he 何 don 东 gchang 昌<br>b 布 ran 兰 don 东    |

Table 1: Typical alignments from the BBAM and cDPMM.

### 3.2 Baselines

We compare our alignment model with GIZA++ (Och et al., 2003) and the Bayesian bilingual alignment model (BBAM). We employ two decoding models: a phrase-based machine translation decoder (specifically Moses (Koehn et al., 2007)), and the DirecTL decoder (Jiamponjamarn et al., 2009). They are based on different decoding strategies and optimization targets, and therefore make the comparison more comprehensive. For the Moses decoder, we applied the grow-diag-final-and heuristic algorithm to extract the phrase table, and tuned the parameters using the BLEU metric.

| Corpora | Corpus Scale |             |         |
|---------|--------------|-------------|---------|
|         | Training     | Development | Testing |
| EO      | 32,681       | 3,267       | 3,267   |
| ECJ-O   | 32,500       | 3,250       | 3,250   |
| Multi-O | 33,291       | 3,328       | 3,328   |

Table 2: Statistics of the experimental corpora.

To evaluate the experimental results, we utilized 3 metrics from the Named Entities Workshop (NEWS) (Zhang et al., 2012a): word accuracy in top-1 (ACC), fuzziness in top-1 (Mean F-score) and mean reciprocal rank (MRR).

### 3.3 Parameter Setting

In our model, there are several important parameters: 1)  $max\_s$ , the maximum length of the source sequences of the alignment tokens; 2)  $max\_t$ , the maximum length of the target sequences of the alignment tokens; and 3)  $nc$ , the initial number of classes for the training data. We set  $max\_s = 6$ ,  $max\_t = 1$  and  $nc = 5$  empirically based on a small pilot experiment. The Moses decoder was used with default settings except for the distortion-limit which was set to 0 to ensure monotonic decoding. For the DirecTL decoder the following settings were used:  $cs = 4$ ,  $ng = 9$  and  $nBest =$

5.  $cs$  denotes the size of context window for features,  $ng$  indicates the size of  $n$ -gram features and  $nBest$  is the size of transliteration candidate list for updating the model in each iteration. The concentration parameter  $\alpha_c$ ,  $\alpha_a$  of the clustering model and the BBAM was learned by sampling its value. Following (Blunsom et al., 2009) we used a vague gamma prior  $\Gamma(10^{-4}, 10^4)$ , and sampled new values from a log-normal distribution whose mean was the value of the parameter, and variance was 0.3. We used the Metropolis-Hastings algorithm to determine whether this new sample would be accepted. The parameters  $\lambda_s$  and  $\lambda_t$  in Equation 4 were set to  $\lambda_s = 4$  and  $\lambda_t = 1$ .

|             | Model  | EO    | ECJ-O | Multi-O |
|-------------|--------|-------|-------|---------|
| #(Clusters) | cDPMM  | 5.8   | 9.5   | 14.3    |
|             | GIZA++ | 14.43 | 5.35  | 6.62    |
| #(Targets)  | BBAM   | 6.06  | 2.45  | 2.91    |
|             | cDPMM  | 9.32  | 3.45  | 4.28    |

Table 3: Alignment statistics.

### 3.4 Experimental Results

Table 3 shows some details of the alignment results. The #(Clusters) represents the average number of clusters from the cDPMM. It is averaged over the final 50 iterations, and the classes which contain less than 10 name pairs are excluded. The #(Targets) represents the average number of English character sequences that are aligned to each Chinese sequence. From the results we can see that in terms of the number of alignment targets: GIZA++ > cDPMM > BBAM. GIZA++ has considerably more targets than the other approaches, and this is likely to be a symptom of it overfitting the data. cDPMM can alleviate the overfitting through its BBAM component, and at the same time effectively model the diversity in Chinese character sequences caused by multi-origin. Table 1 shows some typical TUs from the alignments produced by BBAM and cDPMM on corpus Multi-O. The information in brackets in Table 1, represents the ID of the class and origin of

| Corpora | Model | Evaluation    |               |               |
|---------|-------|---------------|---------------|---------------|
|         |       | ACC           | M-Fscore      | MRR           |
| EO      | GIZA  | 0.7241        | 0.8881        | 0.8061        |
|         | BBAM  | 0.7286        | 0.8920        | 0.8043        |
|         | cDPMM | <b>0.7398</b> | <b>0.8983</b> | <b>0.8126</b> |
| ECJ-O   | GIZA  | 0.5471        | 0.7278        | 0.6268        |
|         | BBAM  | 0.5522        | 0.7370        | 0.6344        |
|         | cDPMM | <b>0.5643</b> | <b>0.7420</b> | <b>0.6446</b> |
| Multi-O | GIZA  | 0.4993        | 0.7587        | 0.5986        |
|         | BBAM  | 0.5163        | 0.7769        | 0.6123        |
|         | cDPMM | <b>0.5237</b> | <b>0.7796</b> | <b>0.6188</b> |

Table 4: Comparison of different methods using the Moses phrase-based decoder.

the name pair; the symbol ‘\_’ indicates a “NULL” alignment. We can see the Chinese characters “丁(ding) 一(yi) 东(dong)” have different alignments in different origins, and that the cDPMM has provided the correct alignments for them.

We used the sampled alignment from running the BBAM and cDPMM models for 100 iterations, and combined the alignment tables of each class together. The experiments are therefore investigating whether the alignment has been meaningfully improved by the clustering process. We would expect further gains from exploiting the class information in the decoding process (as in (Li et al., 2007)), but this remains future research. The top-10 transliteration candidates were used for testing. The detailed experimental results are shown in Tables 4 and 5.

Our proposed model obtained the highest performance on all three datasets for all evaluation metrics by a considerable margin. Surprisingly, for dataset EO although there is no multi-origin factor, we still observed a respectable improvement in every metric. This shows that although names may have monolingual origin, there are hidden factors which can allow our model to succeed, possibly related to gender or convention. Other models based on supervised classification or clustering with fixed classes may fail to capture these characteristics.

To guarantee the reliability of the comparative results, we performed significance testing based on paired bootstrap resampling (Efron et al., 1993). We found all differences to be significant ( $p < 0.05$ ).

## 4 Conclusion

In this paper we propose an elegant unsupervised technique for monotonic sequence alignment based on a single generative model. The key ben-

| Corpora | Model | Evaluation    |               |               |
|---------|-------|---------------|---------------|---------------|
|         |       | ACC           | M-Fscore      | MRR           |
| EO      | GIZA  | 0.6950        | 0.8812        | 0.7632        |
|         | BBAM  | 0.7152        | 0.8899        | 0.7839        |
|         | cDPMM | <b>0.7231</b> | <b>0.8933</b> | <b>0.7941</b> |
| ECJ-O   | GIZA  | 0.3325        | 0.6208        | 0.4064        |
|         | BBAM  | 0.3427        | 0.6259        | 0.4192        |
|         | cDPMM | <b>0.3521</b> | <b>0.6302</b> | <b>0.4316</b> |
| Multi-O | GIZA  | 0.3815        | 0.7053        | 0.4592        |
|         | BBAM  | 0.3934        | 0.7146        | 0.4799        |
|         | cDPMM | <b>0.3970</b> | <b>0.7179</b> | <b>0.4833</b> |

Table 5: Comparison of different methods using the DirecTL decoder.

efits of our model are that it can handle data from multiple origins, and model using many-to-many alignment without over-fitting. The model operates by clustering the data into classes while simultaneously aligning it, and is able to discover an appropriate number of classes from the data. Our results show that our alignment model can improve the performance of a transliteration generation system relative to two other state-of-the-art aligners. Furthermore, the system produced gains even on data of monolingual origin, where no obvious clusters in the data were expected.

## Acknowledgments

We thank the anonymous reviewers for their valuable comments and helpful suggestions. We also thank Chonghui Zhu, Mo Yu, and Wenwen Zhang for insightful discussions. This work was supported by National Natural Science Foundation of China (61173073), and the Key Project of the National High Technology Research and Development Program of China (2011AA01A207).

## References

- D.J. Aldous. 1985. Exchangeability and Related Topics. *École d’Été St Flour 1983*. Springer, 1985, 1117:1–198.
- C.E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*. 2:1152, 174.
- Taylor Berg-Kirkpatrick and Dan Klein. 2011. Simple effective decipherment via combinatorial optimization. In *Proc. of EMNLP*, pages 313–321.
- P. Blunsom, T. Cohn, C. Dyer, and Osborne, M. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proc. of ACL*, pages 782–790.
- Eric Brill and Robert C. Moore. 2000. An Improved Error Model for Noisy Channel Spelling Correction. In *Proc. of ACL*, pages 286–293.

- B. Efron and R. J. Tibshirani 1993. An Introduction to the Bootstrap. Chapman & Hall, New York, NY.
- Andrew Finch and Eiichiro Sumita. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. In *Proc. of the 7th International Workshop on Spoken Language Translation*, pages 259–266.
- Masato Hagiwara and Satoshi Sekine. 2011. Latent Class Transliteration based on Source Language Origin. In *Proc. of ACL (Short Papers)*, pages 53–57.
- Masato Hagiwara and Satoshi Sekine. 2012. Latent semantic transliteration using dirichlet mixture. In *Proc. of the 4th Named Entity Workshop*, pages 30–37.
- Fei Huang, Stephan Vogel, and Alex Waibel. 2005. Clustering and Classifying Person Names by Origin. In *Proc. of AAAI*, pages 1056–1061.
- Yun Huang, Min Zhang and Chew Lim Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In *Proc. of ACL*, pages 534–539.
- Sittichai Jiampojarn, Grzegorz Kondrak and Tarek Sherif. 2007. Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Proc. of NAACL*, pages 372–379.
- Sittichai Jiampojarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer and Grzegorz Kondrak. 2009. DirecTL: a Language Independent Approach to Transliteration. In *Proc. of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1056–1061.
- Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Journal of Computational Linguistics*, pages 28–31.
- Philipp Koehn and Hieu Hoang and Alexandra Birch and Chris Callison-Burch and Marcello Federico and Nicola Bertoldi and Brooke Cowan and Wade Shen and Christine Moran and Richard Zens and Chris Dyer and Ondrej Bojar and Alexandra Constantin and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL*.
- Haizou Li, Min Zhang, and Jian Su 2004. A joint source-channel model for machine transliteration. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Morristown, NJ, USA, 159.
- Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic Transliteration of Personal Names. In *Proc. of ACL*, pages 120–127.
- Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proc. of ACL/IJCNLP*, pages 100–108.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Journal of Comput. Linguist.*, 29(1):19–51.
- Jong-Hoon Oh, and Key-Sun Choi. 2005. Machine Learning Based English-to-Korean Transliteration Using Grapheme and Phoneme Information. *Journal of IEICE Transactions*, 88-D(7):1737–1748.
- Jong-Hoon Oh, Key-Sun Choi, and Hitoshi Isahara. 2006. A machine transliteration model based on correspondence between graphemes and phonemes. *Journal of ACM Trans. Asian Lang. Inf. Process.*, 5(3):185–208.
- Min Zhang, Haizhou Li, Ming Liu and A Kumaran. 2012a. Whitepaper of NEWS 2012 shared task on machine transliteration. In *Proc. of the 4th Named Entity Workshop (NEWS 2012)*, pages 1–9.
- Min Zhang, Haizhou Li, A Kumaran and Ming Liu. 2012b. Report of NEWS 2012 Machine Transliteration Shared Task. In *Proc. of the 4th Named Entity Workshop (NEWS 2012)*, pages 10–20.

# Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?

**Nadir Durrani**

University of Edinburgh  
dnadir@inf.ed.ac.uk

**Alexander Fraser**

Ludwig Maximilian University Munich  
fraser,schmid@cis.uni-muenchen.de

**Helmut Schmid**

**Hieu Hoang**     **Philipp Koehn**

University of Edinburgh  
hieu.hoang,pkoehn@inf.ed.ac.uk

## Abstract

The phrase-based and N-gram-based SMT frameworks complement each other. While the former is better able to memorize, the latter provides a more principled model that captures dependencies across phrasal boundaries. Some work has been done to combine insights from these two frameworks. A recent successful attempt showed the advantage of using phrase-based search on top of an N-gram-based model. We probe this question in the reverse direction by investigating whether integrating N-gram-based translation and reordering models into a phrase-based decoder helps overcome the problematic phrasal independence assumption. A large scale evaluation over 8 language pairs shows that performance does significantly improve.

## 1 Introduction

Phrase-based models (Koehn et al., 2003; Och and Ney, 2004) learn local dependencies such as reorderings, idiomatic collocations, deletions and insertions by memorization. A fundamental drawback is that phrases are translated and reordered independently of each other and contextual information outside of phrasal boundaries is ignored. The monolingual language model somewhat reduces this problem. However i) often the language model cannot overcome the dispreference of the translation model for nonlocal dependencies, ii) source-side contextual dependencies are still ignored and iii) generation of lexical translations and reordering is separated.

The N-gram-based SMT framework addresses these problems by learning Markov chains over se-

quences of minimal translation units (MTUs) also known as tuples (Mariño et al., 2006) or over operations coupling lexical generation and reordering (Durrani et al., 2011). Because the models condition the MTU probabilities on the previous MTUs, they capture non-local dependencies and both source and target contextual information across phrasal boundaries.

In this paper we study the effect of integrating tuple-based N-gram models (TSM) and operation-based N-gram models (OSM) into the phrase-based model in Moses, a state-of-the-art phrase-based system. Rather than using POS-based rewrite rules (Crego and Mariño, 2006) to form a search graph, we use the ability of the phrase-based system to memorize larger translation units to replicate the effect of source linearization as done in the TSM model.

We also show that using phrase-based search with MTU N-gram translation models helps to address some of the search problems that are non-trivial to handle when decoding with minimal translation units. An important limitation of the OSM N-gram model is that it does not handle unaligned or discontinuous target MTUs and requires post-processing of the alignment to remove these. Using phrases during search enabled us to make novel changes to the OSM generative story (also applicable to the TSM model) to handle unaligned target words and to use target linearization to deal with discontinuous target MTUs.

We performed an extensive evaluation, carrying out translation experiments from French, Spanish, Czech and Russian to English and in the opposite direction. Our integration of the OSM model into Moses and our modification of the OSM model to deal with unaligned and discontinuous target tokens consistently improves BLEU scores over the

baseline system, and shows statistically significant improvements in seven out of eight cases.

## 2 Previous Work

Several researchers have tried to combine the ideas of phrase-based and N-gram-based SMT. Costajussà et al. (2007) proposed a method for combining the two approaches by applying sentence level reranking. Feng et al. (2010) added a linearized source-side language model in a phrase-based system. Crego and Yvon (2010) modified the phrase-based lexical reordering model of Tillman (2004) for an N-gram-based system. Niehues et al. (2011) integrated a bilingual language model based on surface word forms and POS tags into a phrase-based system. Zhang et al. (2013) explored multiple decomposition structures for generating MTUs in the task of lexical selection, and to rerank the N-best candidate translations in the output of a phrase-based. A drawback of the TSM model is the assumption that source and target information is generated monotonically. The process of reordering is disconnected from lexical generation which restricts the search to a small set of pre-computed reorderings. Durrani et al. (2011) addressed this problem by coupling lexical generation and reordering information into a single generative process and enriching the N-gram models to learn lexical reordering triggers. Durrani et al. (2013) showed that using larger phrasal units during decoding is superior to MTU-based decoding in an N-gram-based system. However, they do not use phrase-based models in their work, relying only on the OSM model. This paper combines insights from these recent pieces of work and show that phrase-based search combined with N-gram-based and phrase-based models in decoding is the overall best way to go. We integrate the two N-gram-based models, TSM and OSM, into phrase-based Moses and show that the translation quality is improved by taking both translation and reordering context into account. Other approaches that explored such models in syntax-based systems used MTUs for sentence level reranking (Khalilov and Fonollosa, 2009), in dependency translation models (Quirk and Menezes, 2006) and in target language syntax systems (Vaswani et al., 2011).

## 3 Integration of N-gram Models

We now describe our integration of TSM and OSM N-gram models into the phrase-based sys-

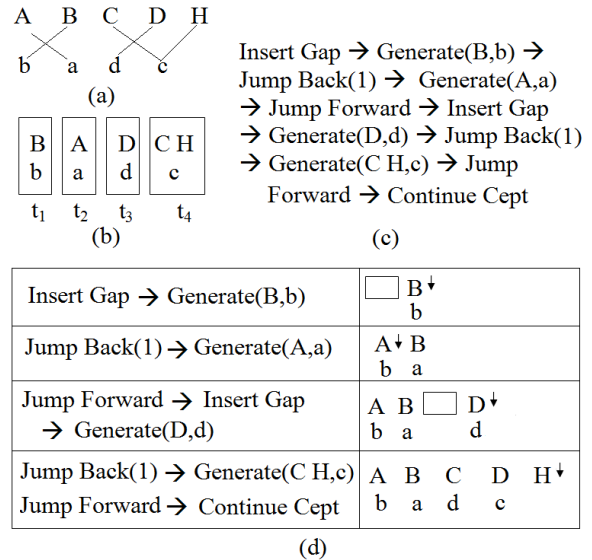


Figure 1: Example (a) Word Alignments (b) Unfolded MTU Sequence (c) Operation Sequence (d) Step-wise Generation

tem. Given a bilingual sentence pair  $(F, E)$  and its alignment  $(A)$ , we first identify minimal translation units (MTUs) from it. An MTU is defined as a translation rule that cannot be broken down any further. The MTUs extracted from Figure 1(a) are  $A \rightarrow a, B \rightarrow b, C \dots H \rightarrow c^1$  and  $D \rightarrow d$ . These units are then generated left-to-right in two different ways, as we will describe next.

### 3.1 Tuple Sequence Model (TSM)

The TSM translation model assumes that MTUs are generated monotonically. To achieve this effect, we enumerate the MTUs in the target left-to-right order. This process is also called source linearization or tuple unfolding. The resulting sequence of monotonic MTUs is shown in Figure 1(b). We then define a TSM model over this sequence  $(t_1, t_2, \dots, t_J)$  as:

$$p_{tsm}(F, E, A) = \prod_{j=1}^J p(t_j | t_{j-n+1}, \dots, t_{j-1})$$

where  $n$  indicates the amount of context used. A 4-gram Kneser-Ney smoothed language model is trained with SRILM (Stolcke, 2002).

**Search:** In previous work, the search graph in TSM N-gram SMT was not built dynamically like in the phrase-based system, but instead constructed as a preprocessing step using POS-based rewrite rules (learned when linearizing the source side). We do not adopt this framework. We use

<sup>1</sup>We use  $\dots$  to denote discontinuous MTUs.



phrase-based search which builds up the decoding graph dynamically and searches through all possible reorderings within a fixed window. During decoding we use the phrase-internal alignments to perform source linearization. For example, if during decoding we would like to apply the phrase pair “C D H – d c”, a combination of  $t_3$  and  $t_4$  in Figure 1(b), then we extract the MTUs from this phrase-pair and linearize the source to be in the order of the target. We then compute the TSM probability given the  $n - 1$  previous MTUs (including MTUs occurring in the previous source phrases). The idea is to replicate rewrite rules with phrase-pairs to linearize the source. Previous work on N-gram-based models restricted the length of the rewrite rules to be 7 or less POS tags. We use phrases of length 6 and less.

### 3.2 Operation Sequence Model (OSM)

The OSM model represents a bilingual sentence pair and its alignment through a sequence of operations that generate the aligned sentence pair. An operation either generates source and target words or it performs reordering by inserting gaps and jumping forward and backward. The MTUs are generated in the target left-to-right order just as in the TSM model. However rather than linearizing the source-side, reordering operations (gaps and jumps) are used to handle crossing alignments. During training, each bilingual sentence pair is deterministically converted to a unique sequence of operations.<sup>2</sup> The example in Figure 1(a) is converted to the sequence of operations shown in Figure 1(c). A step-wise generation of MTUs along with reordering operations is shown in Figure 1(d). We learn a Markov model over a sequence of operations  $(o_1, o_2, \dots, o_J)$  that encapsulate MTUs and reordering information which is defined as follows:

$$p_{osm}(F, E, A) = \prod_{j=1}^J p(o_j | o_{j-n+1}, \dots, o_{j-1})$$

A 9-gram Kneser-Ney smoothed language model is trained with SRILM.<sup>3</sup> By coupling reordering with lexical generation, each (translation or reordering) decision conditions on  $n - 1$  previous (translation and reordering) decisions spanning across phrasal boundaries. The reordering decisions therefore influence lexical selection and

<sup>2</sup>Please refer to Durrani et al. (2011) for a list of operations and the conversion algorithm.

<sup>3</sup>We also tried a 5-gram model, the performance decreased slightly in some cases.

vice versa. A heterogeneous mixture of translation and reordering operations enables the OSM model to memorize reordering patterns and lexicalized triggers unlike the TSM model where translation and reordering are modeled separately.

**Search:** We integrated the generative story of the OSM model into the hypothesis extension process of the phrase-based decoder. Each hypothesis maintains the position of the source word covered by the last generated MTU, the right-most source word generated so far, the number of open gaps and their relative indexes, etc. This information is required to generate the operation sequence for the MTUs in the hypothesized phrase-pair. After the operation sequence is generated, we compute its probability given the previous operations. We define the main OSM feature, and borrow 4 supportive features, the *Gap*, *Open Gap*, *Gap-width* and *Deletion* penalties (Durrani et al., 2011).

### 3.3 Problem: Target Discontinuity and Unaligned Words

Two issues that we have ignored so far are the handling of MTUs which have discontinuous targets, and the handling of unaligned target words. Both TSM and OSM N-gram models generate MTUs linearly in left-to-right order. This assumption becomes problematic in the cases of MTUs that have target-side discontinuities (See Figure 2(a)). The MTU  $A \rightarrow g \dots a$  can not be generated because of the intervening MTUs  $B \rightarrow b, C \dots H \rightarrow c$  and  $D \rightarrow d$ . In the original TSM model, such cases are dealt with by merging all the intervening MTUs to form a bigger unit  $t'_1$  in Figure 2(c). A solution that uses split-rules is proposed by Crego and Yvon (2009) but has not been adopted in Ncode (Crego et al., 2011), the state-of-the-art TSM N-gram system. Durrani et al. (2011) dealt with this problem by applying a post-processing (PP) heuristic that modifies the alignments to remove such cases. When a source word is aligned to a discontinuous target-cept, first the link to the least frequent target word is identified, and the group of links containing this word is retained while the others are deleted. The alignment in Figure 2(a), for example, is transformed to that in Figure 2(b). This allows OSM to extract the intervening MTUs  $t_2 \dots t_5$  (Figure 2(c)). Note that this problem does not exist when dealing with source-side discontinuities: the TSM model linearizes discontinuous source-side MTUs such as  $C \dots H \rightarrow c$ . The

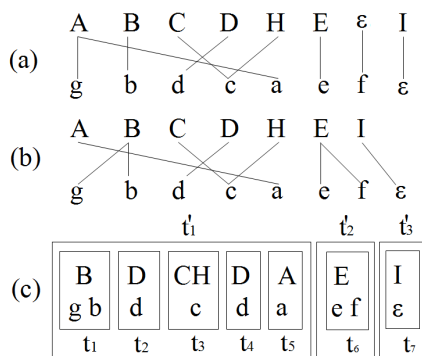


Figure 2: Example (a) Original Alignments (b) Post-Processed Alignments (c) Extracted MTUs –  $t'_1 \dots t'_3$  (from (a)) and  $t_1 \dots t_7$  (from (b))

OSM model deals with such cases through *Insert Gap* and *Continue Cept* operations.

The second problem is the unaligned target-side MTUs such as  $\varepsilon \rightarrow f$  in Figure 2(a). Inserting target-side words “spuriously” during decoding is a non-trivial problem because there is no evidence of when to hypothesize such words. These cases are dealt with in N-gram-based SMT by merging such MTUs to the MTU on the left or right based on attachment counts (Durrani et al., 2011), lexical probabilities obtained from IBM Model 1 (Mariño et al., 2006), or POS entropy (Gispert and Mariño, 2006). Notice how  $\varepsilon \rightarrow f$  (Figure 2(a)) is merged with the neighboring MTU  $E \rightarrow e$  to form a new MTU  $E \rightarrow ef$  (Figure 2(c)). We initially used the post-editing heuristic (PP) as defined by Durrani et al. (2011) for both TSM and OSM N-gram models, but found that it lowers the translation quality (See Row 2 in Table 2) in some language pairs.

### 3.4 Solution: Insertion and Linearization

To deal with these problems, we made novel modifications to the generative story of the OSM model. Rather than merging the unaligned target MTU such as  $\varepsilon - f$ , to its right or left MTU, we generate it through a new *Generate Target Only* ( $f$ ) operation. Orthogonal to its counterpart *Generate Source Only* ( $I$ ) operation (as used for MTU  $t_7$  in Figure 2(c)), this operation is generated as soon as the MTU containing its previous target word is generated. In Figure 2(a),  $\varepsilon - f$  is generated immediately after MTU  $E - e$  is generated. In a sequence of unaligned source and target MTUs, unaligned source MTUs are generated before the unaligned target MTUs. We do not modify the decoder to arbitrarily generate unaligned MTUs but hypothesize these only when they appear within

an extracted phrase-pair. The constraint provided by the phrase-based search makes the *Generate Target Only* operation tractable. Using phrase-based search therefore helps addressing some of the problems that exist in the decoding framework of N-gram SMT.

The remaining problem is the discontinuous target MTUs such as  $A \rightarrow g \dots a$  in Figure 2(a). We handle this with target linearization similar to the TSM source linearization. We collapse the target words  $g$  and  $a$  in the MTU  $A \rightarrow g \dots a$  to occur consecutively when generating the operation sequence. The conversion algorithm that generates the operations thinks that  $g$  and  $a$  occurred adjacently. During decoding we use the phrasal alignments to linearize such MTUs within a phrasal unit. This linearization is done only to compute the OSM feature. Other features in the phrase-based system (e.g., language model) work with the target string in its original order. Notice again how memorizing larger translation units using phrases helps us reproduce such patterns. This is achieved in the tuple N-gram model by using POS-based split and rewrite rules.

## 4 Evaluation

**Corpus:** We ran experiments with data made available for the translation task of the *Eighth Workshop on Statistical Machine Translation*. The sizes of bitext used for the estimation of translation and monolingual language models are reported in Table 1. All data is true-cased.

| Pair  | Parallel | Monolingual | Lang |
|-------|----------|-------------|------|
| fr-en | ≈39 M    | ≈91 M       | fr   |
| cs-en | ≈15.6 M  | ≈43.4 M     | cs   |
| es-en | ≈15.2 M  | ≈65.7 M     | es   |
| ru-en | ≈2 M     | ≈21.7 M     | ru   |
|       |          | ≈287.3 M    | en   |

Table 1: Number of Sentences (in Millions) used for Training

We follow the approach of Schwenk and Koehn (2008) and trained domain-specific language models separately and then linearly interpolated them using SRILM with weights optimized on the held-out dev-set. We concatenated the news-test sets from four years (2008-2011) to obtain a large dev-set in order to obtain more stable weights (Koehn and Haddow, 2012). For Russian-English and English-Russian language pairs, we divided the tuning-set news-test 2012 into two halves and used

| No. | System   | fr-en | es-en        | cs-en        | ru-en        | en-fr        | en-es        | en-cs        | en-ru        |
|-----|----------|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1.  | Baseline | 31.89 | 35.07        | 23.88        | 33.45        | 29.89        | 35.03        | 16.22        | 23.88        |
| 2.  | 1+pp     | 31.87 | 35.09        | 23.64        | 33.04        | 29.70        | 35.00        | 16.17        | 24.05        |
| 3.  | 1+pp+tsm | 31.94 | 35.25        | 23.85        | 32.97        | 29.98        | 35.06        | 16.30        | 23.96        |
| 4.  | 1+pp+osm | 32.17 | <b>35.50</b> | 24.14        | 33.21        | <b>30.35</b> | <b>35.34</b> | 16.49        | <b>24.22</b> |
| 5.  | 1+osm*   | 32.13 | <b>35.65</b> | <b>24.23</b> | <b>33.91</b> | <b>30.54</b> | <b>35.49</b> | <b>16.62</b> | <b>24.25</b> |

Table 2: Translating into and from English. Bold: Statistically Significant (Koehn, 2004) w.r.t Baseline

the first half for tuning and second for test. We test our systems on news-test 2012. We tune with the k-best batch MIRA algorithm (Cherry and Foster, 2012).

**Moses Baseline:** We trained a Moses system (Koehn et al., 2007) with the following settings: maximum sentence length 80, grow-diag-final and symmetrization of GIZA++ alignments, an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield, 2011) used at runtime, msd-bidirectional-fe lexicalized reordering, sparse lexical and domain features (Hasler et al., 2012), distortion limit of 6, 100-best translation options, minimum bayes-risk decoding (Kumar and Byrne, 2004), cube-pruning (Huang and Chiang, 2007) and the no-reordering-over-punctuation heuristic.

**Results:** Table 2 shows uncased BLEU scores (Papineni et al., 2002) on the test set. Row 2 (+pp) shows that the post-editing of alignments to remove unaligned and discontinuous target MTUs decreases the performance in the case of ru-en, cs-en and en-fr. Row 3 (+pp+tsm) shows that our integration of the TSM model slightly improves the BLEU scores for en-fr, and es-en. Results drop in ru-en and en-ru. Row 4 (+pp+osm) shows that the OSM model consistently improves the BLEU scores over the Baseline systems (Row 1) giving significant improvements in half the cases. The only result that is lower than the baseline system is that of the ru-en experiment, because OSM is built with PP alignments which particularly hurt the performance for ru-en. Finally Row 5 (+osm\*) shows that our modifications to the OSM model (Section 3.4) give the best result ranging from [0.24–0.65] with statistically significant improvements in seven out of eight cases. It also shows improvements over Row 4 (+pp+osm) even in some cases where the PP heuristic doesn’t hurt. The largest gains are obtained in the ru-en translation task (where the PP heuristic inflicted maximum damage).

## 5 Conclusion and Future Work

We have addressed the problem of the independence assumption in PBSMT by integrating N-gram-based models inside a phrase-based system using a log-linear framework. We try to replicate the effect of rewrite and split rules as used in the TSM model through phrasal alignments. We presented a novel extension of the OSM model to handle unaligned and discontinuous target MTUs in the OSM model. Phrase-based search helps us to address these problems that are non-trivial to handle in the decoding frameworks of the N-gram-based models. We tested our extensions and modifications by evaluating against a competitive baseline system over 8 language pairs. Our integration of TSM shows small improvements in a few cases. The OSM model which takes both reordering and lexical context into consideration consistently improves the performance of the baseline system. Our modification to the OSM model produces the best results giving significant improvements in most cases. Although our modifications to the OSM model enables discontinuous MTUs, we did not fully utilize these during decoding, as Moses only uses continuous phrases. The discontinuous MTUs that span beyond a phrasal length of 6 words are therefore never hypothesized. We would like to explore this further by extending the search to use discontinuous phrases (Galley and Manning, 2010).

## Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n<sup>o</sup> 287658. Alexander Fraser was funded by Deutsche Forschungsgemeinschaft grant Models of Morphosyntax for Statistical Machine Translation. Helmut Schmid was supported by Deutsche Forschungsgemeinschaft grant SFB 732. This publication only reflects the authors views.

## References

- Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.
- Marta R. Costa-jussà, Josep M. Crego, David Vilar, José A.R. Fonollosa, José B. Mariño, and Hermann Ney. 2007. Analysis and System Combination of Phrase- and N-Gram-Based Statistical Machine Translation Systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 137–140, Rochester, New York, April.
- Josep M. Crego and José B. Mariño. 2006. Improving Statistical MT by Coupling Reordering and Decoding. *Machine Translation*, 20(3):199–215.
- Josep M. Crego and François Yvon. 2009. Gappy Translation Units under Left-to-Right SMT Decoding. In *Proceedings of the Meeting of the European Association for Machine Translation (EAMT)*, pages 66–73, Barcelona, Spain.
- Josep M. Crego and François Yvon. 2010. Improving Reordering with Linguistically Informed Bilingual N-Grams. In *Coling 2010: Posters*, pages 197–205, Beijing, China, August. Coling 2010 Organizing Committee.
- Josep M. Crego, François Yvon, and José B. Mariño. 2011. Ncode: an Open Source Bilingual N-gram SMT Toolkit. *The Prague Bulletin of Mathematical Linguistics*, 96:49–58.
- Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A Joint Sequence Translation Model with Integrated Reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA, June.
- Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model With Minimal Translation Units, But Decode With Phrases. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
- Minwei Feng, Arne Mauser, and Hermann Ney. 2010. A Source-side Decoding Sequence Model for Statistical Machine Translation. In *Conference of the Association for Machine Translation in the Americas 2010*, Denver, Colorado, USA, October.
- Michel Galley and Christopher D. Manning. 2010. Accurate Non-Hierarchical Phrase-Based Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June. Association for Computational Linguistics.
- Adrià Gispert and José B. Mariño. 2006. Linguistic Tuple Segmentation in N-Gram-Based Statistical Machine Translation. In *INTERSPEECH*.
- Eva Hasler, Barry Haddow, and Philipp Koehn. 2012. Sparse Lexicalised Features and Topic Adaptation for SMT. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 268–275.
- Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, 7.
- Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.
- Maxim Khalilov and José A. R. Fonollosa. 2009. N-Gram-Based Statistical Machine Translation Versus Syntax Augmented Machine Translation: Comparison and System Combination. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 424–432, Athens, Greece, March. Association for Computational Linguistics.
- Philipp Koehn and Barry Haddow. 2012. Towards Effective Use of Training Data in Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 317–321, Montréal, Canada, June. Association for Computational Linguistics.
- Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL*, pages 127–133, Edmonton, Canada.
- 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 Demonstrations*, Prague, Czech Republic.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.
- Shankar Kumar and William J. Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *HLT-NAACL*, pages 169–176.

- José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. 2006. N-gram-Based Machine Translation. *Computational Linguistics*, 32(4):527–549.
- Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider Context by Using Bilingual Language Models in Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206, Edinburgh, Scotland, July. Association for Computational Linguistics.
- Franz J. Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(1):417–449.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA.
- Christopher Quirk and Arul Menezes. 2006. Do We Need Phrases? Challenging the Conventional Wisdom in Statistical Machine Translation. In *HLT-NAACL*.
- Holger Schwenk and Philipp Koehn. 2008. Large and Diverse Language Models for Statistical Machine Translation. In *International Joint Conference on Natural Language Processing*, pages 661–666, January 2008.
- Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Intl. Conf. Spoken Language Processing*, Denver, Colorado.
- Christoph Tillman. 2004. A Unigram Orientation Model for Statistical Machine Translation. In *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts.
- Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule Markov Models for Fast Tree-to-String Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 856–864, Portland, Oregon, USA, June.
- Hui Zhang, Kristina Toutanova, Chris Quirk, and Jianfeng Gao. 2013. Beyond Left-to-Right: Multiple Decomposition Structures for SMT. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

# Learning Non-linear Features for Machine Translation Using Gradient Boosting Machines

Kristina Toutanova  
Microsoft Research  
Redmond, WA 98502

kristout@microsoft.com

Byung-Gyu Ahn\*  
Johns Hopkins University  
Baltimore, MD 21218

bahn@cs.jhu.edu

## Abstract

In this paper we show how to automatically induce non-linear features for machine translation. The new features are selected to approximately maximize a BLEU-related objective and decompose on the level of local phrases, which guarantees that the asymptotic complexity of machine translation decoding does not increase. We achieve this by applying gradient boosting machines (Friedman, 2000) to learn new weak learners (features) in the form of regression trees, using a differentiable loss function related to BLEU. Our results indicate that small gains in performance can be achieved using this method but we do not see the dramatic gains observed using feature induction for other important machine learning tasks.

## 1 Introduction

The linear model for machine translation (Och and Ney, 2002) has become the de-facto standard in the field. Recently, researchers have proposed a large number of additional features (TaroWatanabe et al., 2007; Chiang et al., 2009) and parameter tuning methods (Chiang et al., 2008b; Hopkins and May, 2011; Cherry and Foster, 2012) which are better able to scale to the larger parameter space. However, a significant feature engineering effort is still required from practitioners. When a linear model does not fit well, researchers are careful to manually add important feature conjunctions, as for example, (Daumé III and Jagarlamudi, 2011; Clark et al., 2012). In the related field of web search ranking, automatically learned non-linear features have brought dramatic improvements in quality (Burgess et al., 2005; Wu

---

This research was conducted during the author's internship at Microsoft Research

et al., 2010). Here we adapt the main insights of such work to the machine translation setting and share results on two language pairs.

Some recent works have attempted to relax the linearity assumption on MT features (Nguyen et al., 2007), by defining non-parametric models on complete translation hypotheses, for use in an n-best re-ranking setting. In this paper we develop a framework for inducing non-linear features in the form of regression decision trees, which decompose locally and can be integrated efficiently in decoding. The regression trees encode non-linear feature combinations of the original features. We build on the work by Friedman (2000) which shows how to induce features to minimize any differentiable loss function. In our application the features are regression decision trees, and the loss function is the pairwise ranking log-loss from the PRO method for parameter tuning (Hopkins and May, 2011). Additionally, we show how to design the learning process such that the induced features are local on phrase-pairs and their language model and reordering context, and thus can be incorporated in decoding efficiently.

Our results using re-ranking on two language pairs show that the feature induction approach can bring small gains in performance. Overall, even though the method shows some promise, we do not see the dramatic gains that have been seen for the web search ranking task (Wu et al., 2010). Further improvements in the original feature set and the induction algorithm, as well as full integration in decoding are needed to potentially result in substantial performance improvements.

## 2 Feature learning using gradient boosting machines

In the linear model for machine translation, the scores of translation hypotheses are weighted sums of a set of input features over the hypotheses.

|       |  |          |                           |  |
|-------|--|----------|---------------------------|--|
|       | konferenciqta<br>the conference center |          | v Bulgaria<br>in Bulgaria |  |
|       | P1 Local                               | P2 Local | Global                    |  |
| $f_0$ | -1.2                                   | -0.8     | -2.0                      |  |
| $f_1$ | -5.3                                   | -1.3     | -6.6                      |  |
| $f_2$ | 1.0                                    | 3.0      | 4.0                       |  |
| $f_3$ | -7.6                                   | -10.2    | -17.8                     |  |
| $h_1$ | -10.6                                  | -1.3     | -11.9                     |  |
| $h_2$ | .5                                     | .8       | 1.3                       |  |

Figure 1: A Bulgarian source sentence (meaning "the conference in Bulgaria", together with a candidate translation. Local and global features for the translation hypothesis are shown.  $f_0$ =smoothed relative frequency estimate of  $\log p(s|t)$ ;  $f_1$ =lexical weighting estimate of  $\log p(s|t)$ ;  $f_2$ =joint count of the phrase-pair;  $f_3$ =sum of language model log-probabilities of target phrase words given context.

For a set of features  $f_1(h), \dots, f_L(h)$  and weights for these features  $\lambda_1, \dots, \lambda_L$ , the hypothesis scores are defined as:  $F(h) = \sum_{l=1..L} \lambda_l f_l(h)$ . In current state-of-the-art models, the features  $f_l(h)$  decompose locally on phrase-pairs (with language model and reordering context) inside the hypotheses. This enables hypothesis recombination during machine translation decoding, leading to faster and more accurate search. As an example, Figure 1 shows a Bulgarian source sentence (spelled phonetically in Latin script) and a candidate translation. Two phrase-pairs are used to compose the translation, and each phrase-pair has a set of local feature function values. A minimal set of four features is shown, for simplicity. We can see that the hypothesis-level (global) feature values are sums of phrase-level (local) feature values. The score of a translation given feature weights  $\lambda$  can be computed either by scoring the phrase-pairs and adding the scores, or by scoring the complete hypothesis by computing its global feature values. The local feature values do look at some limited context outside of a phrase-pair, to compute language model scores and re-ordering scores; therefore we say that the features are defined on phrase-pairs in context.

We start with such a state-of-the-art linear model with decomposable features and show how we can automatically induce additional features. The new features are also locally decomposable, so that the scores of hypotheses can be computed as sums of phrase-level scores. The new local phrase-level features are non-linear combinations of the original phrase-level features.

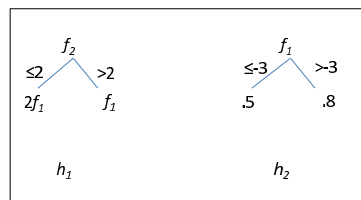


Figure 2: Example of two decision tree features. The left decision tree has linear nodes and the right decision tree has constant nodes.

## 2.1 Form of induced features

We will use the example in Figure 1 to introduce the form of the new features we induce and to give an intuition of why such features might be useful. The new features are expressed by regression decision trees; Figure 2 shows two examples.

One intuition we might have is that, if a phrase pair has been seen very few times in the training corpus (for example, the first phrase pair P1 in the Figure has been seen only one time  $f_2 = 1$ ), we would like to trust its lexical weighting channel model score  $f_1$  more than its smoothed relative-frequency channel estimate  $f_0$ . The first regression tree feature  $h_1$  in Figure 2 captures this intuition. The feature value for a phrase-pair of this feature is computed as follows: if  $f_2 \leq 2$ , then  $h_1(f_0, f_1, f_2, f_3) = 2 \times f_1$ ; otherwise,  $h_1(f_0, f_1, f_2, f_3) = f_1$ . The effect of this new feature  $h_1$  is to boost the importance of the lexical weighting score for phrase-pairs of low joint count. More generally, the regression tree features we consider have either linear or constant leaf nodes, and have up to 8 leaves. Deeper trees can capture more complex conditions on several input feature values. Each non-leaf node performs a comparison of some input feature value to a threshold and each leaf node (for linear nodes) returns the value of some input feature multiplied by some factor. For a given regression tree with linear nodes, all leaf nodes are expressions of the same input feature but have different coefficients for it (for example, both leaf nodes of  $h_1$  return affine functions of the input feature  $f_1$ ). A decision tree feature with constant-valued leaf nodes is illustrated by the right-hand-side tree in Figure 2. For these decision trees, the leaf nodes contain a constant, which is specific to each leaf. These kinds of trees can effectively perform conjunctions of several binary-valued input feature functions; or they can achieve binning of real-values features together with conjunctions over binned values.

Having introduced the form of the new features we learn, we now turn to the methodology for inducing them. We apply the framework of gradient boosting for decision tree weak learners (Friedman, 2000). To define the framework, we need to introduce the original input features, the differentiable loss function, and the details of the tree growing algorithm. We discuss these in turn next.

## 2.2 Initial features

Our baseline MT system uses relative frequency and lexical weighting channel model weights, one or more language models, distortion penalty, word count, phrase count, and multiple lexicalized re-ordering weights, one for each distortion type. We have around 15 features in this **base** feature set. We further expand the input set of features to increase the possibility that useful feature combinations could be found by our feature induction method. The **large** feature set contains around 190 features, including source and target word count features, joint phrase count, lexical weighting scores according to alternative word-alignment model ran over morphemes instead of words, indicator lexicalized features for insertion and deletion of the top 15 words in each language, cluster-based insertion and deletion indicators using hard word clustering, and cluster based signatures of phrase-pairs. This is the feature set we use as a basis for weak learner induction.

## 2.3 Loss function

We use a pair-wise ranking log-loss as in the PRO parameter tuning method (Hopkins and May, 2011). The loss is defined by comparing the model scores of pairs of hypotheses  $h_i$  and  $h_j$  where the BLEU score of the first hypothesis is greater than the BLEU score of the second hypothesis by a specified threshold.<sup>1</sup>

We denote the sentences in a corpus as  $s^1, s^2, \dots, s^N$ . For each sentence  $s^n$ , we denote the ordered selected pairs of hypotheses as  $[h_{i_1}^n, h_{j_1}^n], \dots, [h_{i_K}^n, h_{j_K}^n]$ . The loss-function  $\Psi$  is defined in terms of the hypothesis model scores

<sup>1</sup>In our implementation, for each sentence, we sample 10,000 pairs of translations and accept a pair of translations for use with probability proportional to the BLEU score difference, if that difference is greater than the threshold of 0.04. The top  $K = 100$  or  $K = 300$  hypothesis pairs with the largest BLEU difference are selected for computation of the loss. We compute sentence-level BLEU scores by add- $\alpha$  smoothing of the match counts for computation of n-gram precision. The  $\alpha$  and  $K$  parameters are chosen via cross-validation.

- 1:  $F_0(x) = \arg \min_{\lambda} \Psi(F(x, \lambda))$
- 2: **for**  $m = 1$  **to**  $M$  **do**
- 3:    $y_r = -[\frac{\partial \Psi(F(x))}{\partial F(x_r)}]_{F(x)=F_{m-1}(x)}$ ,  $r = 1 \dots R$
- 4:    $\alpha_m = \arg \min_{\alpha, \beta} \sum_{r=1}^R [y_r - \beta h(x_i; \alpha)]^2$
- 5:    $\rho_m = \arg \min_{\rho} \Psi(F_{m-1}(x) + \rho h(x; \alpha_m))$
- 6:    $F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$
- 7: **end for**

Figure 3: A gradient boosting algorithm for local feature functions.

$F(h)$  as follows:  $\sum_{n=1 \dots N} \sum_{k=1 \dots K} \log(1 + e^{F(h_{j_k}^n) - F(h_{i_k}^n)})$ .

The idea of the gradient boosting method is to induce additional features by computing a functional gradient of the target loss function and iteratively selecting the next weak learner (feature) that is most parallel to the negative gradient. Since we want to induce features such that the hypothesis scores decompose locally, we need to formulate our loss function as a function of local phrase-pair in context scores. Having the model scores decompose locally means that the scores of hypotheses  $F(h)$  decompose as  $F(h) = \sum_{p_r \in h} F(p_r)$ , where by  $p_r \in h$  we denote the enumeration over phrase pairs in context that are parts of  $h$ . If  $x_r$  denotes the input feature vector for a phrase-pair in context  $p_r$ , the score of this phrase-pair can be expressed as  $F(x_r)$ . Appendix A expresses the pairwise log-loss as a function of the phrase scores.

We are now ready to introduce the gradient boosting algorithm, summarized in Figure 3. In the first step of the algorithm, we start by setting the phrase-pair in context scoring function  $F_0(x)$  as a linear function of the input feature values, by selecting the feature weights  $\lambda$  to minimize the PRO loss  $\Psi(F_0(x))$  as a function of  $\lambda$ . The initial scores have the form  $F_0(x) = \sum_{l=1 \dots L} \lambda_l f_l(x)$ . This is equivalent to using the (Hopkins and May, 2011) method of parameter tuning for a fixed input feature set and a linear model. We used LBFGS for the optimization in Line 1. Then we iterate and induce a new decision tree weak learner  $h(x; \alpha_m)$  like the examples in Figure 2 at each iteration. The parameter vectors  $\alpha_m$  encode the topology and parameters of the decision trees, including which feature value is tested at each node, what the comparison cutoffs are, and the way to compute the values at the leaf nodes. After a new decision tree



| Language | Train | Dev-Train | Dev-Select | Test   |
|----------|-------|-----------|------------|--------|
| Chs-En   | 999K  | NIST02+03 | 2K         | NIST05 |
| Fin-En   | 2.2M  | 12K       | 2K         | 4.8K   |

Table 1: Data sets for the two language pairs Chinese-English and Finnish-English.

| Features     | Tune | Chs-En    |              | Fin-En    |       |
|--------------|------|-----------|--------------|-----------|-------|
|              |      | Dev-Train | Test         | Dev-Train | Test  |
| base         | MERT | 31.3      | 30.76        | 49.8      | 51.31 |
| base         | PRO  | 31.1      | 31.16        | 49.7      | 51.56 |
| large        | PRO  | 31.8      | <b>31.44</b> | 49.8      | 51.77 |
| boost-global | PRO  | 31.8      | 31.30        | 50.0      | 51.87 |
| boost-local  | PRO  | 31.8      | <b>31.44</b> | 50.1      | 51.95 |

Table 2: Results for the two language pairs using different weight tuning methods and feature sets.

$h(x; \alpha_m)$  is induced, it is treated as new feature and a linear coefficient  $\rho_m$  for that feature is set by minimizing the loss as a function of this parameter (Line 5). The new model scores are set as the old model scores plus a weighted contribution from the new feature (Line 6). At the end of learning, we have a linear model over the input features and additional decision tree features.  $F_M(x) = \sum_{l=1..L} \lambda_l f_l(x) + \sum_{m=1..M} \rho_m h(x; \alpha_m)$ . The most time-intensive step of the algorithm is the selection of the next decision tree  $h$ . This is done by first computing the functional gradient of the loss with respect to the phrase scores  $F(x_r)$  at the point of the current model scores  $F_{m-1}(x_r)$ . Appendix A shows a derivation of this gradient. We then induce a regression tree using mean-square-error minimization, setting the direction given by the negative gradient as a target to be predicted using the features of each phrase-pair in context instance. This is shown as the setting of the  $\alpha_m$  parameters by mean-squared-error minimization in Line 4 of the algorithm. The minimization is done approximately by a standard greedy tree-growing algorithm (Breiman et al., 1984). When we tune weights to minimize the loss, such as the weights  $\lambda$  of the initial features, or the weights  $\rho_m$  of induced learners, we also include an  $L_2$  penalty on the parameters, to prevent overfitting.

### 3 Experiments

We report experimental results on two language pairs: Chinese-English, and Finnish-English. Table 1 summarizes statistics about the data. For each language pair, we used a training set (Train) for extracting phrase tables and language models, a Dev-Train set for tuning feature weights and inducing features, a Dev-Select set for selecting hyperparameters of PRO tuning and selecting a stop-

ping point and other hyperparameters of the boosting method, and a Test set for reporting final results. For Chinese-English, the training corpus consists of approximately one million sentence pairs from the FBIS and HongKong portions of the LDC data for the NIST MT evaluation and the Dev-Train and Test sets are from NIST competitions. The MT system is a phrasal system with a 4-gram language model, trained on the Xinhua portion of the English Gigaword corpus. The phrase table has maximum phrase length of 7 words on either side. For Finnish-English we used a dataset from a technical domain of software manuals. For this language pair we used two language models: one very large model trained on billions of words, and another language model trained from the target side of the parallel training set. We report performance using the BLEU-SBP metric proposed in (Chiang et al., 2008a). This is a variant of BLEU (Papineni et al., 2002) with strict brevity penalty, where a long translation for one sentence can not be used to counteract the brevity penalty for another sentence with a short translation. Chiang et al. (2008a) showed that this metric overcomes several undesirable properties of BLEU and has better correlation with human judgements. In our experiments with different feature sets and hyperparameters we observed more stable results and better correlation of Dev-Train, Dev-Select, and Test results using BLEU-SBP. For our experiments, we first trained weights for the **base** feature sets described in Section 2.2 using MERT. We then decoded the Dev-Train, Dev-Select, and Test datasets, generating 500-best lists for each set. All results in Table 2 report performance of re-ranking on these 500-best lists using different feature sets and parameter tuning methods.

The baseline (**base** feature set) performance using MERT and PRO tuning on the two language pairs is shown on the first two lines. In line with prior work, PRO tuning achieves a bit lower scores on the tuning set but higher scores on the test set, compared to MERT. The **large** feature set additionally contains over 170 manually specified features, described in Section 2.2. It was infeasible to run MERT training on this feature set. The test set results using PRO tuning for the **large** set are about a quarter of a BLEU-SBP point higher than the results using the **base** feature set on both language pairs. Finally, the last two rows show the performance of the gradient boosting method. In

addition to learning locally decomposable features **boost-local**, we also implemented **boost-global**, where we are learning combinations of the global feature values and lose decomposability. The features learned by **boost-global** can not be computed exactly on partial hypotheses in decoding and thus this method has a speed disadvantage, but we wanted to compare the performance of **boost-local** and **boost-global** on n-best list re-ranking to see the potential accuracy gain of the two methods. We see that **boost-local** is slightly better in performance, in addition to being amenable to efficient decoder integration.

The gradient boosting results are mixed; for Finnish-English, we see around .2 gain of the **boost-local** model over the **large** feature set. There is no improvement on Chinese-English, and the **boost-global** method brings slight degradation. We did not see a large difference in performance among models using different decision tree leaf node types and different maximum numbers of leaf nodes. The selected **boost-local** model for FIN-ENU used trees with maximum of 2 leaf nodes and linear leaf values; 25 new features were induced before performance started to degrade on the Dev-Select set. The induced features for Finnish included combinations of language model and channel model scores, combinations of word count and channel model scores, and combinations of channel and lexicalized reordering scores. For example, one feature increases the contribution of the relative frequency channel score for phrases with many target words, and decreases the channel model contribution for shorter phrases.

The best **boost-local** model for Chs-Enu used trees with a maximum of 2 constant-values leaf nodes, and induced 24 new tree features. The features effectively promoted and demoted phrase-pairs in context based on whether an input feature's value was smaller than a determined cutoff.

In conclusion, we proposed a new method to induce feature combinations for machine translation, which do not increase the decoding complexity. There were small improvements on one language pair in a re-ranking setting. Further improvements in the original feature set and the induction algorithm, as well as full integration in decoding are needed to result in substantial performance improvements.

This work did not consider alternative ways of generating non-linear features, such as taking

products of two or more input features. It would be interesting to compare such alternatives to the regression tree features we explored.

## References

- Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Chapman and Hall.
- Chris Burges, Tal Shaked, Erin Renshaw, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *ICML*.
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *HLT-NAACL*.
- David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008a. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *EMNLP*.
- David Chiang, Yuval Marton, and Philp Resnik. 2008b. Online large margin training of syntactic and structural translation features. In *EMNLP*.
- D. Chiang, W. Wang, and K. Knight. 2009. 11,001 new features for statistical machine translation. In *NAACL*.
- Jonathan Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *AMTA*.
- Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *ACL*.
- Jerome H. Friedman. 2000. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.
- Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *EMNLP*.
- Patrick Nguyen, Milind Mahajan, and Xiaodong He. 2007. Training non-parametric features for statistical machine translation. In *Second Workshop on Statistical Machine Translation*.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.
- Taro Watanabe, Jun Suzuki, Hajime Tsukuda, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *EMNLP*.

Qiang Wu, Christopher J. Burges, Krysta M. Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3), June.

#### 4 Appendix A: Derivation of derivatives

Here we express the loss as a function of phrase-level in context scores and derive the derivative of the loss with respect to these scores.

Let us number all phrase-pairs in context in all hypotheses in all sentences as  $p_1, \dots, p_R$  and denote their input feature vectors as  $\mathbf{x}_1, \dots, \mathbf{x}_R$ . We will use  $F(p_r)$  and  $F(x_r)$  interchangeably, because the score of a phrase-pair in context is defined by its input feature vector. The loss  $\Psi(F(x_r))$  is expressed as follows:

$$\sum_{n=1}^N \sum_{k=1}^K \log(1 + e^{\sum_{p_r \in h_{jk}^n} F(x_r) - \sum_{p_r \in h_{ik}^n} F(x_r)}).$$

Next we derive the derivatives of the loss  $\Psi(F(x))$  with respect to the phrase scores. Intuitively, we are treating the scores we want to learn as parameters for the loss function; thus the loss function has a huge number of parameters, one for each instance of each phrase pair in context in each translation. We ask the loss function if these scores could be set in an arbitrary way, what direction it would like to move them in to be minimized. This is the direction given by the negative gradient.

Each phrase-pair in context  $p_r$  occurs in exactly one hypothesis  $h$  in one sentence. It is possible that two phrase-pairs in context share the same set of input features, but for ease of implementation and exposition, we treat these as different training instances. To express the gradient with respect to  $F(x_r)$  we therefore need to focus on the terms of the loss from a single sentence and to take into account the hypothesis pairs  $[h_{j,k}, h_{i,k}]$  where the left or the right hypothesis is the hypothesis  $h$  containing our focus phrase pair  $p_r$ .  $\frac{\partial \Psi(F(x))}{\partial F(x_r)}$  is expressed as:

$$\begin{aligned} &= \sum_{k:h=h_{ik}} \frac{e^{\sum_{p_r \in h_{jk}^n} F(x_r) - \sum_{p_r \in h_{ik}^n} F(x_r)}}{1 + e^{\sum_{p_r \in h_{jk}^n} F(x_r) - \sum_{p_r \in h_{ik}^n} F(x_r)}} \\ &+ \sum_{k:h=h_{jk}} \frac{e^{\sum_{p_r \in h_{jk}^n} F(x_r) - \sum_{p_r \in h_{ik}^n} F(x_r)}}{1 + e^{\sum_{p_r \in h_{jk}^n} F(x_r) - \sum_{p_r \in h_{ik}^n} F(x_r)}} \end{aligned}$$

Since in the boosting step we induce a decision tree to fit the negative gradient, we can see that the feature induction algorithm is trying to increase the scores of phrases that occur in better

hypotheses (the first hypothesis in each pair), and it increases the scores more if weaker hypotheses have higher advantage; it is also trying to decrease the scores of phrases in weaker hypotheses that are currently receiving high scores.

# Language Independent Connectivity Strength Features for Phrase Pivot Statistical Machine Translation

Ahmed El Kholly, 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

An important challenge to statistical machine translation (SMT) is the lack of parallel data for many language pairs. One common solution is to pivot through a third language for which there exist parallel corpora with the source and target languages. Although pivoting is a robust technique, it introduces some low quality translations. In this paper, we present two language-independent features to improve the quality of phrase-pivot based SMT. The features, source connectivity strength and target connectivity strength reflect the quality of projected alignments between the source and target phrases in the pivot phrase table. We show positive results (0.6 BLEU points) on Persian-Arabic SMT as a case study.

## 1 Introduction

One of the main issues 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. 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 main issues of

this technique is that the size of the newly created pivot phrase table is very large (Utiyama and Isahara, 2007). Moreover, many of the produced phrase pairs are of low quality which affects the translation choices during decoding and the overall translation quality. In this paper, we introduce language independent features to determine the quality of the pivot phrase pairs between source and target. We show positive results (0.6 BLEU points) on Persian-Arabic SMT.

Next, we briefly discuss some related work. We then review two common pivoting strategies and how we use them in Section 3. This is followed by our approach to using connectivity strength features in Section 4. We present our experimental results in Section 5.

## 2 Related Work

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 main 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 pivot strategies have been presented in the literature. The following three are perhaps the most common.

The first strategy is the sentence translation technique 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 based on 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. Lexical weights and translation probabilities are computed from the two translation models.

The third 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).

In this paper, we build on the phrase pivoting approach, which has been shown to be the best with comparable settings (Utiyama and Isahara, 2007). We extend phrase table scores with two other features that are language independent.

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 Ofizer, 2010; Elming and Habash, 2009; El Kholly and Habash, 2010a; Habash and Sadat, 2006; Kathol and Zheng, 2008). Most of these efforts are focused on syntactic and morphological processing to improve the quality of translation.

To our knowledge, there hasn't been a lot of work on Persian and Arabic as a language pair. The only effort that we are aware of is based on improving the reordering models for Persian-Arabic SMT (Matusov and Köprü, 2010).

### 3 Pivoting Strategies

In this section, we review the two pivoting strategies that are our baselines. We also discuss how we overcome the large expansion of source-to-target phrase pairs in the process of creating a pivot phrase table.

#### 3.1 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.

#### 3.2 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 on top of Moses phrase-based SMT (Koehn et al., 2007), we need to provide the same set of phrase translation probability distributions.<sup>1</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 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 weights 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>2</sup>

We introduce a basic filtering technique discussed next to address this issue and present some baseline experiments to test its performance in Section 5.3.

#### 3.3 Filtering for Phrase Pivoting

The main idea of the filtering process is to select the top  $[n]$  English candidate phrases for each Persian phrase from the Persian-English phrase table and similarly select the top  $[n]$  Arabic target phrases for each English phrase from the English-Arabic phrase table and then perform the pivoting process described earlier to create a pivoted

<sup>1</sup>Four different phrase translation scores are computed in Moses' phrase tables: two lexical weighting scores and two phrase translation probabilities.

<sup>2</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.

Persian-Arabic phrase table. To select the top candidates, we first rank all the candidates based on the log linear scores computed from the phrase translation probabilities and lexical weights multiplied by the optimized decoding weights then we pick the top  $[n]$  pairs.

We compare the different pivoting strategies and various filtering thresholds in Section 5.3.

## 4 Approach

One of the main challenges in phrase pivoting is the very large size of the induced phrase table. It becomes even more challenging if either the source or target language is morphologically rich. The number of translation candidates (fanout) increases due to ambiguity and richness (discussed in more details in Section 5.2) which in return increases the number of combinations between source and target phrases. Since the only criteria of matching between the source and target phrase is through a pivot phrase, many of the induced phrase pairs are of low quality. These phrase pairs unnecessarily increase the search space and hurt the overall quality of translation.

To solve this problem, we introduce two language-independent features which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. We call these features *connectivity strength features*.

**Connectivity Strength Features** provide two scores, Source Connectivity Strength (SCS) and Target Connectivity Strength (TCS). These two scores are similar to precision and recall metrics. They depend on the number of alignment links between words in the source phrase to words of the target phrase. SCS and TCS are defined in equations 1 and 2 where  $\mathcal{S} = \{i : 1 \leq i \leq S\}$  is the set of source words in a given phrase pair in the pivot phrase table and  $\mathcal{T} = \{j : 1 \leq j \leq T\}$  is the set of the equivalent target words. The word alignment between  $\mathcal{S}$  and  $\mathcal{T}$  is defined as

$$\mathcal{A} = \{(i, j) : i \in \mathcal{S} \text{ and } j \in \mathcal{T}\}.$$

$$SCS = \frac{|\mathcal{A}|}{|\mathcal{S}|} \quad (1)$$

$$TCS = \frac{|\mathcal{A}|}{|\mathcal{T}|} \quad (2)$$

We get the alignment links by projecting the alignments of source-pivot to the pivot-target phrase pairs used in pivoting. If the source-target phrase pair are connected through more than one pivot phrase, we take the union of the alignments.

In contrast to the aggregated values represented in the lexical weights and the phrase probabilities, connectivity strength features provide additional information by counting the actual links between the source and target phrases. They provide an independent and direct approach to measure how good or bad a given phrase pair are connected.

Figure 1 and 2 are two examples (one good, one bad) Persian-Arabic phrase pairs in a pivot phrase table induced by pivoting through English.<sup>3</sup> In the first example, each Persian word is aligned to an Arabic word. The meaning is preserved in both phrases which is reflected in the SCS and TCS scores. In the second example, only one Persian word in aligned to one Arabic word in the equivalent phrase and the two phrases conveys two different meanings. The English phrase is not a good translation for either, which leads to this bad pairing. This is reflected in the SCS and TCS scores.

## 5 Experiments

In this section, we present a set of baseline experiments including a simple filtering technique to overcome the huge expansion of the pivot phrase table. Then we present our results in using connectivity strength features to improve Persian-Arabic pivot translation quality.

<sup>3</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).

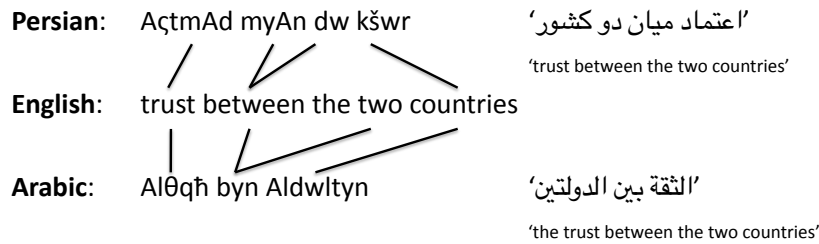


Figure 1: An example of strongly connected Persian-Arabic phrase pair through English. All Persian words are connected to one or more Arabic words. SCS=1.0 and TCS=1.0.

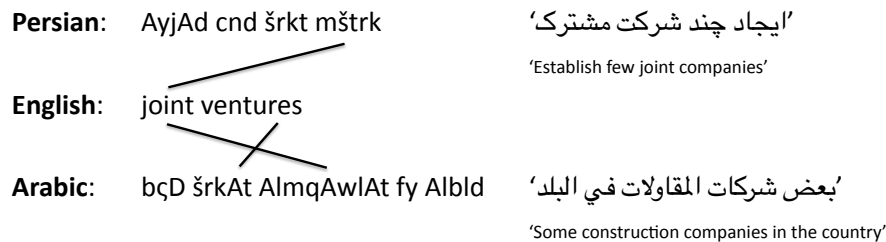


Figure 2: An example of weakly connected Persian-Arabic phrase pairs through English. Only one Persian word is connected to an Arabic word. SCS=0.25 and TCS=0.2.

## 5.1 Experimental Setup

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 ( $\approx 60$ M words) available from LDC<sup>4</sup> and GALE<sup>5</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 English Gigaword Corpus with 5-gram LM using the KenLM toolkit (Heafield, 2011).

All experiments are conducted using the Moses phrase-based SMT system (Koehn et al., 2007). We use MERT (Och, 2003) for decoding weight

<sup>4</sup>LDC Catalog IDs: LDC2005E83, LDC2006E24, LDC2006E34, LDC2006E85, LDC2006E92, LDC2006G05, LDC2007E06, LDC2007E101, LDC2007E103, LDC2007E46, LDC2007E86, LDC2008E40, LDC2008E56, LDC2008G05, LDC2009E16, LDC2009G01.

<sup>5</sup>Global Autonomous Language Exploitation, or GALE, is a DARPA-funded research project.

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 (MT04). The optimized weights are used for ranking and filtering (discussed in Section 3.3).

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) and METEOR (Lavie and Agarwal, 2007).

## 5.2 Linguistic Preprocessing

In this section we present our motivation and choice for preprocessing Arabic, Persian, 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.

One aspect of Arabic’s complexity is its various attachable clitics and numerous morphological features (Habash, 2010). We follow El Kholy and Habash (2010a) and use the PATB tokenization scheme (Maamouri et al., 2004) in our

experiments. 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  $\text{ها}$  + *hA* and sometimes  $\text{ان}$  + *An*, or one of the Arabic plural markers. 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., 2013). We use Perstem (Jadidinejad et al., 2010) for segmenting Persian text.

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”.

### 5.3 Baseline Evaluation

We compare the performance of sentence pivoting against phrase pivoting with different filtering thresholds. The results are presented in Table 2. 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        | METEOR      |
|-------------------|-------------|-------------|
| Sentence Pivoting | 19.2        | 36.4        |
| Phrase_Pivot_F100 | 19.4        | 37.4        |
| Phrase_Pivot_F500 | 20.1        | 38.1        |
| Phrase_Pivot_F1K  | <b>20.5</b> | <b>38.6</b> |

Table 2: Sentence pivoting versus phrase pivoting with different filtering thresholds (100/500/1000).

We use the best performing setup across the rest of the experiments.

### 5.4 Connectivity Strength Features Evaluation

In this experiment, we test the performance of adding the connectivity strength features (+*Conn*) to the best performing phrase pivoting model (*Phrase\_Pivot\_F1K*).

| Model                 | BLEU        | METEOR      |
|-----------------------|-------------|-------------|
| Sentence Pivoting     | 19.2        | 36.4        |
| Phrase_Pivot_F1K      | 20.5        | 38.6        |
| Phrase_Pivot_F1K+Conn | <b>21.1</b> | <b>38.9</b> |

Table 3: Connectivity strength features experiment result.

The results in Table 3 show that we get a nice improvement of  $\approx 0.6/0.5$  (BLEU/METEOR) points by adding the connectivity strength features. The differences in BLEU scores between this setup and all other systems are statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling (Koehn, 2004).

## 6 Conclusion and Future Work

We presented an experiment showing the effect of using two language independent features, source connectivity score and target connectivity score, to improve the quality of pivot-based SMT. We showed that these features help improving the overall translation quality. In the future, we plan to explore other features, e.g., the number of the pivot phrases used in connecting the source and target phrase pair and the similarity between these pivot phrases. We also plan to explore language specific features which could be extracted from some seed parallel data, e.g., syntactic and morphological compatibility of the source and target phrase pairs.

### 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 and Jon Dehdari for helpful discussions and insights into Persian. We also thank the anonymous reviewers for their insightful comments.



## References

- Nicola Bertoldi, Madalina Barbaiani, Marcello Federico, and Roldano Cattoni. 2008. Phrase-based statistical machine translation with pivot languages. *Proceeding of IWSLT*, pages 143–149.
- Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 728.
- Ahmed El Kholly and Nizar Habash. 2010a. Orthographic and Morphological Processing for English-Arabic Statistical Machine Translation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-10)*. Montréal, Canada.
- Ahmed El Kholly and Nizar Habash. 2010b. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.
- Jakob Elming and Nizar Habash. 2009. Syntactic Reordering for English-Arabic Phrase-Based Machine Translation. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69–77, Athens, Greece, March.
- David Graff. 2007. Arabic Gigaword 3, LDC Catalog No.: LDC2003T40. Linguistic Data Consortium, University of Pennsylvania.
- Nizar Habash and Jun Hu. 2009. Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece, March.
- Nizar Habash and Owen Rambow. 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.
- Nizar Habash and Fatiha Sadat. 2006. Arabic Pre-processing Schemes for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52, New York City, USA.
- 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, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. The MEDAR Consortium, April.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Jan Hajič, Jan Hric, and Vladislav Kubon. 2000. Machine Translation of Very Close Languages. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'2000)*, pages 7–12, Seattle.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, UK.
- Carlos Henriquez, Rafael E. Banchs, and José B. Mariño. 2010. Learning reordering models for statistical machine translation with a pivot language.
- Amir Hossein Jadidnejad, 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*, pages 98–101.
- Andreas Kathol and Jing Zheng. 2008. Strategies for building a Farsi-English smt system from limited resources. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH2008)*, pages 2731–2734, Brisbane, Australia.
- 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 *International Workshop on Spoken Language Translation. IWSLT 2008*, pg. 116–123.
- 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 *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.
- Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 2009. 462 machine translation systems for europe. *Proceedings of MT Summit XII*, pages 65–72.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04)*, Barcelona, Spain.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic.
- Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus.

- In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.
- Evgeny Matusov and Selçuk Köprü. 2010. Improving reordering in statistical machine translation from farsi. In *AMTA The Ninth Conference of the Association for Machine Translation in the Americas*, Denver, Colorado, USA.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.
- Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. 2013. Development of a Persian syntactic dependency treebank. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, Atlanta, USA.
- Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO.
- Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York, April. Association for Computational Linguistics.
- Hua Wu and Haifeng Wang. 2009. Revisiting pivot language approach 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*, pages 154–162, Suntec, Singapore, August. Association for Computational Linguistics.
- Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from english to turkish. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 454–464, Uppsala, Sweden, July. Association for Computational Linguistics.

# Semantic Roles for String to Tree Machine Translation

Marzieh Bazrafshan and Daniel Gildea

Department of Computer Science  
University of Rochester  
Rochester, NY 14627

## Abstract

We experiment with adding semantic role information to a string-to-tree machine translation system based on the rule extraction procedure of Galley et al. (2004). We compare methods based on augmenting the set of nonterminals by adding semantic role labels, and altering the rule extraction process to produce a separate set of rules for each predicate that encompass its entire predicate-argument structure. Our results demonstrate that the second approach is effective in increasing the quality of translations.

## 1 Introduction

Statistical machine translation (SMT) has made considerable advances in using syntactic properties of languages in both the training and the decoding of translation systems. Over the past few years, many researchers have started to realize that incorporating semantic features of languages can also be effective in increasing the quality of translations, as they can model relationships that often are not derivable from syntactic structures.

Wu and Fung (2009) demonstrated the promise of using features based on semantic predicate-argument structure in machine translation, using these features to re-rank machine translation output. In general, re-ranking approaches are limited by the set of translation hypotheses, leading to a desire to incorporate semantic features into the translation model used during MT decoding.

Liu and Gildea (2010) introduced two types of semantic features for tree-to-string machine translation. These features model the reorderings and deletions of the semantic roles in the source sentence during decoding. They showed that addition of these semantic features helps improve the quality of translations. Since tree-to-string systems are

trained on parse trees, they are constrained by the tree structures and are generally outperformed by string-to-tree systems.

Xiong et al. (2012) integrated two discriminative feature-based models into a phrase-based SMT system, which used the semantic predicate-argument structure of the source language. Their first model defined features based on the context of a verbal predicate, to predict the target translation for that verb. Their second model predicted the reordering direction between a predicate and its arguments from the source to the target sentence.

Wu et al. (2010) use a head-driven phrase structure grammar (HPSG) parser to add semantic representations to their translation rules.

In this paper, we use semantic role labels to enrich a string-to-tree translation system, and show that this approach can increase the BLEU (Papineni et al., 2002) score of the translations. We extract GHKM-style (Galley et al., 2004) translation rules from training data where the target side has been parsed and labeled with semantic roles. Our general method of adding information to the syntactic tree is similar to the “tree grafting” approach of Baker et al. (2010), although we focus on predicate-argument structure, rather than named entity tags and modality. We modify the rule extraction procedure of Galley et al. (2004) to produce rules representing the overall predicate-argument structure of each verb, allowing us to model alternations in the mapping from syntax to semantics of the type described by Levin (1993).

## 2 Semantic Roles for String-to-Tree Translation

### 2.1 Semantic Role Labeling

Semantic Role Labeling (SRL) is the task of identifying the arguments of the predicates in a sentence, and classifying them into different argument labels. Semantic roles can provide a level

of understanding that cannot be derived from syntactic analysis of a sentence. For example, in sentences “*Ali opened the door.*” and “*The door opened*”, the word *door* has two different syntactic roles but only one semantic role in the two sentences.

Semantic arguments can be classified into core and non-core arguments (Palmer et al., 2010). Core arguments are necessary for understanding the sentence. Non-core arguments add more information about the predicate but are not essential.

Automatic semantic role labelers have been developed by training classifiers on hand annotated data (Gildea and Jurafsky, 2000; Srikumar and Roth, 2011; Toutanova et al., 2005; Fürstenaу and Lapata, 2012). State-of-the-art semantic role labelers can predict the labels with accuracies of around 90%.

## 2.2 String-to-Tree Translation

We adopt the GHKM framework of Galley et al. (2004) using the parses produced by the split-merge parser of Petrov et al. (2006) as the English trees. As shown by Wang et al. (2010), the refined nonterminals produced by the split-merge method can aid machine translation. Furthermore, in all of our experiments, we exclude unary rules during extraction by ensuring that no rules will have the same span in the source side (Chung et al., 2011).

## 2.3 Using Semantic Role Labels in SMT

To incorporate semantic information into a string-to-tree SMT system, we tried two approaches:

- Using semantically enriched GHKM rules, and
- Extracting semantic rules separately from the regular GHKM rules, and adding a new feature for distinguishing the semantic rules.

The next two sections will explain these two methods in detail.

## 2.4 Semantically Enriched Rules (Method 1)

In this method, we tag the target trees in the training corpus with semantic role labels, and extract the translation rules from the tagged corpus. Since the SCFG rule extraction methods do not assume any specific set of non-terminals for the target parse trees, we can attach the semantic roles of each constituent to its label in the tree, and use

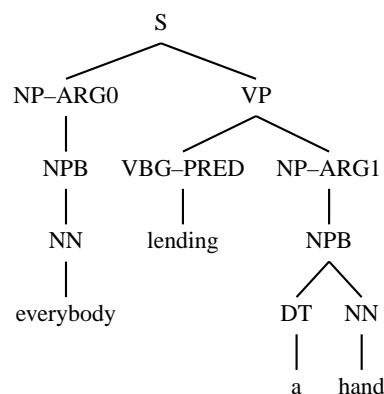


Figure 1: A target tree after inserting semantic roles. “Lending” is the predicate, “everybody” is argument 0, and “a hand” is argument 1 for the predicate.

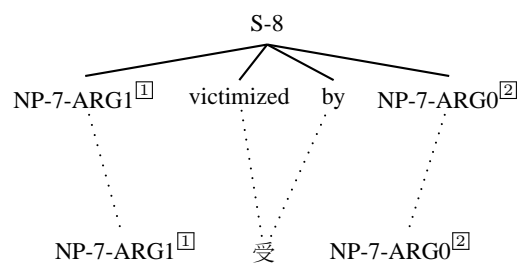


Figure 2: A complete semantic rule.

these new labels for rule extraction. We only label the core arguments of each predicate, to make sure that the rules are not too specific to the training data. We attach each semantic label to the root of the subtree that it is labeling. Figure 1 shows an example target tree after attaching the semantic roles. We then run a GHKM rule extractor on the labeled training corpus and use the semantically enriched rules with a syntax-based decoder.

## 2.5 Complete Semantic Rules with Added Feature (Method 2)

This approach uses the semantic role labels to extract a set of special translation rules, that on the target side form the smallest tree fragments in which one predicate and all of its core arguments are present. These rules model the complete semantic structure of each predicate, and are used by the decoder in addition to the normal GHKM rules, which are extracted separately.

Starting by semantic role labeling the target parse trees, we modify the GHKM component of the system to extract a semantic rule for each predicate. We define  $labels_p$  as the set of semantic role labels related to predicate  $p$ . That includes all

|          | Number of rules |         |
|----------|-----------------|---------|
|          | dev             | test    |
| Baseline | 1292175         | 1300589 |
| Method 1 | 1340314         | 1349070 |
| Method 2 | 1416491         | 1426159 |

Table 1: The number of the translation rules used by the three experimented methods

of the labels of the arguments of  $p$ , and the label of  $p$  itself. Then we add the following condition to the definition of the “frontier node” defined in Galley et al. (2004):

*A frontier node must have either all or none of the semantic role labels from labels $_p$  in its descendants in the tree.*

Adding this new condition, we extract one semantic rule for each predicate, and for that rule we discard the labels related to the other predicates. This semantic rule will then have on its target side, the smallest tree fragment that contains all of the arguments of predicate  $p$  and the predicate itself.

Figure 2 depicts an example of a complete semantic rule. Numbers following grammatical categories (for example, S-8 at the root) are the refined nonterminals produced by the split-merge parser. In general, the tree side of the rule may extend below the nodes with semantic role labels because of the general constraint on frontier nodes that they must have a continuous span in the source (Chinese) side. Also, the internal nodes of the rules (such as a node with PRED label in Figure 2) are removed because they are not used in decoding.

We also extract the regular GHKM rules using the original definition of the frontier nodes, and add the semantic rules to them. To differentiate the semantic rules from the non-semantic ones, we add a new binary feature that is set to 1 for the semantic rules and to 0 for the rest of the rules.

### 3 Experiments

Semantic role labeling was done using the Prop-Bank standard (Palmer et al., 2005). Our labeler uses a maximum entropy classifier and for identification and classification of semantic roles, and has a precision of 90% and a recall of 88%. The features used for training the labeler are a subset of the features used by Gildea and Jurafsky (2000), Xue and Palmer (2004), and Pradhan et al. (2004).

The string-to-tree training data that we used is a Chinese to English parallel corpus that contains

more than 250K sentence pairs, which consist of 6.3M English words. The corpus was drawn from the newswire texts available from LDC.<sup>1</sup> We used a 392-sentence development set with four references for parameter tuning, and a 428-sentence test set with four references for testing. They are drawn from the newswire portion of NIST evaluation (2004, 2005, 2006). The development set and the test set only had sentences with less than 30 words for decoding speed. A set of nine standard features, which include globally normalized count of rules, lexical weighting (Koehn et al., 2003), length penalty, and number of rules used, was used for the experiments. In all of our experiments, we used the split-merge parsing method of Petrov et al. on the training corpus, and mapped the semantic roles from the original trees to the result of the split-merge parser. We used a syntax-based decoder with Earley parsing and cube pruning (Chiang, 2007). We used the Minimum Error Rate Training (Och, 2003) to tune the decoding parameters for the development set and tested the best weights that were found on the test set.

We ran three sets of experiments: Baseline experiments, where we did not do any semantic role labeling prior to rule extraction and only extracted regular GHKM rules, experiments with our method of Section 2.4 (Method 1), and a set of experiments with our method of Section 2.5 (Method 2).

Table 1 contains the numbers of the GHKM translation rules used by our three method. The rules were filtered by the development and the test to increase the decoding speed. The increases in the number of rules were expected, but they were not big enough to significantly change the performance of the decoder.

### 3.1 Results

For every set of experiments, we ran MERT on the development set with 8 different starting weight vectors picked randomly. For Method 2 we added a new random weight for the new feature. We then tested the system on the test set, using for each experiment the weight vector from the iteration of MERT with the maximum BLEU score on the development set. Table 3 shows the BLEU scores that we found on the test set, and their corresponding scores on the development set.

<sup>1</sup>We randomly sampled our data from various different sources. The language model is trained on the English side of entire data (1.65M sentences, which is 39.3M words.)

|           |   |
|-----------|---|
| Source    | 解决 13 亿人的问题, 不能靠别人, 只能靠自己.  |
| Reference | to solve the problem of 1.3 billion people, we can only rely on ourselves and nobody else.  |
| Baseline  | cannot rely on others, can only resolve the problem of 13 billion people, on their own.   |
| Method 2  | to resolve the issue of 1.3 billion people, they can't rely on others, and it can only rely on themselves.                              |
| Source    | 在新世纪新形势下, 亚洲的发展面临着新的机遇.   |
| Reference | in the new situation of the millennium, the development of asia is facing new opportunities.  |
| Baseline  | facing new opportunities in the new situation in the new century, the development of asia.  |
| Method 2  | under the new situation in the new century, the development of asia are facing a new opportunity.                                       |
| Source    | 他说, 阿盟是同美国讨论中东地区进行民主改革的最佳伙伴.  |
| Reference | he said the arab league is the best partner to discuss with the united states about carrying out democratic reforms in the middle east. |
| Baseline  | arab league is the best with democratic reform in the middle east region in the discussion of the united states, he said.               |
| Method 2  | arab league is the best partner to discuss the middle east region democratic reform with the united states, he said.                    |

Table 2: Comparison of example translations from the baseline method and our Method 2.

The best BLEU score on the test set is 25.92, which is from the experiments of Method 2. Method 1 system seems to behave slightly worse than the baseline and Method 2. The reason for this behavior is that the rules that were extracted from the semantic role labeled corpus could have isolated semantic roles in them which would not necessarily get connected to the right predicate or argument during decoding. In other words, it is possible for a rule to only contain one or some of the semantic arguments of a predicate, and not even include the predicate itself, and therefore there is no guarantee that the predicate will be translated with the right arguments and in the right order. The difference between the BLEU scores of the best Method 2 results and the baseline is 0.92. This improvement is statistically significant ( $p = 0.032$ ) and it shows that incorporating semantic roles in machine translation is an effective approach.

Table 2 compares some translations from the baseline decoder and our Method 2. The first line of each example is the Chinese source sentence, and the second line is one of the reference translations. The last two lines compare the baseline and Method 2. These examples show how our Method 2 can outperform the baseline method, by translating complete semantic structures, and generating the semantic roles in the correct order in the target language. In the first example, the predicate *rely on* for the argument *themselves* was not translated by the baseline decoder, but it was correctly translated by Method 2. The second example is a case where the baseline method generated the arguments in the wrong order (in the case of *facing* and *development*), but the translation by Method 2 has the correct order. In the last example we see that the arguments of the predicate *discuss* have the wrong order in the baseline translation,

|          | BLEU Score |              |
|----------|------------|--------------|
|          | dev        | test         |
| Baseline | 26.01      | 25.00        |
| Method 1 | 26.12      | 24.84        |
| Method 2 | 26.5       | <b>25.92</b> |

Table 3: BLEU scores on the test and development sets, of 8 experiments with random initial feature weights.

but Method 2 generated the correct order.

## 4 Conclusion

We proposed two methods for incorporating semantic role labels in a string-to-tree machine translation system, by learning translation rules that are semantically enriched. In one approach, the system learned the translation rules by using a semantic role labeled corpus and augmenting the set of nonterminals used in the rules, and in the second approach, in addition to the regular SCFG rules, the system learned semantic roles which contained the complete semantic structure of a predicate, and added a feature to distinguish those rules.

The first approach did not perform any better than the baseline, which we explained as being due to having rules with only partial semantic structures and not having a way to guarantee that those rules will be used with each other in the right way. The second approach significantly outperformed the baseline of our experiments, which shows that complete predicate-argument structures can improve the quality of machine translation.

**Acknowledgments** Partially funded by NSF grant IIS-0910611.

## References

- Kathryn Baker, Michael Bloodgood, Chris Callison-Burch, Bonnie J. Dorr, Nathaniel W. Filardo, Lori Levin, Scott Miller, and Christine Piatko. 2010. Semantically-informed machine translation: A tree-grafting approach. In *Proceedings of The Ninth Biennial Conference of the Association for Machine Translation in the Americas*, Denver, Colorado.
- David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- Tagyoung Chung, Licheng Fang, and Daniel Gildea. 2011. Issues concerning decoding with synchronous context-free grammar. In *Proceedings of the ACL 2011 Conference Short Papers*, Portland, Oregon. Association for Computational Linguistics.
- Hagen Fürstenaу and Mirella Lapata. 2012. Semi-supervised semantic role labeling via structural alignment. *Computational Linguistics*, 38(1):135–171.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proceedings of NAACL-04*, pages 273–280, Boston.
- Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of ACL-00*, pages 512–520, Hong Kong, October.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL-03*, pages 48–54, Edmonton, Alberta.
- Beth Levin. 1993. *English Verb Classes And Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.
- Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *COLING-10*, Beijing.
- Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL-03*, pages 160–167.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*. Synthesis Lectures on Human Language Technology Series. Morgan and Claypool.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL-02*, 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 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.
- Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of NAACL-04*.
- V. Srikumar and D. Roth. 2011. A joint model for extended semantic role labeling. In *EMNLP*, Edinburgh, Scotland.
- Kristina Toutanova, Aria Haghighi, and Christopher Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of ACL-05*, pages 589–596.
- Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36:247–277, June.
- Dekai Wu and Pascale Fung. 2009. Semantic roles for smt: A hybrid two-pass model. In *Proceedings of the HLT-NAACL 2009: Short Papers*, Boulder, Colorado.
- Xianchao Wu, Takuya Matsuzaki, and Jun’ichi Tsujii. 2010. Fine-grained tree-to-string translation rule extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *ACL (1)*, pages 902–911.
- Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP*.

# Minimum Bayes Risk based Answer Re-ranking for Question Answering

Nan Duan

Natural Language Computing

Microsoft Research Asia

nanduan@microsoft.com

## Abstract

This paper presents two minimum Bayes risk (MBR) based Answer Re-ranking (MBRAR) approaches for the question answering (QA) task. The first approach re-ranks single QA system's outputs by using a traditional MBR model, by measuring correlations between answer candidates; while the second approach re-ranks the combined outputs of multiple QA systems with heterogeneous answer extraction components by using a mixture model-based MBR model. Evaluations are performed on factoid questions selected from two different domains: Jeopardy! and Web, and significant improvements are achieved on all data sets.

## 1 Introduction

Minimum Bayes Risk (MBR) techniques have been successfully applied to a wide range of natural language processing tasks, such as statistical machine translation (Kumar and Byrne, 2004), automatic speech recognition (Goel and Byrne, 2000), parsing (Titov and Henderson, 2006), etc. This work makes further exploration along this line of research, by applying MBR technique to question answering (QA).

The function of a typical factoid question answering system is to automatically give answers to questions in most cases asking about entities, which usually consists of three key components: question understanding, passage retrieval, and answer extraction. In this paper, we propose two *MBR-based Answer Re-ranking (MBRAR)* approaches, aiming to re-rank answer candidates from either single and multiple QA systems. The first one re-ranks answer outputs from single QA system based on a traditional MBR model by measuring the correlations between each answer candidates

and all the other candidates; while the second one re-ranks the combined answer outputs from multiple QA systems based on a mixture model-based MBR model. The key contribution of this work is that, our MBRAR approaches assume little about QA systems and can be easily applied to QA systems with arbitrary sub-components.

The remainder of this paper is organized as follows: Section 2 gives a brief review of the QA task and describes two types of QA systems with different pros and cons. Section 3 presents two MBRAR approaches that can re-rank the answer candidates from single and multiple QA systems respectively. The relationship between our approach and previous work is discussed in Section 4. Section 5 evaluates our methods on large scale questions selected from two domains (Jeopardy! and Web) and shows promising results. Section 6 concludes this paper.

## 2 Question Answering

### 2.1 Overview

Formally, given an input question  $Q$ , a typical factoid QA system generates answers on the basis of the following three procedures:

(1) *Question Understanding*, which determines the answer type and identifies necessary information contained in  $Q$ , such as question focus and lexical answer type (LAT). Such information will be encoded and used by the following procedures.

(2) *Passage Retrieval*, which formulates queries based on  $Q$ , and retrieves passages from offline corpus or online search engines (e.g. Google and Bing).

(3) *Answer Extraction*, which first extracts answer candidates from retrieved passages, and then ranks them based on specific ranking models.



## 2.2 Two Types of QA Systems

We present two different QA systems, which are distinguished from three aspects: answer typing, answer generation, and answer ranking.

The 1<sup>st</sup> QA system is denoted as *Type-Dependent* QA engine (**TD-QA**). In answer typing phase, TD-QA assigns the most possible answer type  $\hat{T}$  to a given question  $Q$  based on:

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(T|Q)$$

$P(T|Q)$  is a probabilistic answer-typing model that is similar to Pinchak and Lin (2006)'s work. In answer generation phase, TD-QA uses a CRF-based Named Entity Recognizer to detect all named entities contained in retrieved passages with the type  $\hat{T}$ , and treat them as the answer candidate space  $\mathcal{H}(Q)$ :

$$\mathcal{H}(Q) = \bigcup_k \mathcal{A}_k$$

In answer ranking phase, the decision rule described below is used to rank answer candidate space  $\mathcal{H}(Q)$ :

$$\begin{aligned} \hat{\mathcal{A}} &= \underset{\mathcal{A} \in \mathcal{H}(Q)}{\operatorname{argmax}} P(\mathcal{A}|\hat{T}, Q) \\ &= \underset{\mathcal{A} \in \mathcal{H}(Q)}{\operatorname{argmax}} \sum_i \lambda_i \cdot h_i(\mathcal{A}, \hat{T}, Q) \end{aligned}$$

where  $\{h_i(\cdot)\}$  is a set of ranking features that measure the correctness of answer candidates, and  $\{\lambda_i\}$  are their corresponding feature weights.

The 2<sup>ed</sup> QA system is denoted as *Type-Independent* QA engine (**TI-QA**). In answer typing phase, TI-QA assigns top  $N$ , instead of the best, answer types  $\mathcal{T}_N(Q)$  for each question  $Q$ . The probability of each type candidate is maintained as well. In answer generation phase, TI-QA extracts all answer candidates from retrieved passages based on answer types in  $\mathcal{T}_N(Q)$ , by the same NER used in TD-QA. In answer ranking phase, TI-QA considers the probabilities of different answer types as well:

$$\begin{aligned} \hat{\mathcal{A}} &= \underset{\mathcal{A} \in \mathcal{H}(Q)}{\operatorname{argmax}} P(\mathcal{A}|Q) \\ &= \underset{\mathcal{A} \in \mathcal{H}(Q)}{\operatorname{argmax}} \sum_{T \in \mathcal{T}_N(Q)} P(\mathcal{A}|T, Q) \cdot P(T|Q) \end{aligned}$$

On one hand, TD-QA can achieve relative high ranking precision, as using a unique answer type greatly reduces the size of the candidate list for

ranking. However, as the answer-typing model is far from perfect, if prediction errors happen, TD-QA can no longer give correct answers at all.

On the other hand, TI-QA can provide higher answer coverage, as it can extract answer candidates with multiple answer types. However, more answer candidates with different types bring more difficulties to the answer ranking model to rank the correct answer to the top 1 position. So the ranking precision of TI-QA is not as good as TD-QA.

## 3 MBR-based Answering Re-ranking

### 3.1 MBRAR for Single QA System

MBR decoding (Bickel and Doksum, 1977) aims to select the hypothesis that minimizes the expected loss in classification. In MBRAR, we replace the loss function with the gain function that measure the correlation between answer candidates. Thus, the objective of the MBRAR approach for single QA system is to find the answer candidate that is most supported by other candidates under QA system's distribution, which can be formally written as:

$$\hat{\mathcal{A}} = \underset{\mathcal{A} \in \mathcal{H}(Q)}{\operatorname{argmax}} \sum_{\mathcal{A}_k \in \mathcal{H}(Q)} \mathcal{G}(\mathcal{A}, \mathcal{A}_k) \cdot P(\mathcal{A}_k|\mathcal{H}(Q))$$

$P(\mathcal{A}_k|\mathcal{H}(Q))$  denotes the *hypothesis distribution* estimated on the search space  $\mathcal{H}(Q)$  based on the following log-linear formulation:

$$P(\mathcal{A}_k|\mathcal{H}(Q)) = \frac{\exp(\beta \cdot P(\mathcal{A}_k|Q))}{\sum_{\mathcal{A}' \in \mathcal{H}} \exp(\beta \cdot P(\mathcal{A}'|Q))}$$

$P(\mathcal{A}_k|Q)$  is the posterior probability of the answer candidate  $\mathcal{A}_k$  based on QA system's ranking model,  $\beta$  is a scaling factor which controls the distribution  $P(\cdot)$  sharp (when  $\beta > 1$ ) or smooth (when  $\beta < 1$ ).

$\mathcal{G}(\mathcal{A}, \mathcal{A}_k)$  is the *gain function* that denotes the degree of how  $\mathcal{A}_k$  supports  $\mathcal{A}$ . This function can be further expanded as a weighted combination of a set of correlation features as:  $\sum_j \lambda_j \cdot h_j(\mathcal{A}, \mathcal{A}_k)$ . The following correlation features are used in  $\mathcal{G}(\cdot)$ :

- answer-level n-gram correlation feature:

$$h_{\text{answer}}(\mathcal{A}, \mathcal{A}_k) = \sum_{\omega \in \mathcal{A}} \#_{\omega}(\mathcal{A}_k)$$

where  $\omega$  denotes an n-gram in  $\mathcal{A}$ ,  $\#_{\omega}(\mathcal{A}_k)$  denotes the number of times that  $\omega$  occurs in  $\mathcal{A}_k$ .

- passage-level n-gram correlation feature:

$$h_{\text{passage}}(\mathcal{A}, \mathcal{A}_k) = \sum_{\omega \in \mathcal{P}_{\mathcal{A}}} \#_{\omega}(\mathcal{P}_{\mathcal{A}_k})$$

where  $\mathcal{P}_{\mathcal{A}}$  denotes passages from which  $\mathcal{A}$  are extracted. This feature measures the degree of  $\mathcal{A}_k$  supports  $\mathcal{A}$  from the context perspective.

- answer-type agreement feature:

$$h_{\text{type}}(\mathcal{A}, \mathcal{A}_k) = \delta(T_{\mathcal{A}}, T_{\mathcal{A}_k})$$

$\delta(T_{\mathcal{A}}, T_{\mathcal{A}_k})$  denotes an indicator function that equals to 1 when the answer types of  $\mathcal{A}$  and  $\mathcal{A}_k$  are the same, and 0 otherwise.

- answer-length feature that is used to penalize long answer candidates.
- averaged passage-length feature that is used to penalize passages with a long averaged length.

### 3.2 MBRAR for Multiple QA Systems

Aiming to apply MBRAR to the outputs from  $N$  QA systems, we modify MBR components as follows.

First, the hypothesis space  $\mathcal{H}_C(Q)$  is built by merging answer candidates of multiple QA systems:

$$\mathcal{H}_C(Q) = \bigcup_i \mathcal{H}_i(Q)$$

Second, the hypothesis distribution is defined as a probability distribution over the combined search space of  $N$  component QA systems and computed as a weighted sum of component model distributions:

$$P(\mathcal{A}|\mathcal{H}_C(Q)) = \sum_{i=1}^N \alpha_i \cdot P(\mathcal{A}|\mathcal{H}_i(Q))$$

where  $\alpha_1, \dots, \alpha_N$  are coefficients with following constraints holds<sup>1</sup>:  $0 \leq \alpha_i \leq 1$  and  $\sum_{i=1}^N \alpha_i = 1$ ,  $P(\mathcal{A}|\mathcal{H}_i(Q))$  is the posterior probability of  $\mathcal{A}$  estimated on the  $i^{\text{th}}$  QA system's search space  $\mathcal{H}_i(Q)$ .

Third, the features used in the gain function  $\mathcal{G}(\cdot)$  can be grouped into two categories, including:

- *system-independent features*, which includes all features described in Section 3.1 for single system based MBRAR method;

- *system-dependent features*, which measure the correctness of answer candidates based on information provided by multiple QA systems:

- system indicator feature  $h_{\text{sys}}(\mathcal{A}, QA_i)$ , which equals to 1 when  $\mathcal{A}$  is generated by the  $i^{\text{th}}$  system  $QA_i$ , and 0 otherwise;
- system ranking feature  $h_{\text{rank}}(\mathcal{A}, QA_i)$ , which equals to the reciprocal of the rank position of  $\mathcal{A}$  predicted by  $QA_i$ . If  $QA_i$  fails to generate  $\mathcal{A}$ , then it equals to 0;
- ensemble feature  $h_{\text{cons}}(\mathcal{A})$ , which equals to 1 when  $\mathcal{A}$  can be generated by all individual QA system, and 0 otherwise.

Thus, the MBRAR for multiple QA systems can be finally formulated as follows:

$$\hat{\mathcal{A}} = \underset{\mathcal{A} \in \mathcal{H}_C(Q)}{\operatorname{argmax}} \sum_{A_i \in \mathcal{H}_C(Q)} \mathcal{G}(\mathcal{A}, A_i) \cdot P(A_i|\mathcal{H}_C(Q))$$

where the training process of the weights in the gain function is carried out with Ranking SVM<sup>2</sup> based on the method described in Verberne et al. (2009).

## 4 Related Work

MBR decoding have been successfully applied to many NLP tasks, e.g. machine translation, parsing, speech recognition and etc. As far as we know, this is the first work that applies MBR principle to QA.

Yaman et al. (2009) proposed a classification based method for QA task that jointly uses multiple 5-W QA systems by selecting one optimal QA system for each question. Comparing to their work, our MBRAR approaches assume few about the question types, and all QA systems contribute in the re-ranking model. Tellez-Valero et al. (2008) presented an answer validation method that helps individual QA systems to automatically detect its own errors based on information from multiple QA systems. Chu-Carroll et al. (2003) presented a multi-level answer resolution algorithm to merge results from the answering agents at the question, passage, and answer levels. Grappy et al.

<sup>1</sup>For simplicity, the coefficients are equally set:  $\alpha_i = 1/N$ .

<sup>2</sup>We use *SVM<sup>Rank</sup>* (Joachims, 2006) that can be found at [www.cs.cornell.edu/people/tj/svm-light/svm\\_rank.html/](http://www.cs.cornell.edu/people/tj/svm-light/svm_rank.html/)

(2012) proposed to use different score combinations to merge answers from different QA systems. Although all methods mentioned above leverage information provided by multiple QA systems, our work is the first time to explore the usage of MBR principle for the QA task.

## 5 Experiments

### 5.1 Data and Metric

Questions from two different domains are used as our evaluation data sets: the first data set includes 10,051 factoid question-answer pairs selected from the Jeopardy! quiz show<sup>3</sup>; while the second data set includes 360 celebrity-asking web questions<sup>4</sup> selected from a commercial search engine, the answers for each question is labeled by human annotators.

The evaluation metric  $Succeed@n$  is defined as the number of questions whose correct answers are successfully ranked to the top  $n$  answer candidates.

### 5.2 MBRAR for Single QA System

We first evaluate the effectiveness of our MBRAR for single QA system. Given the N-best answer outputs from each single QA system, together with their ranking scores assigned by the corresponding ranking components, we further perform MBRAR to re-rank them and show resulting numbers on two evaluation data sets in Table 1 and 2 respectively.

Both Table 1 and Table 2 show that, by leveraging our MBRAR method on individual QA systems, the rankings of correct answers are consistently improved on both Jeopardy! and web questions.

| Jeopardy!    | $Succeed@1$  | $Succeed@2$  | $Succeed@3$  |
|--------------|--------------|--------------|--------------|
| TD-QA        | 2,289        | 2,693        | 2,885        |
| <b>MBRAR</b> | <b>2,372</b> | <b>2,784</b> | <b>2,982</b> |
| TI-QA        | 2,527        | 3,397        | 3,821        |
| <b>MBRAR</b> | <b>2,628</b> | <b>3,500</b> | <b>3,931</b> |

Table 1: Impacts of MBRAR for single QA system on Jeopardy! questions.

We also notice TI-QA performs significantly better than TD-QA on Jeopardy! questions, but worse on web questions. This is due to fact that when the answer type is fixed (PERSON for

<sup>3</sup><http://www.jeopardy.com/>

<sup>4</sup>The answers of such questions are person names.

| Web          | $Succeed@1$ | $Succeed@2$ | $Succeed@3$ |
|--------------|-------------|-------------|-------------|
| TD-QA        | 97          | 128         | 146         |
| <b>MBRAR</b> | <b>99</b>   | <b>130</b>  | <b>148</b>  |
| TI-QA        | 95          | 122         | 136         |
| <b>MBRAR</b> | <b>97</b>   | <b>126</b>  | <b>143</b>  |

Table 2: Impacts of MBRAR for single QA system on web questions.

celebrity-asking questions), TI-QA will generate candidates with wrong answer types, which will definitely deteriorate the ranking accuracy.

### 5.3 MBRAR for Multiple QA Systems

We then evaluate the effectiveness of our MBRAR for multiple QA systems. The mixture model-based MBRAR method described in Section 3.2 is used to rank the combined answer outputs from TD-QA and TI-QA, with ranking results shown in Table 3 and 4.

From Table 3 and Table 4 we can see that, comparing to the ranking performances of single QA systems TD-QA and TI-QA, MBRAR using two QA systems' outputs shows significant improvements on both Jeopardy! and web questions. Furthermore, comparing to MBRAR on single QA system, MBRAR on multiple QA systems can provide extra gains on both questions sets as well.

| Jeopardy!    | $Succeed@1$  | $Succeed@2$  | $Succeed@3$  |
|--------------|--------------|--------------|--------------|
| TD-QA        | 2,289        | 2,693        | 2,885        |
| TI-QA        | 2,527        | 3,397        | 3,821        |
| <b>MBRAR</b> | <b>2,891</b> | <b>3,668</b> | <b>4,033</b> |

Table 3: Impacts of MBRAR for multiple QA systems on Jeopardy! questions.

| Web          | $Succeed@1$ | $Succeed@2$ | $Succeed@3$ |
|--------------|-------------|-------------|-------------|
| TD-QA        | 97          | 128         | 146         |
| TI-QA        | 95          | 122         | 136         |
| <b>MBRAR</b> | <b>108</b>  | <b>137</b>  | <b>152</b>  |

Table 4: Impacts of MBRAR for multiple QA systems on web questions.

## 6 Conclusions and Future Work

In this paper, we present two MBR-based answer re-ranking approaches for QA. Comparing to previous methods, MBRAR provides a systematic way to re-rank answers from either single or multiple QA systems, without considering their heterogeneous implementations of internal components.

Experiments on questions from two different domains show that, our proposed method can significantly improve the ranking performances. In future, we will add more QA systems into our MBRAR framework, and design more features for the MBR gain function.

## References

- P. J. Bickel and K. A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day Inc.
- Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah. 2003. *In Question Answering, Two Heads Are Better Than One*. In proceeding of HLT-NAACL.
- Vaibhava Goel and William Byrne. 2000. *Minimum bayes-risk automatic speech recognition*, Computer Speech and Language.
- Arnaud Grappy, Brigitte Grau, and Sophie Rosset. 2012. *Methods Combination and ML-based Re-ranking of Multiple Hypothesis for Question-Answering Systems*, In proceeding of EACL.
- Thorsten Joachims. 2006. *Training Linear SVMs in Linear Time*, In proceeding of KDD.
- Shankar Kumar and William Byrne. 2004. *Minimum Bayes-Risk Decoding for Statistical Machine Translation*. In proceeding of HLT-NAACL.
- Christopher Pinchak and Dekang Lin. 2006. *A Probabilistic Answer Type Model*. In proceeding of EACL.
- Ivan Titov and James Henderson. 2006. *Bayes Risk Minimization in Natural Language Parsing*. Technical report.
- Alberto Tellez-Valero, Manuel Montes-y-Gomez, Luis Villaseñor-Pineda, and Anselmo Penas. 2008. *Improving Question Answering by Combining Multiple Systems via Answer Validation*. In proceeding of CILing.
- Suzan Verberne, Clst Ru Nijmegen, Hans Van Halteren, Clst Ru Nijmegen, Daphne Theijssen, Ru Nijmegen, Stephan Raaijmakers, Lou Boves, and Clst Ru Nijmegen. 2009. *Learning to rank qa data. evaluating machine learning techniques for ranking answers to why-questions*. In proceeding of SIGIR workshop.
- Sibel Yaman, Dilek Hakkani-Tur, Gokhan Tur, Ralph Grishman, Mary Harper, Kathleen R. McKeown, Adam Meyers, Kartavya Sharma. 2009. *Classification-Based Strategies for Combining Multiple 5-W Question Answering Systems*. In proceeding of INTERSPEECH.

# Question Classification Transfer

Anne-Laure Ligozat

LIMSI-CNRS / BP133, 91403 Orsay cedex, France

ENSIIE / 1, square de la résistance, Evry, France

firstname.lastname@limsi.fr

## Abstract

Question answering systems have been developed for many languages, but most resources were created for English, which can be a problem when developing a system in another language such as French. In particular, for question classification, no labeled question corpus is available for French, so this paper studies the possibility to use existing English corpora and transfer a classification by translating the question and their labels. By translating the training corpus, we obtain results close to a monolingual setting.

## 1 Introduction

In question answering (QA), as in most Natural Language Processing domains, English is the best resourced language, in terms of corpora, lexicons, or systems. Many methods are based on supervised machine learning which is made possible by the great amount of resources for this language.

While developing a question answering system for French, we were thus limited by the lack of resources for this language. Some were created, for example for answer validation (Grappy et al., 2011). Yet, for question classification, although question corpora in French exist, only a small part of them is annotated with question classes, and such an annotation is costly. We thus wondered if it was possible to use existing English corpora, in this case the data used in (Li and Roth, 2002), to create a classification module for French.

Transferring knowledge from one language to another is usually done by exploiting parallel corpora; yet in this case, few such corpora exists (CLEF QA datasets could be used, but question classes are not very precise). We thus investigated the possibility of using machine translation to create a parallel corpus, as has been done for spoken

language understanding (Jabaian et al., 2011) for example. The idea is that using machine translation would enable us to have a large training corpus, either by using the English one and translating the test corpus, or by translating the training corpus. One of the questions posed was whether the quality of present machine translation systems would enable to learn the classification properly.

This paper presents a question classification transfer method, which results are close to those of a monolingual system. The contributions of the paper are the following:

- comparison of train-on-target and test-on-source strategies for question classification;
- creation of an effective question classification system for French, with minimal annotation effort.

This paper is organized as follows: The problem of Question Classification is defined in section 2. The proposed methods are presented in section 3, and the experiments in section 4. Section 5 details the related works in Question Answering. Finally, Section 6 concludes with a summary and a few directions for future work.

## 2 Problem definition

A Question Answering (QA) system aims at returning a precise answer to a natural language question: if asked "How large is the Lincoln Memorial?", a QA system should return the answer "164 acres" as well as a justifying snippet. Most systems include a question classification step which determines the expected answer type, for example *area* in the previous case. This type can then be used to extract the correct answer in documents.

Detecting the answer type is usually considered as a multiclass classification problem, with each answer type representing a class. (Zhang and

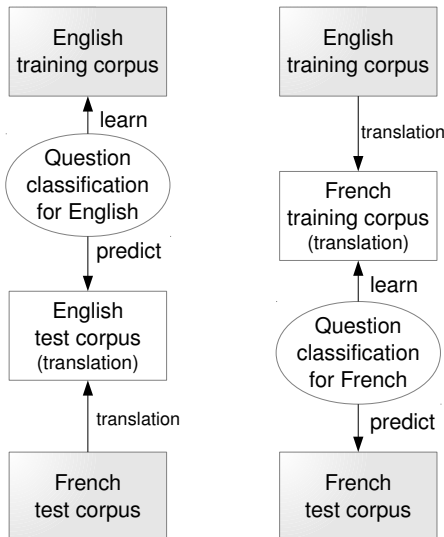


Figure 1: Methods for transferring question classification

Lee, 2003) showed that a training corpus of several thousands of questions was required to obtain around 90% correct classification, which makes it a costly process to adapt a system to another language than English. In this paper, we wish to learn such a system for French, without having to manually annotate thousands of questions.

### 3 Transferring question classification

The two methods tested for transferring the classification, following (Jabaian et al., 2011), are presented in Figure 1:

- The first one (on the left), called *test-on-source*, consists in learning a classification model in English, and to translate the test corpus from French to English, in order to apply the English model on the translated test corpus.
- The second one (on the right), called *train-on-target*, consists in translating the training corpus from English to French. We obtain an labeled French corpus, on which it is possible to learn a classification model.

In the first case, classification is learned on well written questions; yet, as the test corpus is translated, translation errors may disturb the classifier. In the second case, the classification model will be learned on less well written questions, but the corpus may be large enough to compensate for the loss in quality.

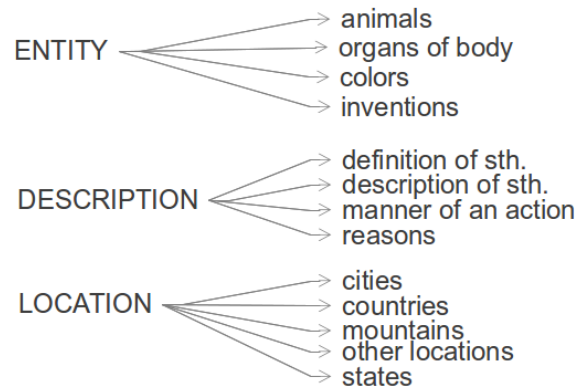


Figure 2: Some of the question categories proposed by (Li and Roth, 2002)

## 4 Experiments

### 4.1 Question classes

We used the question taxonomy proposed by (Li and Roth, 2002), which enabled us to compare our results to those obtained by (Zhang and Lee, 2003) on English. This taxonomy contains two levels: the first one contains 50 fine grained categories, the second one contains 6 coarse grained categories. Figure 2 presents a few of these categories.

### 4.2 Corpora

For English, we used the data from (Li and Roth, 2002), which was assembled from USC, UIUC and TREC collections, and has been manually labeled according to their taxonomy. The training set contains 5,500 labeled questions, and the testing set contains 500 questions.

For French, we gathered questions from several evaluation campaigns: QA@CLEF 2005, 2006, 2007, EQueR and Quæro 2008, 2009 and 2010. After elimination of duplicated questions, we obtained a corpus of 1,421 questions, which were divided into a training set of 728 questions, and a test set of 693 questions<sup>1</sup>. Some of these questions were already labeled, and we manually annotated the rest of them.

Translation was performed by Google Translate online interface, which had satisfactory performance on interrogative forms, which are not well handled by all machine translation systems<sup>2</sup>.

<sup>1</sup>This distribution is due to further constraints on the system.

<sup>2</sup>We tested other translation systems, but Google Translate gave the best results.

| Train         | en   | en                     | fr<br>(trans.)          | fr   |
|---------------|------|------------------------|-------------------------|------|
| Test          | en   | en<br>(trans.)         | fr                      | fr   |
| Method        |      | test-<br>on-<br>source | train-<br>on-<br>target |      |
| 50<br>classes | .798 | .677                   | <b>.794</b>             | .769 |
| 6<br>classes  | .90  | .735                   | <b>.828</b>             | .84  |

Table 1: Question classification precision for both levels of the hierarchy (features = word n-grams, classifier = libsvm)

### 4.3 Classification parameters

The classifier used was LibSVM (Chang and Lin, 2011) with default parameters, which offers one-vs-one multiclass classification, and which (Zhang and Lee, 2003) showed to be most effective for this task.

We only considered surface features, and extracted bag-of-ngrams (with  $n = 1..2$ ).

### 4.4 Results and discussion

Table 1 shows the results obtained with the basic configuration, for both transfer methods.

Results are given in precision, i.e. the proportion of correctly classified questions among the test questions<sup>3</sup>.

Using word n-grams, monolingual English classification obtains .798 correct classification for the fine grained classes, and .90 for the coarse grained classes, results which are very close to those obtained by (Zhang and Lee, 2003).

On French, we obtain lower results: .769 for fine grained classes, and .84 for coarse grained classes, probably mostly due to the smallest size of the training corpus: (Zhang and Lee, 2003) had a precision of .65 for the fine grained classification with a 1,000 questions training corpus.

When translating test questions from French to English, classification precision decreases, as was expected from (Cumbreras et al., 2006). Yet, when translating the training corpus from English to French and learning the classification model

<sup>3</sup>We measured the significance of precision differences (Student t test,  $p=.05$ ), for each level of the hierarchy between each test, and, unless indicated otherwise, comparable results are significantly different in each condition.

| Train         | en   | fr<br>(trans.)          | fr   |
|---------------|------|-------------------------|------|
| Test          | en   | fr                      | fr   |
| Method        |      | train-<br>on-<br>target |      |
| 50<br>classes | .822 | <b>.798</b>             | .807 |
| 6<br>classes  | .92  | <b>.841</b>             | .872 |

Table 2: Question classification precision for both levels of the hierarchy (features = word n-grams with abbreviations, classifier = libsvm)

on this translated corpus, precision is close to the French monolingual one for coarse grained classes and a little higher than monolingual for fine grained classification (and close to the English monolingual one): this method gives precisions of .794 for fine grained classes and .828 for coarse grained classes.

One possible explanation is that the condition when test questions are translated is very sensitive to translation errors: if one of the test questions is not correctly translated, the classifier will have a hard time categorizing it. If the training corpus is translated, translation errors can be counterbalanced by correct translations. In the following results, we do not consider the "en to en (trans)" method since it systematically gives lower results.

As results were lower than our existing rule-based method, we added parts-of-speech as features in order to try to improve them, as well as semantic classes: the classes are lists of words related to a particular category; for example "president" usually means that a person is expected as an answer. Table 2 shows the classification performance with this additional information.

Classification is slightly improved, but only for coarse grained classes (the difference is not significant for fine grained classes).

When analyzing the results, we noted that most confusion errors were due to the type of features given as inputs: for example, to correctly classify the question "What is BPH?" as a question expecting an expression corresponding to an abbreviation (*ABBR:exp class* in the hierarchy), it is necessary to know that "BPH" is an abbreviation. We thus added a specific feature to detect if a question word is an abbreviation, simply by test-

| Train         | en   | fr<br>(trans.) | fr   |
|---------------|------|----------------|------|
| Test          | en   | fr             | fr   |
| 50<br>classes | .804 | <b>.837</b>    | .828 |
| 6<br>classes  | .904 | <b>.869</b>    | .900 |

Table 3: Question classification precision for both levels of the hierarchy (features = word n-grams with abbreviations, classifier = libsvm)

ing if it contains only upper case letters, and normalizing them. Table 3 gives the results with this additional feature (we only kept the method with translation of the training corpus since results were much higher).

Precision is improved for both levels of the hierarchy: for fine grained classes, results increase from .794 to .837, and for coarse grained classes, from .828 to .869. Remaining classification errors are much more disparate.

## 5 Related work

Most question answering systems include question classification, which is generally based on supervised learning. (Li and Roth, 2002) trained the SNoW hierarchical classifier for question classification, with a 50 classes fine grained hierarchy, and a coarse grained one of 6 classes. The features used are words, parts-of-speech, chunks, named entities, chunk heads and words related to a class. They obtain 98.8% correct classification of the coarse grained classes, and 95% on the fine grained one. This hierarchy was widely used by other QA systems.

(Zhang and Lee, 2003) studied the classification performance according to the classifier and training dataset size, as well as the contribution of question parse trees. Their results are 87% correct classification on coarse grained classes and 80% on fine grained classes with vectorial attributes, and 90% correct classification on coarse grained classes and 80% on fine grained classes with structured input and tree kernels.

These question classifications were used for English only. Adapting the methods to other languages requires to annotated large corpora of questions.

In order to classify questions in different languages, (Solorio et al., 2004) proposed an in-

ternet based approach to determine the expected type. By combining this information with question words, they obtain 84% correct classification for English, 84% for Spanish and 89% for Italian, with a cross validation on a 450 question corpus for 7 question classes. One of the limitations raised by the authors is the lack of large labeled corpora for all languages.

A possibility to overcome this lack of resources is to use existing English resources. (Cumbreras et al., 2006) developed a QA system for Spanish, based on an English QA system, by translating the questions from Spanish to English. They obtain a 65% precision for Spanish question classification, while English classification are correctly classified with an 80% precision. This method thus leads to an important drop in performance.

Crosslingual QA systems, in which the question is in a different language than the documents, also usually rely on English systems, and translate answers for example (Bos and Nissim, 2006; Bowden et al., 2008).

## 6 Conclusion

This paper presents a comparison between two transfer modes to adapt question classification from English to French. Results show that translating the training corpus gives better results than translating the test corpus.

Part-of-speech information only was used, but since (Zhang and Lee, 2003) showed that best results are obtained with parse trees and tree kernels, it could be interesting to test this additional information; yet, parsing translated questions may prove unreliable.

Finally, as interrogative forms occur rarely in corpora, their translation is usually of a slightly lower quality. A possible future direction for this work could be to use a specific model of translation for questions in order to learn question classification on higher quality translations.

## References

- J. Bos and M. Nissim. 2006. Cross-lingual question answering by answer translation. In *Working Notes of the Cross Language Evaluation Forum*.
- M. Bowden, M. Olteanu, P. Suriyentrakorn, T. d’Silva, and D. Moldovan. 2008. Multilingual question answering through intermediate translation: Lcc’s poweranswer at qa@clef 2007. *Advances in Mul-*



*tilingual and Multimodal Information Retrieval*, 5152:273–283.

- 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.
- M.Á.G. Cumbreiras, L. López, and F.M. Santiago. 2006. Bruja: Question classification for spanish. using machine translation and an english classifier. In *Proceedings of the Workshop on Multilingual Question Answering*, pages 39–44. Association for Computational Linguistics.
- Arnaud Grappy, Brigitte Grau, Mathieu-Henri Falco, Anne-Laure Ligozat, Isabelle Robba, and Anne Vilnat. 2011. Selecting answers to questions from web documents by a robust validation process. In *IEEE/WIC/ACM International Conference on Web Intelligence*.
- Bassam Jabaian, Laurent Besacier, and Fabrice Lefèvre. 2011. Combination of stochastic understanding and machine translation systems for language portability of dialogue systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5612–5615. IEEE.
- X. Li and D. Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.
- T. Solorio, M. Pérez-Coutino, et al. 2004. A language independent method for question classification. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 1374–1380. Association for Computational Linguistics.
- D. Zhang and W.S. Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–32. ACM.

# Latent Semantic Tensor Indexing for Community-based Question Answering

Xipeng Qiu, Le Tian, Xuanjing Huang

Fudan University, 825 Zhangheng Road, Shanghai, China  
xpqiu@fudan.edu.cn, tianlefd@gmail.com, xjhuang@fudan.edu.cn

## Abstract

Retrieving similar questions is very important in community-based question answering(CQA). In this paper, we propose a unified question retrieval model based on latent semantic indexing with tensor analysis, which can capture word associations among different parts of CQA triples simultaneously. Thus, our method can reduce lexical chasm of question retrieval with the help of the information of question content and answer parts. The experimental result shows that our method outperforms the traditional methods.

## 1 Introduction

Community-based (or collaborative) question answering(CQA) such as Yahoo! Answers<sup>1</sup> and Baidu Zhidao<sup>2</sup> has become a popular online service in recent years. Unlike traditional question answering (QA), information seekers can post their questions on a CQA website which are later answered by other users. However, with the increase of the CQA archive, there accumulate massive duplicate questions on CQA websites. One of the primary reasons is that information seekers cannot retrieve answers they need and thus post another new question consequently. Therefore, it becomes more and more important to find semantically similar questions.

The major challenge for CQA retrieval is the lexical gap (or lexical chasm) among the questions (Jeon et al., 2005b; Xue et al., 2008),

<sup>1</sup><http://answers.yahoo.com/>

<sup>2</sup><http://zhidao.baidu.com/>

|  |
|--|
| <b>Query:</b><br>Q: Why is my laptop screen blinking?                            |
| <b>Expected:</b><br>Q1: How to troubleshoot a flashing screen on an LCD monitor? |
| <b>Not Expected:</b><br>Q2: How to blinking text on screen with PowerPoint?      |

Table 1: An example on question retrieval

as shown in Table 1. Since question-answer pairs are usually short, the word mismatching problem is especially important. However, due to the lexical gap between questions and answers as well as spam typically existing in user-generated content, filtering and ranking answers is very challenging.

The earlier studies mainly focus on generating redundant features, or finding textual clues using machine learning techniques; none of them ever consider questions and their answers as relational data but instead model them as independent information. Moreover, they only consider the answers of the current question, and ignore any previous knowledge that would be helpful to bridge the lexical and semantic gap.

In recent years, many methods have been proposed to solve the word mismatching problem between user questions and the questions in a QA archive(Blooma and Kurian, 2011), among which the translation-based (Riezler et al., 2007; Xue et al., 2008; Zhou et al., 2011) or syntactic-based approaches (Wang et al., 2009) methods have been proven to improve the performance of CQA retrieval.

However, most of these approaches used

pipeline methods: (1) modeling word association; (2) question retrieval combined with other models, such as vector space model (VSM), Okapi model (Robertson et al., 1994) or language model (LM). The pipeline methods often have many non-trivial experimental setting and result to be very hard to reproduce.

In this paper, we propose a novel unified retrieval model for CQA, **latent semantic tensor indexing (LSTI)**, which is an extension of the conventional latent semantic indexing (LSI) (Deerwester et al., 1990). Similar to LSI, LSTI can integrate the two detached parts (modeling word association and question retrieval) into a single model.

In traditional document retrieval, LSI is an effective method to overcome two of the most severe constraints on Boolean keyword queries: synonymy, that is, multiple words with similar meanings, and polysemy, or words with more than one meanings.

Usually in a CQA archive, each entry (or question) is in the following triple form: **(question title, question content, answer)**. Because the performance based solely on the content or the answer part is less than satisfactory, many works proposed that additional relevant information should be provided to help question retrieval (Xue et al., 2008). For example, if a question title contains the keyword “why”, the CQA triple, which contains “because” or “reason” in its answer part, is more likely to be what the user looks for.

Since each triple in CQA has three parts, the natural representation of the CQA collection is a three-dimensional array, or 3rd-order tensor, rather than a matrix. Based on the tensor decomposition, we can model the word association simultaneously in the pairs: question-question, question-body and question-answer.

The rest of the paper is organized as follows: Section 3 introduces the concept of LSI. Section 4 presents our method. Section 5 describes the experimental analysis. Section 6 concludes the paper.

## 2 Related Works

There are some related works on question retrieval in CQA. Various query expansion tech-

niques have been studied to solve word mismatch problems between queries and documents. The early works on question retrieval can be traced back to finding similar questions in Frequently Asked Questions (FAQ) archives, such as the FAQ finder (Burke et al., 1997), which usually used statistical and semantic similarity measures to rank FAQs.

Jeon et al. (2005a; 2005b) compared four different retrieval methods, i.e., the vector space model (Jijkoun and de Rijke, 2005), the Okapi BM25 model (Robertson et al., 1994), the language model, and the translation model, for question retrieval on CQA data, and the experimental results showed that the translation model outperforms the others. However, they focused only on similarity measures between queries (questions) and question titles.

In subsequent work (Xue et al., 2008), a translation-based language model combining the translation model and the language model for question retrieval was proposed. The results showed that translation models help question retrieval since they could effectively address the word mismatch problem of questions. Additionally, they also explored answers in question retrieval.

Duan et al. (2008) proposed a solution that made use of question structures for retrieval by building a structure tree for questions in a category of Yahoo! Answers, which gave more weight to important phrases in question matching.

Wang et al. (2009) employed a parser to build syntactic trees for questions, and questions were ranked based on the similarity between their syntactic trees and that of the query question.

It is worth noting that our method is totally different to the work (Cai et al., 2006) of the same name. They regard documents as matrices, or the second order tensors to generate a low rank approximations of matrices (Ye, 2005). For example, they convert a 1,000,000-dimensional vector of word space into a  $1000 \times 1000$  matrix. However in our model, a document is still represented by a vector. We just project a higher-dimensional vector to a lower-dimensional vector, but not a matrix in Cai’s model. A 3rd-order tensor is

also introduced in our model for better representation for CQA corpus.

### 3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) (Deerwester et al., 1990), also called Latent Semantic Analysis (LSA), is an approach to automatic indexing and information retrieval that attempts to overcome these problems by mapping documents as well as terms to a representation in the so-called latent semantic space.

The key idea of LSI is to map documents (and by symmetry terms) to a low dimensional vector space, the latent semantic space. This mapping is computed by decomposing the term-document matrix  $N$  with SVD,  $N = U\Sigma V^t$ , where  $U$  and  $V$  are orthogonal matrices  $U^tU = V^tV = I$  and the diagonal matrix  $\Sigma$  contains the singular values of  $N$ . The LSA approximation of  $N$  is computed by just keep the largest  $K$  singular values in  $\Sigma$ , which is rank  $K$  optimal in the sense of the  $L^2$ -norm.

LSI has proven to result in more robust word processing in many applications.

## 4 Tensor Analysis for CQA

### 4.1 Tensor Algebra

We first introduce the notation and basic definitions of multilinear algebra. Scalars are denoted by lower case letters ( $a, b, \dots$ ), vectors by bold lower case letters ( $\mathbf{a}, \mathbf{b}, \dots$ ), matrices by bold upper-case letters ( $\mathbf{A}, \mathbf{B}, \dots$ ), and higher-order tensors by calligraphic upper-case letters ( $\mathcal{A}, \mathcal{B}, \dots$ ).

A tensor, also known as  $n$ -way array, is a higher order generalization of a vector (first order tensor) and a matrix (second order tensor). The order of tensor  $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is  $N$ . An element of  $\mathcal{D}$  is denoted as  $d_{i_1, \dots, i_N}$ .

An  $N$ th-order tensor can be flattened into a matrix by  $N$  ways. We denote the matrix  $\mathbf{D}(n)$  as the mode- $n$  flattening of  $\mathcal{D}$  (Kolda, 2002).

Similar with a matrix, an  $N$ th-order tensor can be decomposed through “ $N$ -mode singular value decomposition (SVD)”, which is an extension of SVD that expresses the tensor as the mode- $n$  product of  $N$ -orthogonal spaces.

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N. \quad (1)$$

Tensor  $\mathcal{Z}$ , known as the core tensor, is analogous to the diagonal singular value matrix in conventional matrix SVD.  $\mathcal{Z}$  is in general a full tensor. The core tensor governs the interaction between the mode matrices  $\mathbf{U}_n$ , for  $n = 1, \dots, N$ . Mode matrix  $\mathbf{U}_n$  contains the orthogonal left singular vectors of the mode- $n$  flattened matrix  $\mathbf{D}(n)$ .

The  $N$ -mode SVD algorithm for decomposing  $\mathcal{D}$  is as follows:

1. For  $n = 1, \dots, N$ , compute matrix  $\mathbf{U}_n$  in Eq.(1) by computing the SVD of the flattened matrix  $\mathbf{D}(n)$  and setting  $\mathbf{U}_n$  to be the left matrix of the SVD.
2. Solve for the core tensor as follows  $\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$ .

### 4.2 CQA Tensor

Given a collection of CQA triples,  $\langle q_i, c_i, a_i \rangle$  ( $i = 1, \dots, K$ ), where  $q_i$  is the question and  $c_i$  and  $a_i$  are the content and answer of  $q_i$  respectively. We can use a 3-order tensor  $\mathcal{D} \in \mathbb{R}^{K \times 3 \times T}$  to represent the collection, where  $T$  is the number of terms. The first dimension corresponds to entries, the second dimension, to parts and the third dimension, to the terms.

For example, the flattened matrix of CQA tensor with “terms” direction is composed by three sub-matrices  $\mathbf{M}_{\text{Title}}$ ,  $\mathbf{M}_{\text{Content}}$  and  $\mathbf{M}_{\text{Answer}}$ , as was illustrated in Figure 1. Each sub-matrix is equivalent to the traditional document-term matrix.

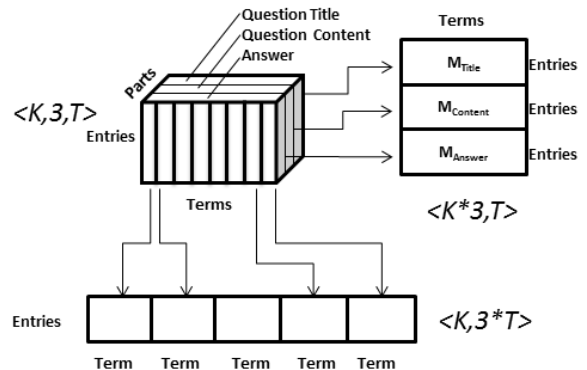


Figure 1: Flattening CQA tensor with “terms” (right matrix) and “entries” (bottom matrix)

Denote  $p_{i,j}$  to be part  $j$  of entry  $i$ . Then we

have the term frequency, defined as follows.

$$\text{tf}_{i,j,k} = \frac{n_{i,j,k}}{\sum_i n_{i,j,k}}, \quad (2)$$

where  $n_{i,j,k}$  is the number of occurrences of the considered term ( $t_k$ ) in  $p_{i,j}$ , and the denominator is the sum of number of occurrences of all terms in  $p_{i,j}$ .

The inverse document frequency is a measure of the general importance of the term.

$$\text{idf}_{j,k} = \log \frac{|K|}{1 + \sum_i I(t_k \in p_{i,j})}, \quad (3)$$

where  $|K|$  is the total number of entries and  $I(\cdot)$  is the indicator function.

Then the element  $d_{i,j,k}$  of tensor  $\mathcal{D}$  is

$$d_{i,j,k} = \text{tf}_{i,j,k} \times \text{idf}_{j,k}. \quad (4)$$

### 4.3 Latent Semantic Tensor Indexing

For the CQA tensor, we can decompose it as illustrated in Figure 2.

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\text{Entry}} \times_2 \mathbf{U}_{\text{Part}} \times_3 \mathbf{U}_{\text{Term}}, \quad (5)$$

where  $\mathbf{U}_{\text{Entry}}$ ,  $\mathbf{U}_{\text{Part}}$  and  $\mathbf{U}_{\text{Term}}$  are left singular matrices of corresponding flattened matrices.  $\mathbf{U}_{\text{Term}}$  spans the term space, and we just use the vectors corresponding to the 1,000 largest singular values in this paper, denoted as  $\mathbf{U}'_{\text{Term}}$ .

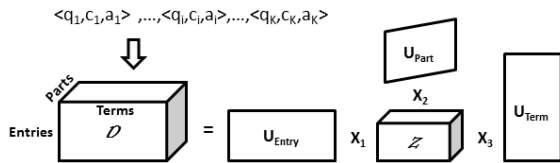


Figure 2: 3-mode SVD of CQA tensor

To deal with such a huge sparse data set, we use singular value decomposition (SVD) implemented in Apache Mahout<sup>3</sup> machine learning library, which is implemented on top of Apache Hadoop<sup>4</sup> using the map/reduce paradigm and scalable to reasonably large data sets.

<sup>3</sup><http://mahout.apache.org/>

<sup>4</sup><http://hadoop.apache.org>

### 4.4 Question Retrieval

In order to retrieve similar question effectively, we project each CQA triple  $\mathcal{D}_q \in \mathbb{R}^{1 \times 3 \times T}$  to the term space by

$$\hat{\mathcal{D}}_i = \mathcal{D}_i \times_3 \mathbf{U}'_{\text{Term}}. \quad (6)$$

Given a new question only with title part, we can represent it by tensor  $\mathcal{D}_q \in \mathbb{R}^{1 \times 3 \times T}$ , and its  $\mathbf{M}_{\text{Content}}$  and  $\mathbf{M}_{\text{Answer}}$  are zero matrices. Then we project  $\mathcal{D}_q$  to the term space and get  $\hat{\mathcal{D}}_q$ .

Here,  $\hat{\mathcal{D}}_q$  and  $\hat{\mathcal{D}}_i$  are degraded tensors and can be regarded as matrices. Thus, we can calculate the similarity between  $\hat{\mathcal{D}}_q$  and  $\hat{\mathcal{D}}_i$  with normalized Frobenius inner product.

For two matrices  $A$  and  $B$ , the Frobenius inner product, indicated as  $A : B$ , is the component-wise inner product of two matrices as though they are vectors.

$$A : B = \sum_{i,j} A_{i,j} B_{i,j} \quad (7)$$

To reduce the affect of length, we use the normalized Frobenius inner product.

$$\overline{A : B} = \frac{A : B}{\sqrt{A : A} \times \sqrt{B : B}} \quad (8)$$

While given a new question both with title and content parts,  $\mathbf{M}_{\text{Content}}$  is not a zero matrix and could be also employed in the question retrieval process. A simple strategy is to sum up the scores of two parts.

## 5 Experiments

### 5.1 Datasets

We collected the resolved CQA triples from the ‘‘computer’’ category of Yahoo! Answers and Baidu Zhidao websites. We just selected the resolved questions that already have been given their best answers. The CQA triples are preprocessed with stopwords removal (Chinese sentences are segmented into words in advance by FudanNLP toolkit(Qiu et al., 2013)).

In order to evaluate our retrieval system, we divide our dataset into two parts. The first part is used as training dataset; the rest is used as test dataset for evaluation. The datasets are shown in Table 2.

| DataSet        | training data size | test data size |
|----------------|--------------------|----------------|
| Baidu Zhidao   | 423k               | 1000           |
| Yahoo! Answers | 300k               | 1000           |

Table 2: Statistics of Collected Datasets

| Methods              | MAP   |
|----------------------|-------|
| Okapi                | 0.359 |
| LSI                  | 0.387 |
| (Jeon et al., 2005b) | 0.372 |
| (Xue et al., 2008)   | 0.381 |
| LSTI                 | 0.415 |

Table 3: Retrieval Performance on Dataset from Yahoo! Answers

## 5.2 Evaluation

We compare our method with two baseline methods: Okapi BM25 and LSI and two state-of-the-art methods: (Jeon et al., 2005b)(Xue et al., 2008). In LSI, we regard each triple as a single document. Three annotators are involved in the evaluation process. Given a returned result, two annotators are asked to label it with “relevant” or “irrelevant”. If an annotator considers the returned result semantically equivalent to the queried question, he labels it as “relevant”; otherwise, it is labeled as “irrelevant”. If a conflict happens, the third annotator will make the final judgement.

We use **mean average precision** (MAP) to evaluate the effectiveness of each method.

The experiment results are illustrated in Table 3 and 4, which show that our method outperforms the others on both datasets.

The primary reason is that we incorporate the content of the question body and the answer parts into the process of question retrieval, which should provide additional relevance information. Different to

| Methods              | MAP   |
|----------------------|-------|
| Okapi                | 0.423 |
| LSI                  | 0.490 |
| (Jeon et al., 2005b) | 0.498 |
| (Xue et al., 2008)   | 0.512 |
| LSTI                 | 0.523 |

Table 4: Retrieval Performance on Dataset from Baidu Zhidao

the translation-based methods, our method can capture the mapping relations in three parts (question, content and answer) simultaneously.

It is worth noting that the problem of data sparsity is more crucial for LSTI since the size of a tensor in LSTI is larger than a term-document matrix in LSI. When the size of data is small, LSTI tends to just align the common words and thus cannot find the corresponding relations among the focus words in CQA triples. Therefore, more CQA triples may result in better performance for our method.

## 6 Conclusion

In this paper, we proposed a novel retrieval approach for community-based QA, called LSTI, which analyzes the CQA triples with naturally tensor representation. LSTI is a unified model and effectively resolves the problem of lexical chasm for question retrieval. For future research, we will extend LSTI to a probabilistic form (Hofmann, 1999) for better scalability and investigate its performance with a larger corpus.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was funded by NSFC (No.61003091 and No.61073069) and 973 Program (No.2010CB327900).

## References

- M.J. Blooma and J.C. Kurian. 2011. Research issues in community based question answering. In *PACIS 2011 Proceedings*.
- R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–66.
- Deng Cai, Xiaofei He, and Jiawei Han. 2006. Tensor space model for document analysis. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.
- S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

- Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *Proceedings of ACL-08: HLT*, pages 156–164, Columbus, Ohio, June. Association for Computational Linguistics.
- 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 Press New York, NY, USA.
- J. Jeon, W.B. Croft, and J.H. Lee. 2005a. Finding semantically similar questions based on their answers. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 617–618. ACM.
- J. Jeon, W.B. Croft, and J.H. Lee. 2005b. Finding similar questions in large question and answer archives. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 84–90.
- V. Jijkoun and M. de Rijke. 2005. Retrieving answers from frequently asked questions pages on the web. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83.
- T.G. Kolda. 2002. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255.
- Xipeng Qiu, Qi Zhang, and Xuanjing Huang. 2013. Fudannlp: A toolkit for chinese natural language processing. In *Proceedings of ACL*.
- S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at trec-3. In *TREC*, pages 109–126.
- K. Wang, Z. Ming, and T.S. Chua. 2009. A syntactic tree matching approach to finding similar questions in community-based QA services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 187–194. ACM.
- X. Xue, J. Jeon, and W.B. Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 475–482. ACM.
- J.M. Ye. 2005. Generalized low rank approximations of matrices. *Mach. Learn.*, 61(1):167–191.
- G. Zhou, L. Cai, J. Zhao, and K. Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 653–662. Association for Computational Linguistics.

# Measuring semantic content in distributional vectors

**Aurélie Herbelot**

EB Kognitionswissenschaft

Universität Potsdam

Golm, Germany

aurelie.herbelot@cantab.net

**Mohan Ganesalingam**

Trinity College

University of Cambridge

Cambridge, UK

mohan0@gmail.com

## Abstract

Some words are more contentful than others: for instance, *make* is intuitively more general than *produce* and *fifteen* is more ‘precise’ than *a group*. In this paper, we propose to measure the ‘semantic content’ of lexical items, as modelled by distributional representations. We investigate the hypothesis that semantic content can be computed using the Kullback-Leibler (KL) divergence, an information-theoretic measure of the relative entropy of two distributions. In a task focusing on retrieving the correct ordering of hyponym-hypernym pairs, the KL divergence achieves close to 80% precision but does not outperform a simpler (linguistically unmotivated) frequency measure. We suggest that this result illustrates the rather ‘intensional’ aspect of distributions.

## 1 Introduction

Distributional semantics is a representation of lexical meaning that relies on a statistical analysis of the way words are used in corpora (Curran, 2003; Turney and Pantel, 2010; Erk, 2012). In this framework, the semantics of a lexical item is accounted for by modelling its co-occurrence with other words (or any larger lexical context). The representation of a target word is thus a vector in a space where each dimension corresponds to a possible context. The weights of the vector components can take various forms, ranging from simple co-occurrence frequencies to functions such as Pointwise Mutual Information (for an overview, see (Evert, 2004)).

This paper investigates the issue of computing the semantic content of distributional vectors.

That is, we look at the ways we can distributionally express that *make* is a more general verb than *produce*, which is itself more general than, for instance, *weave*. Although the task is related to the identification of hyponymy relations, it aims to reflect a more encompassing phenomenon: we wish to be able to compare the semantic content of words within parts-of-speech where the standard notion of hyponymy does not apply (e.g. prepositions: see *with* vs. *next to* or *of* vs. *concerning*) and across parts-of-speech (e.g. *fifteen* vs. *group*).

The hypothesis we will put forward is that semantic content is related to notions of relative entropy found in information theory. More specifically, we hypothesise that the more specific a word is, the more the distribution of the words co-occurring with it will differ from the baseline distribution of those words in the language as a whole. (A more intuitive way to phrase this is that the more specific a word is, the more information it gives us about which other words are likely to occur near it.) The specific measure of difference that we will use is the Kullback-Leibler divergence of the distribution of words co-occurring with the target word against the distribution of those words in the language as a whole. We evaluate our hypothesis against a subset of the WordNet hierarchy (given by (Baroni et al, 2012)), relying on the intuition that in a hyponym-hypernym pair, the hyponym should have higher semantic content than its hypernym.

The paper is structured as follows. We first define our notion of semantic content and motivate the need for measuring semantic content in distributional setups. We then describe the implementation of the distributional system we use in this paper, emphasising our choice of weighting measure. We show that, using the compo-



nents of the described weighting measure, which are both probability distributions, we can calculate the relative entropy of a distribution by inserting those probability distributions in the equation for the Kullback-Leibler (KL) divergence. We finally evaluate the KL measure against a basic notion of frequency and conclude with some error analysis.

## 2 Semantic content

As a first approximation, we will define semantic content as informativeness with respect to denotation. Following Searle (1969), we will take a ‘successful reference’ to be a speech act where the choice of words used by the speaker appropriately identifies a referent for the hearer. Glossing over questions of pragmatics, we will assume that a more informative word is more likely to lead to a successful reference than a less informative one. That is, if Kim owns a cat and a dog, the identifying expression *my cat* is a better referent than *my pet* and so *cat* can be said to have more semantic content than *pet*.

While our definition relies on reference, it also posits a correspondence between actual utterances and denotation. Given two possible identifying expressions  $e_1$  and  $e_2$ ,  $e_1$  may be preferred in a particular context, and so, context will be an indicator of the amount of semantic content in an expression. In Section 5, we will produce an explicit hypothesis for how the amount of semantic content in a lexical item affects the contexts in which it appears.

A case where semantic content has a direct correspondence with a lexical relation is hyponymy. Here, the correspondence relies entirely on a basic notion of extension. For instance, it is clear that *hammer* is more contentful than *tool* because the extension of *hammer* is smaller than that of *tool*, and therefore more discriminating in a given identifying expression (See *Give me the hammer* versus *Give me the tool*). But we can also talk about semantic content in cases where the notion of extension does not necessarily apply. For example, it is not usual to talk of the extension of a preposition. However, in context, the use of a preposition against another one might be more discriminating in terms of reference. Compare a) *Sandy is with Kim* and b) *Sandy is next to Kim*. Given a set of possible situations involving, say, Kim and Sandy at a party, we could show that b) is more discriminating than a), because it excludes the sit-

uations where Sandy came to the party with Kim but is currently talking to Kay at the other end of the room. The fact that *next to* expresses physical proximity, as opposed to just being in the same situation, confers it more semantic content according to our definition. Further still, there may be a need for comparing the informativeness of words across parts of speech (compare *A group of/Fifteen people was/were waiting in front of the town hall*).

Although we will not discuss this in detail, there is a notion of semantic content above the word level which should naturally derive from composition rules. For instance, we would expect the composition of a given intersective adjective and a given noun to result into a phrase with a semantic content greater than that of its components (or at least equal to it).

## 3 Motivation

The last few years have seen a growing interest in distributional semantics as a representation of lexical meaning. Owing to their mathematical interpretation, distributions allow linguists to simulate human similarity judgements (Lund, Burgess and Atchley, 1995), and also reproduce some of the features given by test subjects when asked to write down the characteristics of a given concept (Baroni and Lenci, 2008). In a distributional semantic space, for instance, the word ‘cat’ may be close to ‘dog’ or to ‘tiger’, and its vector might have high values along the dimensions ‘meow’, ‘mouse’ and ‘pet’. Distributional semantics has had great successes in recent years, and for many computational linguists, it is an essential tool for modelling phenomena affected by lexical meaning.

If distributional semantics is to be seen as a general-purpose representation, however, we should evaluate it across all properties which we deem relevant to a model of the lexicon. We consider semantic content to be one such property. It underlies the notion of hyponymy and naturally models our intuitions about the ‘precision’ (as opposed to ‘vagueness’) of words.

Further, semantic content may be crucial in solving some fundamental problems of distributional semantics. As pointed out by McNally (2013), there is no easy way to define the notion of a function word and this has consequences for theories where function words are *not* assigned a distributional representation. McNally suggests that the most appropriate way to separate function

from content words might, in the end, involve taking into account how much ‘descriptive’ content they have.

#### 4 An implementation of a distributional system

The distributional system we implemented for this paper is close to the system of Mitchell and Lapata (2010) (subsequently M&L). As background data, we use the British National Corpus (BNC) in lemmatised format. Each lemma is followed by a part of speech according to the CLAWS tagset format (Leech, Garside, and Bryant, 1994). For our experiments, we only keep the first letter of each part-of-speech tag, thus obtaining broad categories such as N or V. Furthermore, we only retain words in the following categories: nouns, verbs, adjectives and adverbs (punctuation is ignored). Each article in the corpus is converted into a 11-word window format, that is, we are assuming that context in our system is defined by the five words preceding and the five words following the target.

To calculate co-occurrences, we use the following equations:

$$freq_{c_i} = \sum_t freq_{c_i,t} \quad (1)$$

$$freq_t = \sum_{c_i} freq_{c_i,t} \quad (2)$$

$$freq_{total} = \sum_{c_i,t} freq_{c_i,t} \quad (3)$$

The quantities in these equations represent the following:

|                |  |
|----------------|--|
| $freq_{c_i,t}$ | frequency of the context word $c_i$ with the target word $t$ |
| $freq_{total}$ | total count of word tokens                                   |
| $freq_t$       | frequency of the target word $t$                             |
| $freq_{c_i}$   | frequency of the context word $c_i$                          |

As in M&L, we use the 2000 most frequent words in our corpus as the semantic space dimensions. M&L calculate the weight of each context term in the distribution as follows:

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)} = \frac{freq_{c_i,t} \times freq_{total}}{freq_t \times freq_{c_i}} \quad (4)$$

We will not directly use the measure  $v_i(t)$  as it is not a probability distribution and so is not suitable for entropic analysis; instead our analysis will

be phrased in terms of the probability distributions  $p(c_i|t)$  and  $p(c_i)$  (the numerator and denominator in  $v_i(t)$ ).

#### 5 Semantic content as entropy: two measures

Resnik (1995) uses the notion of information content to improve on the standard edge counting methods proposed to measure similarity in taxonomies such as WordNet. He proposes that the information content of a term  $t$  is given by the self-information measure  $-\log p(t)$ . The idea behind this measure is that, as the frequency of the term increases, its informativeness decreases. Although a good first approximation, the measure cannot be said to truly reflect our concept of semantic content. For instance, in the British National Corpus, *time* and *see* are more frequent than *thing* or *may* and *man* is more frequent than *part*. However, it seems intuitively right to say that *time*, *see* and *man* are more ‘precise’ concepts than *thing*, *may* and *part* respectively. Or said otherwise, there is no indication that more general concepts occur in speech more than less general ones. We will therefore consider self-information as a baseline.

As we expect more specific words to be more informative about which words co-occur with them, it is natural to try to measure the specificity of a word by using notions from information theory to analyse the probability distribution  $p(c_i|t)$  associated with the word. The standard notion of entropy is not appropriate for this purpose, because it does not take account of the fact that the words serving as semantic space dimensions may have different frequencies in language as a whole, i.e. of the fact that  $p(c_i)$  does not have a uniform distribution. Instead we need to measure the degree to which  $p(c_i|t)$  differs from the context word distribution  $p(c_i)$ . An appropriate measure for this is the Kullback-Leibler (KL) divergence or relative entropy:

$$D_{KL}(P||Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right)P(i) \quad (5)$$

By taking  $P(i)$  to be  $p(c_i|t)$  and  $Q(i)$  to be  $p(c_i)$  (as given by Equation 4), we calculate the relative entropy of  $p(c_i|t)$  and  $p(c_i)$ . The measure is clearly informative: it reflects the way that  $t$  modifies the expectation of seeing  $c_i$  in the corpus. We hypothesise that when compared to the distribution  $p(c_i)$ , more informative words will have a

more ‘distorted’ distribution  $p(c_i|t)$  and that the KL divergence will reflect this.<sup>1</sup>

## 6 Evaluation

In Section 2, we defined semantic content as a notion encompassing various referential properties, including a basic concept of extension in cases where it is applicable. However, we do not know of a dataset providing human judgements over the general informativeness of lexical items. So in order to evaluate our proposed measure, we investigate its ability to retrieve the right ordering of hyponym pairs, which can be considered a subset of the issue at hand.

Our assumption is that if  $X$  is a hypernym of  $Y$ , then the information content in  $X$  will be lower than in  $Y$  (because it has a more ‘general’ meaning). So, given a pair of words  $\{w_1, w_2\}$  in a known hyponymy relation, we should be able to tell which of  $w_1$  or  $w_2$  is the hypernym by computing the respective KL divergences.

We use the hypernym data provided by (Baroni et al, 2012) as testbed for our experiment.<sup>2</sup> This set of hyponym-hypernym pairs contains 1385 instances retrieved from the WordNet hierarchy. Before running our system on the data, we make slight modifications to it. First, as our distributions are created over the British National Corpus, some spellings must be converted to British English: for instance, *color* is replaced by *colour*. Second, five of the nouns included in the test set are not in the BNC. Those nouns are *brethren*, *intranet*, *iPod*, *webcam* and *IX*. We remove the pairs containing those words from the data. Third, numbers such as *eleven* or *sixty* are present in the Baroni et al set as nouns, but not in the BNC. Pairs containing seven such numbers are therefore also removed from the data. Finally, we encounter tagging issues with three words, which we match to their BNC equivalents: *acoustics* and *annals* are matched to *acoustic* and *annal*, and *trouser* to *trousers*. These modifications result in a test set of 1279 remaining pairs.

We then calculate both the self-information measure and the KL divergence of all terms in-

<sup>1</sup>Note that KL divergence is not symmetric:  $D_{KL}(p(c_i|t)||p(c_i))$  is not necessarily equal to  $D_{KL}(p(c_i)||p(c_i|t))$ . The latter is inferior as a few very small values of  $p(c_i|t)$  can have an inappropriately large effect on it.

<sup>2</sup>The data is available at <http://clic.cimec.unitn.it/Files/PublicData/eac12012-data.zip>.

cluded in our test set. In order to evaluate the system, we record whether the calculated entropies match the order of each hypernym-hyponym pair. That is, we count a pair as correctly represented by our system if  $w_1$  is a hypernym of  $w_2$  and  $KL(w_1) < KL(w_2)$  (or, in the case of the baseline,  $SI(w_1) < SI(w_2)$  where  $SI$  is self-information).

Self-information obtains 80.8% precision on the task, with the KL divergence lagging a little behind with 79.4% precision (the difference is not significant). In other terms, both measures perform comparably. We analyse potential reasons for this disappointing result in the next section.

## 7 Error analysis

It is worth reminding ourselves of the assumption we made with regard to semantic content. Our hypothesis was that with a ‘more general’ target word  $t$ , the  $p(c_i|t)$  distribution would be fairly similar to  $p(c_i)$ .

Manually checking some of the pairs which were wrongly classified by the KL divergence reveals that our hypothesis might not hold. For example, the pair *beer* – *beverage* is classified incorrectly. When looking at the *beverage* distribution, it is clear that it does not conform to our expectations: it shows high  $v_i(t)$  weights along the *food*, *wine*, *coffee* and *tea* dimensions, for instance, i.e. there is a large difference between  $p(c_{food})$  and  $p(c_{food}|t)$ , etc. Although *beverage* is an umbrella word for many various types of drinks, speakers of English use it in very particular contexts. So, distributionally, it is *not* a ‘general word’. Similar observations can be made for, e.g. *liquid* (strongly associated with *gas*, presumably via coordination), *anniversary* (linked to the verb *mark* or the noun *silver*), or again *projectile* (co-occurring with *weapon*, *motion* and *speed*).

The general point is that, as pointed out elsewhere in the literature (Erk, 2013), distributions are a good representation of (some aspects of) intension, but they are less apt to model extension.<sup>3</sup> So a term with a large extension like *beverage* may have a more restricted (distributional) intension than a word with a smaller extension, such as

<sup>3</sup>We qualify ‘intension’ here, because in the sense of a mapping from possible worlds to extensions, intension cannot be said to be provided by distributions: the distribution of *beverage*, it seems, does not allow us to successfully pick out all beverages in the real world.

beer.<sup>4</sup>

Contributing to this issue, fixed phrases, named entities and generally strong collocations skew our distributions. So for instance, in the *jewelry* distribution, the most highly weighted context is *mental* (with  $v_i(t) = 395.3$ ) because of the music album *Mental Jewelry*. While named entities could easily be eliminated from the system's results by pre-processing the corpus with a named entity recogniser, the issue is not so simple when it comes to fixed phrases of a more compositional nature (e.g. *army ant*): excluding them might be detrimental for the representation (it is, after all, part of the meaning of *ant* that it can be used metaphorically to refer to people) and identifying such phrases is a non-trivial problem in itself.

Some of the errors we observe may also be related to word senses. For instance, the word *medium*, to be found in the pair *magazine – medium*, can be synonymous with *middle*, *clairvoyant* or again *mode of communication*. In the sense of *clairvoyant*, it is clearly more specific than in the sense intended in the test pair. As distributions do not distinguish between senses, this will have an effect on our results.

## 8 Conclusion

In this paper, we attempted to define a measure of distributional semantic content in order to model the fact that some words have a more general meaning than others. We compared the Kullback-Leibler divergence to a simple self-information measure. Our experiments, which involved retrieving the correct ordering of hyponym-hypernym pairs, had disappointing results: the KL divergence was unable to outperform self-information, and both measures misclassified around 20% of our testset.

Our error analysis showed that several factors contributed to the misclassifications. First, distributions are unable to model extensional properties which, in many cases, account for the feeling that a word is more general than another. Second, strong collocation effects can influence the measurement of information negatively: it is an open question which phrases should be considered 'words-with-spaces' when building distributions. Finally, dis-

<sup>4</sup>Although it is more difficult to talk of the extension of e.g. adverbials (*very*) or some adjectives (*skillful*), the general point is that text is biased towards a certain usage of words, while the general meaning a competent speaker ascribes to lexical items does not necessarily follow this bias.

tributional representations do not distinguish between word senses, which in many cases is a desirable feature, but interferes with the task we suggested in this work.

To conclude, we would like to stress that we do not think another information-theoretic measure would perform hugely better than the KL divergence. The point is that the nature of distributional vectors makes them sensitive to word usage and that, despite the general assumption behind distributional semantics, word usage might not suffice to model all aspects of lexical semantics. We leave as an open problem the issue of whether a modified form of our 'basic' distributional vectors would encode the right information.

## Acknowledgements

This work was funded by a postdoctoral fellowship from the Alexander von Humboldt Foundation to the first author, and a Title A Fellowship from Trinity College, Cambridge, to the second author.

## References

- Baroni, Marco, and Lenci, Alessandro. 2008. Concepts and properties in word spaces. In Alessandro Lenci (ed.), *From context to meaning: Distributional models of the lexicon in linguistics and cognitive science* (Special issue of the Italian Journal of Linguistics 20(1)), pages 55–88.
- Baroni, Marco, Raffaella Bernardi, Ngoc-Quynh Do and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL2012)*, pages 23–32.
- Baroni, Marco, Raffaella Bernardi, and Roberto Zamparelli. 2012. Frege in Space: a Program for Compositional Distributional Semantics. Under review.
- Curran, James. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh, Scotland, UK.
- Erk, Katrin. 2012. Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass*, 6:10:635–653.
- Erk, Katrin. 2013. Towards a semantics for distributional representations. In *Proceedings of the Tenth International Conference on Computational Semantics (IWCS2013)*.
- Evert, Stefan. 2004. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, University of Stuttgart.

- Leech, Geoffrey, Roger Garside, and Michael Bryant. 1994. Claws4: The tagging of the british national corpus. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, pages 622–628, Kyoto, Japan.
- Lund, Kevin, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th annual conference of the Cognitive Science Society*, Vol. 17, pages 660–665.
- McNally, Louise. 2013. Formal and distributional semantics: From romance to relationship. In *Proceedings of the 'Towards a Formal Distributional Semantics' workshop*, 10th International Conference on Computational Semantics (IWCS2013), Potsdam, Germany. Invited talk.
- Mitchell, Jeff and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, November.
- Resnik, Philipp. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 448–453.
- Searle, John R. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge University Press.
- Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

# Modeling Human Inference Process for Textual Entailment Recognition

Hen-Hsen Huang

Kai-Chun Chang

Hsin-Hsi Chen

Department of Computer Science and Information Engineering

National Taiwan University

No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan

{hhhuang, kcchang}@nlg.csie.ntu.edu.tw; hhchen@ntu.edu.tw

## Abstract

This paper aims at understanding what human think in textual entailment (*TE*) recognition process and modeling their thinking process to deal with this problem. We first analyze a labeled RTE-5 test set and find that the negative entailment phenomena are very effective features for *TE* recognition. Then, a method is proposed to extract this kind of phenomena from text-hypothesis pairs automatically. We evaluate the performance of using the negative entailment phenomena on both the English RTE-5 dataset and Chinese NTCIR-9 RITE dataset, and conclude the same findings.

## 1 Introduction

Textual Entailment (*TE*) is a directional relationship between pairs of text expressions, text (*T*) and hypothesis (*H*). If human would agree that the meaning of *H* can be inferred from the meaning of *T*, we say that *T* entails *H* (Dagan et al., 2006). The researches on textual entailment have attracted much attention in recent years due to its potential applications (Androutsopoulos and Malakasiotis, 2010). Recognizing Textual Entailment (*RTE*) (Bentivogli, et al., 2011), a series of evaluations on the developments of English *TE* recognition technologies, have been held seven times up to 2011. In the meanwhile, *TE* recognition technologies in other languages are also underway (Shima, et al., 2011).

Sammons, et al., (2010) propose an evaluation metric to examine the characteristics of a *TE* recognition system. They annotate text-hypothesis pairs selected from the RTE-5 test set with a series of linguistic phenomena required in the human inference process. The *RTE* systems are evaluated by the new indicators, such as how many *T-H* pairs annotated with a particular phe-

nomenon can be correctly recognized. The indicators can tell developers which systems are better to deal with *T-H* pairs with the appearance of which phenomenon. That would give developers a direction to enhance their *RTE* systems.

Such linguistic phenomena are thought as important in the human inference process by annotators. In this paper, we use this valuable resource from a different aspect. We aim at knowing the ultimate performance of *TE* recognition systems which embody human knowledge in the inference process. The experiments show five negative entailment phenomena are strong features for *TE* recognition, and this finding confirms the previous study of Vanderwende et al. (2006). We propose a method to acquire the linguistic phenomena automatically and use them in *TE* recognition.

This paper is organized as follows. In Section 2, we introduce linguistic phenomena used by annotators in the inference process and point out five significant negative entailment phenomena. Section 3 proposes a method to extract them from *T-H* pairs automatically, and discuss their effects on *TE* recognition. In Section 4, we extend the methodology to the BC (binary class subtask) dataset distributed by NTCIR-9 RITE task (Shima, et al., 2011) and discuss their effects on *TE* recognition in Chinese. Section 5 concludes the remarks.

## 2 Human Inference Process in TE

We regard the human annotated phenomena as features in recognizing the binary entailment relation between the given *T-H* pairs, i.e., ENTAILMENT and NO ENTAILMENT. Total 210 *T-H* pairs are chosen from the RTE-5 test set by Sammons et al. (2010), and total 39 linguistic phenomena divided into the 5 aspects, including knowledge domains, hypothesis structures, inference phenomena, negative entailment phenome-

na, and knowledge resources, are annotated on the selected dataset.

## 2.1 Five aspects as features

We train SVM classifiers to evaluate the performances of the five aspects of phenomena as features for *TE* recognition. LIBSVM RBF kernel (Chang and Lin, 2011) is adopted to develop classifiers with the parameters tuned by grid search. The experiments are done with 10-fold cross validation.

For the dataset of Sammons et al. (2010), two annotators are involved in labeling the above 39 linguistic phenomena on the *T-H* pairs. They may agree or disagree in the annotation. In the experiments, we consider the effects of their agreement. Table 1 shows the results. Five aspects are first regarded as individual features, and are then merged together. Schemes “Annotator A” and “Annotator B” mean the phenomena labelled by annotator A and annotator B are used as features respectively. The “A AND B” scheme, a strict criterion, denotes a phenomenon exists in a *T-H* pair only if both annotators agree with its appearance. In contrast, the “A OR B” scheme, a looser criterion, denotes a phenomenon exists in a *T-H* pair if at least one annotator marks its appearance.

We can see that the aspect of *negative entailment phenomena* is the most significant feature among the five aspects. With only 9 phenomena in this aspect, the SVM classifier achieves accuracy above 90% no matter which labeling schemes are adopted. Comparatively, the best accuracy in RTE-5 task is 73.5% (Iftene and Moruz, 2009). In negative entailment phenomena aspect, the “A OR B” scheme achieves the best accuracy. In the following experiments, we adopt this labeling scheme.

## 2.2 Negative entailment phenomena

There is a large gap between using negative entailment phenomena and using the second effective features (i.e., inference phenomena). Moreover, using the negative entailment phenomena as features only is even better than using all the 39 linguistic phenomena. We further analyze which negative entailment phenomena are more significant.

There are nine linguistic phenomena in the aspect of negative entailment. We take each phenomenon as a single feature to do the two-way textual entailment recognition. The “A OR B” scheme is applied. Table 2 shows the experimental results.

|                               | Annotator A | Annotator B | A AND B | A OR B |
|-------------------------------|-------------|-------------|---------|--------|
| Knowledge Domains             | 50.95%      | 52.38%      | 52.38%  | 50.95% |
| Hypothesis Structures         | 50.95%      | 51.90%      | 50.95%  | 51.90% |
| Inference Phenomena           | 74.29%      | 72.38%      | 72.86%  | 74.76% |
| Negative Entailment Phenomena | 97.14%      | 95.71%      | 92.38%  | 97.62% |
| Knowledge Resources           | 69.05%      | 69.52%      | 67.62%  | 69.52% |
| ALL                           | 97.14%      | 92.20%      | 90.48%  | 97.14% |

Table 1: Accuracy of recognizing binary *TE* relation with the five aspects as features.

| Phenomenon ID | Negative entailment Phenomenon | Accuracy |
|---------------|--------------------------------|----------|
| 0             | Named Entity mismatch          | 60.95%   |
| 1             | Numeric Quantity mismatch      | 54.76%   |
| 2             | Disconnected argument          | 55.24%   |
| 3             | Disconnected relation          | 57.62%   |
| 4             | Exclusive argument             | 61.90%   |
| 5             | Exclusive relation             | 56.67%   |
| 6             | Missing modifier               | 56.19%   |
| 7             | Missing argument               | 69.52%   |
| 8             | Missing relation               | 68.57%   |

Table 2: Accuracy of recognizing *TE* relation with individual negative entailment phenomena.

The 1<sup>st</sup> column is phenomenon ID, the 2<sup>nd</sup> column is the phenomenon, and the 3<sup>rd</sup> column is the accuracy of using the phenomenon in the binary classification. Comparing with the best accuracy 97.62% shown in Table 1, the highest accuracy in Table 2 is 69.52%, when missing argument is adopted. It shows that each phenomenon is suitable for some *T-H* pairs, and merging all negative entailment phenomena together achieves the best performance.

We consider all possible combinations of these 9 negative entailment phenomena, i.e.,  $C_1^9 + \dots + C_9^9 = 511$  feature settings, and use each feature setting to do 2-way entailment relation recognition by LIBSVM. The notation  $C_n^m$  denotes a set of  $\frac{m!}{(m-n)!n!}$  feature settings, each with  $n$  features.

The model using all nine phenomena achieves the best accuracy of 97.62%. Examining the combination sets, we find phenomena IDs 3, 4, 5, 7 and 8 appear quite often in the top 4 feature settings of each combination set. In fact, this setting achieves an accuracy of 95.24%, which is the best performance in  $C_5^9$  combination set. On the one hand, adding more phenomena into (3, 4, 5, 7, 8) setting does not have much performance difference.

In the above experiments, we do all the analyses on the corpus annotated with linguistic phenomena by human. We aim at knowing the ulti-

mate performance of *TE* recognition systems embodying human knowledge in the inference. The human knowledge in the inference cannot be captured by *TE* recognition systems fully correctly. In the later experiments, we explore the five critical features, (3, 4, 5, 7, 8), and examine how the performance is affected if they are extracted automatically.

### 3 Negative Entailment Phenomena Extraction

The experimental results in Section 2.2 show that disconnected relation, exclusive argument, exclusive relation, missing argument, and missing relation are significant. We follow the definitions of Sammons et al. (2010) and show them as follows.

(a) Disconnected Relation. The arguments and the relations in Hypothesis (*H*) are all matched by counterparts in Text (*T*). None of the arguments in *T* is connected to the matching relation.

(b) Exclusive Argument. There is a relation common to both the hypothesis and the text, but one argument is matched in a way that makes *H* contradict *T*.

(c) Exclusive Relation. There are two or more arguments in the hypothesis that are also related in the text, but by a relation that means *H* contradicts *T*.

(d) Missing Argument. Entailment fails because an argument in the Hypothesis is not present in the Text, either explicitly or implicitly.

(e) Missing Relation. Entailment fails because a relation in the Hypothesis is not present in the Text, either explicitly or implicitly.

To model the annotator’s inference process, we must first determine the arguments and the relations existing in *T* and *H*, and then align the arguments and relations in *H* to the related ones in *T*. It is easy for human to find the important parts in a text description in the inference process, but it is challenging for a machine to determine what words are important and what are not, and to detect the boundary of arguments and relations. Moreover, two arguments (relations) of strong semantic relatedness are not always literally identical.

In the following, a method is proposed to extract the phenomena from *T-H* pairs automatically. Before extraction, the English *T-H* pairs are pre-processed by numerical character transformation, POS tagging, and dependency parsing with Stanford Parser (Marneffe, et al., 2006;

Levy and Manning, 2003), and stemming with NLTK (Bird, 2006).

#### 3.1 A feature extraction method

Given a *T-H* pair, we first extract 4 sets of noun phrases based on their POS tags, including {noun in *H*}, {named entity (nnp) in *H*}, {compound noun (cnn) in *T*}, and {compound noun (cnn) in *H*}. Then, we extract 2 sets of relations, including {relation in *H*} and {relation in *T*}, where each relation in the sets is in a form of *Predicate*(*Argument*1, *Argument*2). Some typical examples of relations are *verb*(*subject*, *object*) for verb phrases, *neg*(*A*, *B*) for negations, *num*(*Noun*, *number*) for numeric modifier, and *tmod*(*C*, *temporal argument*) for temporal modifier. A predicate has only 2 arguments in this representation. Thus, a di-transitive verb is in terms of two relations.

Instead of measuring the relatedness of *T-H* pairs by comparing *T* and *H* on the predicate-argument structure (Wang and Zhang, 2009), our method tries to find the five negative entailment phenomena based on the similar representation. Each of the five negative entailment phenomena is extracted as follows according to their definitions. To reduce the error propagation which may be arisen from the parsing errors, we directly match those nouns and named entities appearing in *H* to the text *T*. Furthermore, we introduce WordNet to align arguments in *H* to *T*.

(a) Disconnected Relation. If (1) for each  $a \in \{\text{noun in } H\} \cup \{\text{nnp in } H\} \cup \{\text{cnn in } H\}$ , we can find  $a \in T$  too, and (2) for each  $r_1 = h(a_1, a_2) \in \{\text{relation in } H\}$ , we can find a relation  $r_2 = h(a_3, a_4) \in \{\text{relation in } T\}$  with the same header  $h$ , but with different arguments, i.e.,  $a_3 \neq a_1$  and  $a_4 \neq a_2$ , then we say the *T-H* pair has the “Disconnected Relation” phenomenon.

(b) Exclusive Argument. If there exist a relation  $r_1 = h(a_1, a_2) \in \{\text{relation in } H\}$ , and a relation  $r_2 = h(a_3, a_4) \in \{\text{relation in } T\}$  where both relations have the same header  $h$ , but either the pair  $(a_1, a_3)$  or the pair  $(a_2, a_4)$  is an antonym by looking up WordNet, then we say the *T-H* pair has the “Exclusive Argument” phenomenon.

(c) Exclusive Relation. If there exist a relation  $r_1 = h_1(a_1, a_2) \in \{\text{relation in } T\}$ , and a relation  $r_2 = h_2(a_1, a_2) \in \{\text{relation in } H\}$  where both relations have the same arguments, but  $h_1$  and  $h_2$  have the opposite meanings by consulting WordNet, then we say that the *T-H* pair has the “Exclusive Relation” phenomenon.



(d) Missing Argument. For each argument  $a_1 \in \{\text{noun in } H\} \cup \{\text{nnp in } H\} \cup \{\text{cnn in } H\}$ , if there does not exist an argument  $a_2 \in T$  such that  $a_1 = a_2$ , then we say that the  $T$ - $H$  pair has “Missing Argument” phenomenon.

(e) Missing Relation. For each relation  $r_1 = h_1(a_1, a_2) \in \{\text{relation in } H\}$ , if there does not exist a relation  $r_2 = h_2(a_3, a_4) \in \{\text{relation in } T\}$  such that  $h_1 = h_2$ , then we say that the  $T$ - $H$  pair has “Missing Relation” phenomenon.

### 3.2 Experiments and discussion

The following two datasets are used in English TE recognition experiments.

(a) 210 pairs from part of RTE-5 test set. The 210  $T$ - $H$  pairs are annotated with the linguistic phenomena by human annotators. They are selected from the 600 pairs in RTE-5 test set, including 51% ENTAILMENT and 49% NO ENTAILMENT.

(b) 600 pairs of RTE-5 test set. The original RTE-5 test set, including 50% ENTAILMENT and 50% NO ENTAILMENT.

Table 3 shows the performances of  $TE$  recognition. The “Machine-annotated” and the “Human-annotated” columns denote that the phenomena annotated by machine and human are used in the evaluation respectively. Using “Human-annotated” phenomena can be seen as the upper-bound of the experiments. The performance of using machine-annotated features in 210-pair and 600-pair datasets is 52.38% and 59.17% respectively.

Though the performance of using the phenomena extracted automatically by machine is not comparable to that of using the human annotated ones, the accuracy achieved by using only 5 features (59.17%) is just a little lower than the average accuracy of all runs in RTE-5 formal runs (60.36%) (Bentivogli, et al., 2009). It shows that the significant phenomena are really effective in dealing with entailment recognition. If we can improve the performance of the automatic phenomena extraction, it may make a great progress on the textual entailment.

| Phenomena             | 210 pairs         |                 | 600 pairs         |
|-----------------------|-------------------|-----------------|-------------------|
|                       | Machine-annotated | Human-annotated | Machine-annotated |
| Disconnected Relation | 50.95%            | 57.62%          | 54.17%            |
| Exclusive Argument    | 50.95%            | 61.90%          | 55.67%            |
| Exclusive Relation    | 50.95%            | 56.67%          | 51.33%            |
| Missing Argument      | 53.81%            | 69.52%          | 56.17%            |
| Missing Relation      | 50.95%            | 68.57%          | 52.83%            |
| All                   | 52.38%            | 95.24%          | 59.17%            |

Table 3: Accuracy of textual entailment recognition using the extracted phenomena as features.

## 4 Negative Entailment Phenomena in Chinese RITE Dataset

To make sure if negative entailment phenomena exist in other languages, we apply the methodologies in Sections 2 and 3 to the *RITE* dataset in NTCIR-9. We annotate all the 9 negative entailment phenomena on Chinese  $T$ - $H$  pairs according to the definitions by Sammons et al. (2010) and analyze the effects of various combinations of the phenomena on the new annotated Chinese data. The accuracy of using all the 9 phenomena as features (i.e.,  $C_9^9$  setting) is 91.11%. It shows the same tendency as the analyses on English data. The significant negative entailment phenomena on Chinese data, i.e., (3, 4, 5, 7, 8), are also identical to those on English data. The model using only 5 phenomena achieves an accuracy of 90.78%, which is very close to the performance using all phenomena.

We also classify the entailment relation using the phenomena extracted automatically by the similar method shown in Section 3.1, and get a similar result. The accuracy achieved by using the five automatically extracted phenomena as features is 57.11%, and the average accuracy of all runs in NTCIR-9 RITE task is 59.36% (Shima, et al., 2011). Compared to the other methods using a lot of features, only a small number of binary features are used in our method. Those observations establish what we can call a useful baseline for  $TE$  recognition.

## 5 Conclusion

In this paper we conclude that the negative entailment phenomena have a great effect in dealing with  $TE$  recognition. Systems with human annotated knowledge achieve very good performance. Experimental results show that not only can it be applied to the English  $TE$  problem, but also has the similar effect on the Chinese  $TE$  recognition. Though the automatic extraction of the negative entailment phenomena still needs a lot of efforts, it gives us a new direction to deal with the  $TE$  problem.

The fundamental issues such as determining the boundary of the arguments and the relations, finding the implicit arguments and relations, verifying the antonyms of arguments and relations, and determining their alignments need to be further examined to extract correct negative entailment phenomena. Besides, learning-based approaches to extract phenomena and multi-class  $TE$  recognition will be explored in the future.

## Acknowledgments

This research was partially supported by Excellent Research Projects of National Taiwan University under contract 102R890858 and 2012 Google Research Award.

## References

- Ion Androutsopoulos and Prodrornos Malakasiotis. 2010. A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research*, 38:135-187.
- Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2011. The seventh PASCAL recognizing textual entailment challenge. In *Proceedings of the 2011 Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA..
- Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the 2009 Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA.
- Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 69-72.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1-27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. *Lecture Notes in Computer Science*, 3944:177-190.
- Adrian Iftene and Mihai Alex Moruz. 2009. UAIC Participation at RTE5. In *Proceedings of the 2009 Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA.
- Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL 2003)*, pages 439-446.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *The Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449-454.
- Mark Sammons, V.G.Vinod Vydiswaran, and Dan Roth. 2010. Ask not what textual entailment can do for you... In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1199-1208, Uppsala, Sweden.
- 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 the NTCIR-9 Workshop Meeting*, Tokyo, Japan.
- Lucy Vanderwende, Arul Menezes, and Rion Snow. 2006. Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation. In *Proceedings of the Second PASCAL Challenges Workshop*.
- 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*, pages 784-792, Singapore.

# Recognizing Partial Textual Entailment

Omer Levy<sup>†</sup>

Torsten Zesch<sup>§</sup>

Ido Dagan<sup>†</sup>

Iryna Gurevych<sup>§</sup>

<sup>†</sup> Natural Language Processing Lab  
Computer Science Department  
Bar-Ilan University

<sup>§</sup> Ubiquitous Knowledge Processing Lab  
Computer Science Department  
Technische Universität Darmstadt

## Abstract

Textual entailment is an asymmetric relation between two text fragments that describes whether one fragment can be inferred from the other. It thus cannot capture the notion that the target fragment is “almost entailed” by the given text. The recently suggested idea of *partial textual entailment* may remedy this problem. We investigate partial entailment under the faceted entailment model and the possibility of adapting existing textual entailment methods to this setting. Indeed, our results show that these methods are useful for recognizing partial entailment. We also provide a preliminary assessment of how partial entailment may be used for recognizing (complete) textual entailment.

## 1 Introduction

Approaches for applied semantic inference over texts gained growing attention in recent years, largely triggered by the *textual entailment* framework (Dagan et al., 2009). Textual entailment is a generic paradigm for semantic inference, where the objective is to recognize whether a textual hypothesis (labeled  $H$ ) can be inferred from another given text (labeled  $T$ ). The definition of textual entailment is in some sense strict, in that it requires that  $H$ 's meaning be implied by  $T$  in its entirety. This means that from an entailment perspective, a text that contains the main ideas of a hypothesis, but lacks a minor detail, is indiscernible from an entirely unrelated text. For example, if  $T$  is “muscles move bones”, and  $H$  “the main job of muscles is to move bones”, then  $T$  does *not* entail  $H$ , and we are left with no sense of how *close* ( $T, H$ ) were to entailment.

In the related problem of semantic text similarity, gradual measures are already in use. The semantic text similarity challenge in SemEval 2012

(Agirre et al., 2012) explicitly defined different levels of similarity from 5 (semantic equivalence) to 0 (no relation). For instance, 4 was defined as “the two sentences are mostly equivalent, but some unimportant details differ”, and 3 meant that “the two sentences are roughly equivalent, but some important information differs”. Though this modeling does indeed provide finer-grained notions of similarity, it is not appropriate for semantic inference for two reasons. First, the term “important information” is vague; what makes one detail more important than another? Secondly, similarity is not sufficiently well-defined for sound semantic inference; for example, “snowdrops bloom in summer” and “snowdrops bloom in winter” may be similar, but have contradictory meanings. All in all, these measures of similarity do not quite capture the gradual relation needed for semantic inference.

An appealing approach to dealing with the rigidity of textual entailment, while preserving the more precise nature of the entailment definition, is by breaking down the hypothesis into components, and attempting to recognize whether each one is individually entailed by  $T$ . It is called *partial textual entailment*, because we are only interested in recognizing whether a single element of the hypothesis is entailed. To differentiate the two tasks, we will refer to the original textual entailment task as *complete textual entailment*.

Partial textual entailment was first introduced by Nielsen et al. (2009), who presented a machine learning approach and showed significant improvement over baseline methods. Recently, a public benchmark has become available through the Joint Student Response Analysis and 8th Recognizing Textual Entailment (RTE) Challenge in SemEval 2013 (Dzikovska et al., 2013), on which we focus in this paper.

Our goal in this paper is to investigate the idea of partial textual entailment, and assess whether

existing complete textual entailment methods can be used to recognize it. We assume the facet model presented in SemEval 2013, and adapt existing technologies to the task of recognizing partial entailment (Section 3). Our work further expands upon (Nielsen et al., 2009) by evaluating these adapted methods on the new RTE-8 benchmark (Section 4). Partial entailment may also facilitate an alternative divide and conquer approach to complete textual entailment. We provide an initial investigation of this approach (Section 5).

## 2 Task Definition

In order to tackle partial entailment, we need to find a way to decompose a hypothesis. Nielsen et al. (2009) defined a model of *facets*, where each such facet is a pair of words in the hypothesis and the direct semantic relation connecting those two words. We assume the simplified model that was used in RTE-8, where the relation between the words is not explicitly stated. Instead, it remains unstated, but its interpreted meaning should correspond to the manner in which the words are related in the hypothesis. For example, in the sentence “the main job of muscles is to move bones”, the pair (*muscles*, *move*) represents a facet. While it is not explicitly stated, reading the original sentence indicates that *muscles* is the agent of *move*.

Formally, the task of recognizing faceted entailment is a binary classification task. Given a text  $T$ , a hypothesis  $H$ , and a facet within the hypothesis ( $w_1, w_2$ ), determine whether the facet is either *expressed* or *unaddressed* by the text. Nielsen et al included additional classes such as *contradicting*, but in the scope of this paper we will only tend to the binary case, as was done in RTE-8.

Consider the following example:

**T:** Muscles generate movement in the body.

**H:** The main job of muscles is to move bones.

The facet (*muscles*, *move*) refers to the agent role in  $H$ , and is expressed by  $T$ . However, the facet (*move*, *bones*), which refers to a theme or direct object relation in  $H$ , is unaddressed by  $T$ .

## 3 Recognizing Faceted Entailment

Our goal is to investigate whether existing entailment recognition approaches can be adapted to recognize faceted entailment. Hence, we specified relatively simple decision mechanisms over a set of entailment detection modules. Given a text

and a facet, each module reports whether it recognizes entailment, and the decision mechanism then determines the binary class (*expressed* or *unaddressed*) accordingly.

### 3.1 Entailment Modules

Current textual entailment systems operate across different linguistic levels, mainly on lexical inference and syntax. We examined three representative modules that reflect these levels: *Exact Match*, *Lexical Inference*, and *Syntactic Inference*.

**Exact Match** We represent  $T$  as a bag-of-words containing all tokens and lemmas appearing in the text. We then check whether both facet lemmas  $w_1, w_2$  appear in the text’s bag-of-words. Exact matching was used as a baseline in previous recognizing textual entailment challenges (Bentivogli et al., 2011), and similar methods of lemma-matching were used as a component in recognizing textual entailment systems (Clark and Harrison, 2010; Shnarch et al., 2011).

**Lexical Inference** This feature checks whether both facet words, or semantically related words, appear in  $T$ . We use WordNet (Fellbaum, 1998) with the Resnik similarity measure (Resnik, 1995) and count a facet term  $w_i$  as matched if the similarity score exceeds a certain threshold (0.9, empirically determined on the training set). Both  $w_1$  and  $w_2$  must match for this module’s entailment decision to be positive.

**Syntactic Inference** This module builds upon the open source<sup>1</sup> Bar-Ilan University Textual Entailment Engine (BIUTEE) (Stern and Dagan, 2011). BIUTEE operates on dependency trees by applying a sequence of knowledge-based transformations that converts  $T$  into  $H$ . It determines entailment according to the “cost” of generating the hypothesis from the text. The cost model can be automatically tuned with a relatively small training set. BIUTEE has shown state-of-the-art performance on previous recognizing textual entailment challenges (Stern and Dagan, 2012).

Since BIUTEE processes dependency trees, both  $T$  and the facet must be parsed. We therefore extract a path in  $H$ ’s dependency tree that represents the facet. This is done by first parsing  $H$ , and then locating the two nodes whose words compose the facet. We then find their lowest common ancestor (LCA), and extract the path  $P$  from  $w_1$  to

<sup>1</sup>[cs.biu.ac.il/~nlp/downloads/biutee](http://cs.biu.ac.il/~nlp/downloads/biutee)

$w_2$  through the LCA. This path is in fact a dependency tree. BIUTEE can now be given  $T$  and  $P$  (as the hypothesis), and try to recognize whether the former entails the latter.

### 3.2 Decision Mechanisms

We started our experimentation process by defining *Exact Match* as a baseline. Though very simple, this unsupervised baseline performed surprisingly well, with 0.96 precision and 0.32 recall on expressed facets of the training data. Given its very high precision, we decided to use this module as an initial filter, and employ the others for classifying the “harder” cases.

We present all the mechanisms that we tested:

**Baseline** *Exact*

**BaseLex** *Exact*  $\vee$  *Lexical*

**BaseSyn** *Exact*  $\vee$  *Syntactic*

**Disjunction** *Exact*  $\vee$  *Lexical*  $\vee$  *Syntactic*

**Majority** *Exact*  $\vee$  (*Lexical*  $\wedge$  *Syntactic*)

Note that since every facet that *Exact Match* classifies as *expressed* is also *expressed* by *Lexical Inference*, *BaseLex* is essentially *Lexical Inference* on its own, and *Majority* is equivalent to the majority rule on all three modules.

## 4 Empirical Evaluation

### 4.1 Dataset: Student Response Analysis

We evaluated our methods as part of RTE-8. The challenge focuses on the domain of scholastic quizzes, and attempts to emulate the meticulous marking process that teachers do on a daily basis. Given a question, a student’s response, and a reference answer, the task of *student response analysis* is to determine whether the student answered correctly. This task can be approximated as a special case of textual entailment; by assigning the student’s answer as  $T$  and the reference answer as  $H$ , we are basically asking whether one can infer the correct (reference) answer from the student’s response.

Recall the example from Section 2. In this case,  $H$  is a reference answer to the question:

**Q:** What is the main job of muscles?

$T$  is essentially the student answer, though it is also possible to define  $T$  as the union of both the question and the student answer. In this work, we chose to exclude the question.

There were two tracks in the challenge: complete textual entailment (the main task) and partial

|             | Unseen<br>Answers | Unseen<br>Questions | Unseen<br>Domains |
|-------------|-------------------|---------------------|-------------------|
| Baseline    | .670              | .688                | .731              |
| BaseLex     | .756              | .710                | .760              |
| BaseSyn     | .744              | .733                | .770              |
| Disjunction | .695              | .655                | .703              |
| Majority    | <b>.782</b>       | <b>.765</b>         | <b>.816</b>       |

Table 1: Micro-averaged  $F_1$  on the faceted SciEntsBank test set.

entailment (the pilot task). Both tasks made use of the SciEntsBank corpus (Dzikovska et al., 2012), which is annotated at facet-level, and provides a convenient test-bed for evaluation of both partial and complete entailment. This dataset was split into train and test subsets. The test set has 16,263 facet-response pairs based on 5,106 student responses over 15 domains (learning modules). Performance was measured using micro-averaged  $F_1$ , over three different scenarios:

**Unseen Answers** Classify new answers to questions seen in training. Contains 464 student responses.

**Unseen Questions** Classify new answers to questions that were not seen in training, but other questions from the same domain were. Contains 631 student responses.

**Unseen Domains** Classify new answers to unseen questions from unseen domains. Contains 4,011 student responses.

### 4.2 Results

Table 1 shows the  $F_1$ -measure of each configuration in each scenario. There is some variance between the different scenarios; this may be attributed to the fact that there are much fewer *Unseen Answers* and *Unseen Questions* instances. In all cases, *Majority* significantly outperformed the other configurations. While *BaseLex* and *BaseSyn* improve upon the baseline, they seem to make different mistakes, in particular false positives. Their conjunction is thus a more conservative indicator of entailment, and proves helpful in terms of  $F_1$ . All improvements over the baseline were found to be statistically significant using McNemar’s test with  $p < 0.01$  (excluding *Disjunction*). It is also interesting to note that the systems’ performance does not degrade in “harder” scenarios; this is a result of the mostly unsupervised nature of our modules.

Unfortunately, our system was the only submission in the partial entailment pilot track of RTE-8, so we have no comparisons with other systems. However, the absolute improvement from the exact-match baseline to the more sophisticated *Majority* is in the same ballpark as that of the best systems in previous recognizing textual entailment challenges. For instance, in the previous recognizing textual entailment challenge (Bentivogli et al., 2011), the best system yielded an  $F_1$  score of 0.48, while the baseline scored 0.374. We can therefore conclude that existing approaches for recognizing textual entailment can indeed be adapted for recognizing partial entailment.

## 5 Utilizing Partial Entailment for Recognizing Complete Entailment

Encouraged by our results, we ask whether the same algorithms that performed well on the faceted entailment task can be used for recognizing complete textual entailment. We performed an initial experiment that examines this concept and sheds some light on the potential role of partial entailment as a possible facilitator for complete entailment.

We suggest the following 3-stage architecture:

1. Decompose the hypothesis into facets.
2. Determine whether each facet is entailed.
3. Aggregate the individual facet results and decide on complete entailment accordingly.

**Facet Decomposition** For this initial investigation, we use the facets provided in SciEntsBank; i.e. we assume that the step of *facet decomposition* has already been carried out. When the dataset was created for RTE-8, many facets were extracted automatically, but only a subset was selected. The facet selection process was done manually, as part of the dataset’s annotation. For example, in “the main job of muscles is to move bones”, the facet (*job, muscles*) was not selected, because it was not critical for answering the question. We refer to the issue of relying on manual input further below.

**Recognizing Faceted Entailment** This step was carried out as explained in the previous sections. We used the *Baseline* configuration and *Majority*, which performed best in our experiments above. In addition, we introduce *GoldBased* that uses the gold annotation of faceted entailment, and thus

|              | Unseen Answers | Unseen Questions | Unseen Domains |
|--------------|----------------|------------------|----------------|
| Baseline     | .575           | .582             | .683           |
| Majority     | .707           | .673             | .764           |
| GoldBased    | .842           | .897             | .852           |
| BestComplete | .773           | .745             | .712           |

Table 2: Micro-averaged  $F_1$  on the 2-way complete entailment SciEntsBank test set.

provides a certain upper bound on the performance of determining complete entailment based on facets.

**Aggregation** We chose the simplest sensible aggregation rule to decide on overall entailment: a student answer is classified as *correct* (i.e. it entails the reference answer) if it expresses each of the reference answer’s facets. Although this heuristic is logical from a strict entailment perspective, it might yield false negatives on this particular dataset. This happens because tutors may sometimes grade answers as valid even if they omit some less important, or indirectly implied, facets.

Table 2 shows the experiment’s results. The first thing to notice is that *GoldBased* is not perfect. There are two reasons for this behavior. First, the task of student response analysis is only an approximation of textual entailment, albeit a good one. This discrepancy was also observed by the RTE-8 challenge organizers (Dzikovska et al., 2013). The second reason is because some of the original facets were filtered when creating the dataset. This caused both false positives (when important facets were filtered out) and false negatives (when unimportant facets were retained).

Our *Majority* mechanism, which requires that the two underlying methods for partial entailment detection (*Lexical Inference* and *Syntactic Inference*) agree on a positive classification, bridges about half the gap from the baseline to the gold based method. As a rough point of comparison, we also show the performance of *BestComplete*, the winning entry in each setting of the RTE-8 main task. This measure is not directly comparable to our facet-based systems, because it did not rely on manually selected facets, and due to some variations in the dataset size (about 20% of the student responses were not included in the pilot task dataset). However, these results may indicate the

prospects of using faceted entailment for complete entailment recognition, suggesting it as an attractive research direction.

## 6 Conclusion and Future Work

In this paper, we presented an empirical attempt to tackle the problem of partial textual entailment. We demonstrated that existing methods for recognizing (complete) textual entailment can be successfully adapted to this setting. Our experiments showed that boolean combinations of these methods yield good results. Future research may add additional features and more complex feature combination methods, such as weighted sums tuned by machine learning. Furthermore, our work focused on a specific decomposition model – faceted entailment. Other flavors of partial entailment should be investigated as well. Finally, we examined the possibility of utilizing partial entailment for recognizing complete entailment in a semi-automatic setting, which relied on the manual facet annotation in the RTE-8 dataset. Our preliminary results suggest that this approach is indeed feasible, and warrant further research on facet-based entailment methods that rely on fully-automatic facet extraction.

## Acknowledgements

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT). We would like to thank the Minerva Foundation for facilitating this cooperation with a short term research grant.

## References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 385–393, Montreal, Canada.

Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. 2011. The seventh pascal recognizing textual entailment challenge. *Proceedings of TAC*.

Peter Clark and Phil Harrison. 2010. Blue-lite: a knowledge-based lexical entailment system for rte6. *Proc. of TAC*.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rationale, evaluation and approaches. *Natural Language Engineering*, 15(4):i–xvii.

Myroslava O Dzikovska, Rodney D Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210. Association for Computational Linguistics.

Myroslava O. Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *\*SEM 2013: The First Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Rodney D Nielsen, Wayne Ward, and James H Martin. 2009. Recognizing entailment in intelligent tutoring systems. *Natural Language Engineering*, 15(4):479–501.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pages 448–453.

Eyal Shnarch, Jacob Goldberger, and Ido Dagan. 2011. A probabilistic modeling framework for lexical entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 558–563, Portland, Oregon, USA, June. Association for Computational Linguistics.

Asher Stern and Ido Dagan. 2011. A confidence model for syntactically-motivated entailment proofs. In *Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011)*, pages 455–462.

Asher Stern and Ido Dagan. 2012. Biutee: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pages 73–78, Jeju Island, Korea, July. Association for Computational Linguistics.

# Sentence Level Dialect Identification in Arabic

**Heba Elfardy**

Department of Computer Science  
Columbia University  
heba@cs.columbia.edu

**Mona Diab**

Department of Computer Science  
The George Washington University  
mtdiab@gwu.edu

## Abstract

This paper introduces a supervised approach for performing sentence level dialect identification between Modern Standard Arabic and Egyptian Dialectal Arabic. We use token level labels to derive sentence-level features. These features are then used with other core and meta features to train a generative classifier that predicts the correct label for each sentence in the given input text. The system achieves an accuracy of 85.5% on an Arabic online-commentary dataset outperforming a previously proposed approach achieving 80.9% and reflecting a significant gain over a majority baseline of 51.9% and two strong baseline systems of 78.5% and 80.4%, respectively.

## 1 Introduction

The Arabic language exists in a state of Diglossia (Ferguson, 1959) where the standard form of the language, Modern Standard Arabic (MSA) and the regional dialects (DA) live side-by-side and are closely related. MSA is the language used in education, scripted speech and official settings while DA is the native tongue of Arabic speakers. Arabic dialects may be divided into five main groups: Egyptian (including Libyan and Sudanese), Levantine (including Lebanese, Syrian, Palestinian and Jordanian), Gulf, Iraqi and Moroccan (Maghrebi) (Habash, 2010). Even though these dialects did not originally exist in a written form, they are pervasively present in social media text (normally mixed with MSA) nowadays. DA does not have a standard orthography leading to many spelling variations and inconsistencies. Linguistic Code switching (LCS) between MSA and DA happens both intra-sententially and inter-sententially. LCS in Arabic poses a serious challenge for almost all NLP tasks since MSA and DA

differ on all levels of linguistic representation. For example, MSA trained tools perform very badly when applied directly to DA or to a code-switched DA-MSA text. Hence a need for a robust dialect identification tool as a preprocessing step arises both on the word and sentence levels.

In this paper, we focus on the problem of dialect identification on the sentence level. We propose a supervised approach for identifying whether a given sentence is prevalently MSA or Egyptian DA (EDA). The system uses the approach that was presented in (Elfardy et al., 2013) to perform token dialect identification. The token level decisions are then combined with other features to train a generative classifier that tries to predict the class of the given sentence. The presented system outperforms the approach presented by Zaidan and Callison-Burch (2011) on the same dataset using 10-fold cross validation.

## 2 Related Work

Dialect Identification in Arabic is crucial for almost all NLP tasks, yet most of the research in Arabic NLP, with few exceptions, is targeted towards MSA. Biadisy et al. (2009) present a system that identifies dialectal words in speech and their dialect of origin through the acoustic signals. Salloum and Habash (2011) tackle the problem of DA to English Machine Translation (MT) by pivoting through MSA. The authors present a system that applies transfer rules from DA to MSA then uses state of the art MSA to English MT system. Habash et al. (2012) present CODA, a Conventional Orthography for Dialectal Arabic that aims to standardize the orthography of all the variants of DA while Dasigi and Diab (2011) present an unsupervised clustering approach to identify orthographic variants in DA. Zaidan and Callison-Burch (2011) crawl a large dataset of MSA-DA news' commentaries. The authors annotate part of the dataset for sentence-level dialectalness on





of the sequence as being EDA. 5-grams are used for building the tokenized LMs (as opposed to 3-grams for the surface LMs)

ex. كده حرام و كثير على نا  
*kdh HrAm w+ ktyr Ely +nA*

4. **Tokenized & Orthography Normalized LMs: (Tokenized-CODA)** The data is tokenized as in (3) then orthography normalization is applied to the tokenized data.

ex. كده حرام و كثير على نا  
*kdh HrAm w+ kvyr Ely +nA*

In addition to the underlying token-level system, we use the following token-level features:

1. Percentage of words in the sentence that is analyzable by an MSA morphological analyzer.
2. Percentage of words in the sentence that is analyzable by an EDA morphological analyzer.
3. Percentage of words in the sentence that exists in a precompiled EDA lexicon.

**3.1.1.2 Perplexity-based Features:** We run each sentence through each of the MSA and EDA LMs and record the perplexity for each of them. The perplexity of a language model on a given test sentence;  $S(w_1, \dots, w_n)$  is defined as:

$$perplexity = (2)^{-(1/N) \sum_i \log_2(p(w_i|h_i))} \quad (1)$$

where  $N$  is the number of tokens in the sentence and  $h_i$  is the history of token  $w_i$ .

The perplexity conveys how confused the LM is about the given sentence so the higher the perplexity value, the less probable that the given sentence matches the LM.<sup>2</sup>

### 3.1.2 Meta Features.

These are the features that do not directly relate to the dialectalness of words in the given sentence but rather estimate how informal the sentence is and include:

- The percentage of punctuation, numbers, special-characters and words written in Roman script.

<sup>2</sup>We repeat this step for each of the preprocessing schemes explained in section 3.1.1.1

- The percentage of words having word-lengthening effects.
- Number of words & average word-length.
- Whether the sentence has consecutive repeated punctuation or not. (Binary feature, yes/no)
- Whether the sentence has an exclamation mark or not. (Binary feature, yes/no)
- Whether the sentence has emoticons or not. (Binary feature, yes/no)

## 3.2 Model Training

We use the WEKA toolkit (Hall et al., 2009) and the derived features to train a Naive-Bayes classifier. The classifier is trained and cross-validated on the gold-training data for each of our different configurations (Surface, CODAified, Tokenized & Tokenized-CODA).

We conduct two sets of experiments. In the first one, Experiment Set A, we split the data into a training set and a held-out test set. In the second set, Experiment Set B, we use the whole dataset for training without further splitting. For both sets of experiments, we apply 10-fold cross validation on the training data. While using a held-out test-set for evaluation (in the first set of experiments) is a better indicator of how well our approach performs on unseen data, only the results from the second set of experiments are directly comparable to those produced by Zaidan and Callison-Burch (2011).

## 4 Experiments

### 4.1 Data

We use the code-switched EDA-MSA portion of the crowd source annotated dataset by Zaidan and Callison-Burch (2011). The dataset consists of user commentaries on Egyptian news articles. Table 1 shows the statistics of the data.

|       | MSA Sent. | EDA Sent. | MSA Tok. | EDA Tok. |
|-------|-----------|-----------|----------|----------|
| Train | 12,160    | 11,274    | 300,181  | 292,109  |
| Test  | 1,352     | 1,253     | 32,048   | 32,648   |

Table 1: Number of EDA and MSA sentences and tokens in the training and test datasets. In Experiment Set A only the train-set is used to perform a 10-fold cross-validation and the test-set is used for evaluation. In experiment Set B, all data is used to perform the 10-fold cross-validation.

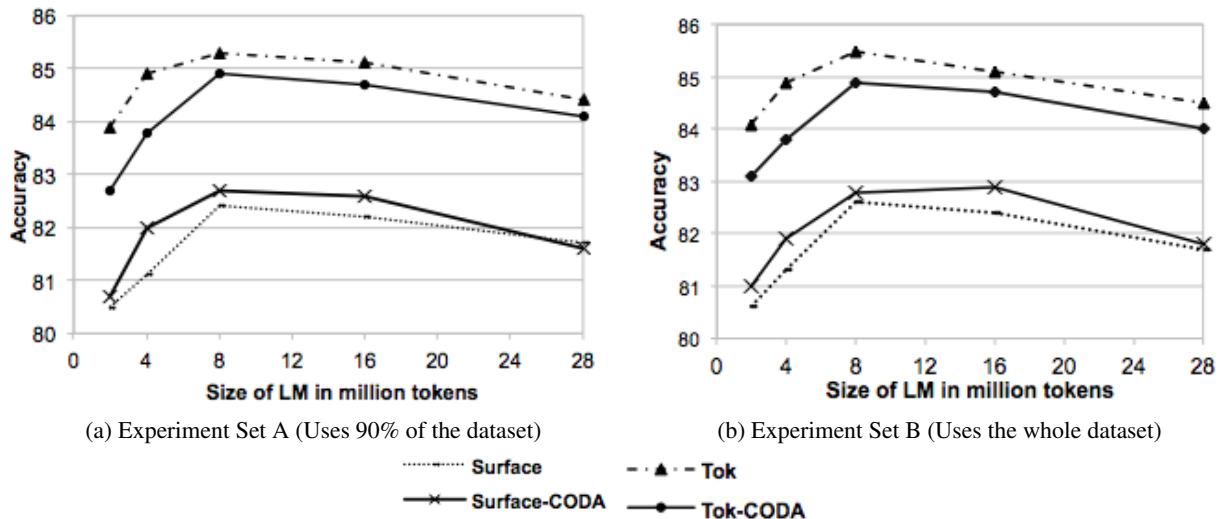


Figure 1: Learning curves for the different configurations (obtained by applying 10-fold cross validation on the training set.)

## 4.2 Baselines

We use four baselines. The first of which is a majority baseline (Maj-BL); that assigns all the sentences the label of the most frequent class observed in the training data. The second baseline (Token-BL) assumes that the sentence is EDA if more than 45% of its tokens are dialectal otherwise it assumes it is MSA.<sup>3</sup> The third baseline (Ppl-BL) runs each sentence through MSA & EDA LMs and assigns the sentence the class of the LM yielding the lower perplexity value. The last baseline (OZ-CCB-BL) is the result obtained by Zaidan and Callison-Burch (2011) which uses the same approach of our third baseline, Ppl-BL.<sup>4</sup> For Token-BL and Ppl-BL, the performance is calculated for all LM-sizes of the four different configurations: Surface, CODAified, Tokenized, Tokenized-CODA and the best performing configuration on the cross-validation set is used as the baseline system.

## 4.3 Results & Discussion

For each of the different configurations, we build a learning curve by varying the size of the LMs between 2M, 4M, 8M, 16M and 28M tokens. Figures 1a and 1b show the learning curves of the different configurations on the cross-validation set for experiments A & B respectively. In Table 2 we note that both CODA and Tokenized solve the data-sparseness issue hence they produce better results

<sup>3</sup>We experimented with different thresholds (15%, 30%, 45%, 60% and 75%) and the 45% threshold setting yielded

| Condition      | Exp. Set A  | Exp. Set B  |
|----------------|-------------|-------------|
| Maj-BL         | 51.9        | 51.9        |
| Token-BL       | 79.1        | 78.5        |
| Ppl-BL         | 80.4        | 80.4        |
| OZ-CCB-BL      | N/A         | 80.9        |
| Surface        | 82.4        | 82.6        |
| CODA           | 82.7        | 82.8        |
| Tokenized      | <b>85.3</b> | <b>85.5</b> |
| Tokenized-CODA | 84.9        | 84.9        |

Table 2: Performance Accuracies of the different configurations of the 8M LM (best-performing LM size) using 10-fold cross validation against the different baselines.

than Surface experimental condition. However, as mentioned earlier, CODA removes some dialectalness cues so the improvement resulting from using CODA is much less than that from using tokenization. Also when combining CODA with tokenization as in the condition Tokenized-CODA, the performance drops since in this case the sparseness issue has been already resolved by tokenization so adding CODA only removes dialectalness cues. For example *وكتير* *wktyr* ‘and a lot’ does not occur frequently in the data so when performing the tokenization it becomes *و كتير* *w+ ktyr* which on the contrary is frequent in the data. Adding

the best performance

<sup>4</sup>This baseline can only be compared to the results of the second set of experiments.

| Condition | Test Set    |
|-----------|-------------|
| Maj-BL    | 51.9        |
| Token-BL  | 77          |
| Ppl-BL    | 81.1        |
| Tokenized | <b>83.3</b> |

Table 3: Performance Accuracies of the best-performing configuration (Tokenized) on the held-out test set against the baselines Maj-BL, Token-BL and Ppl-BL.

Orthography-Normalization converts it to **و كثير**  $w+ kvyr$  which is more MSA-like hence the confusability increases.

All configurations outperform all baselines with the Tokenized configuration producing the best results. The performance of all systems drop as the size of the LM increases beyond 16M tokens. As indicated in (Elfardy et al., 2013) as the size of the MSA & EDA LMs increases, the shared ngrams increase leading to higher confusability between the classes of tokens in a given sentence. Table 3 presents the results on the held out dataset compared against three of the baselines, Maj-BL, Token-BL and Ppl-BL. We note that the Tokenized condition, the best performing condition, outperforms all baselines with a significant margin.

## 5 Conclusion

We presented a supervised approach for sentence level dialect identification in Arabic. The approach uses features from an underlying system for token-level identification of Egyptian Dialectal Arabic in addition to other core and meta features to decide whether a given sentence is MSA or EDA. We studied the impact of two types of pre-processing techniques (Tokenization and Orthography Normalization) as well as varying the size of the LM on the performance of our approach. The presented approach produced significantly better results than a previous approach in addition to beating the majority baseline and two other strong baselines.

## Acknowledgments

This work is supported by the Defense Advanced Research Projects Agency (DARPA) BOLT program under contract number HR0011-12-C-0014.

## References

- Fadi Biadisy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at the meeting of the European Association for Computational Linguistics (EACL), Athens, Greece*.
- Pradeep Dasigi and Mona Diab. 2011. Codact: Towards identifying orthographic variants in dialectal arabic. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJNLP), Chiangmai, Thailand*.
- Heba Elfardy and Mona Diab. 2012a. Simplified guidelines for the creation of large scale dialectal arabic annotations. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*.
- Heba Elfardy and Mona Diab. 2012b. Token level identification of linguistic code switching. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING), Mumbai, India*.
- Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2013. Code Switch Point Detection in Arabic. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB2013), MediaCity, UK, June*.
- Ramy Eskander, Nizar Habash, Owen Rambow, and Nadi Tomeh. 2013. Processing Spontaneous Orthography. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, GA*.
- Ferguson. 1959. *Diglossia*. Word 15. 325340.
- Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+ token: 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*, pages 102–109.
- Nizar Habash, Mona Diab, and Owen Rabmow. 2012. Conventional orthography for dialectal arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Istanbul*.
- Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, GA*.
- Nizar Habash. 2010. Introduction to arabic natural language processing. *Advances in neural information processing systems*.

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.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of ACL*, pages 37–41.

# Leveraging Domain-Independent Information in Semantic Parsing

**Dan Goldwasser**

University of Maryland  
College Park, MD 20740  
goldwas1@umiacs.umd.edu

**Dan Roth**

University of Illinois  
Urbana, IL 61801  
danr@illinois.edu

## Abstract

Semantic parsing is a domain-dependent process by nature, as its output is defined over a set of domain symbols. Motivated by the observation that interpretation can be decomposed into domain-dependent and independent components, we suggest a novel interpretation model, which augments a domain dependent model with abstract information that can be shared by multiple domains. Our experiments show that this type of information is useful and can reduce the annotation effort significantly when moving between domains.

## 1 Introduction

Natural Language (NL) understanding can be intuitively understood as a general capacity, mapping words to entities and their relationships. However, current work on automated NL understanding (typically referenced as *semantic parsing* (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Chen and Mooney, 2008; Kwiatkowski et al., 2010; Börschinger et al., 2011)) is restricted to a given output domain<sup>1</sup> (or task) consisting of a closed set of meaning representation symbols, describing domains such as robotic soccer, database queries and flight ordering systems.

In this work, we take a first step towards constructing a semantic interpreter that can leverage information from multiple tasks. This is not a straight forward objective – the domain specific nature of semantic interpretation, as described in the current literature, does not allow for an easy move between domains. For example, a system trained for the task of understanding database queries will not be of any use when it will be given a sentence describing robotic soccer instructions.

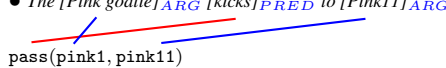
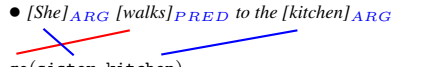
In order to understand this difficulty, a closer look at semantic parsing is required. Given a sentence, the interpretation process breaks it into a

<sup>1</sup>The term *domain* is overloaded in NLP; in this work we use it to refer to the set of output symbols.

set of interdependent decisions, which rely on an underlying representation mapping words to symbols and syntactic patterns into compositional decisions. This representation takes into account domain specific information (e.g., a lexicon mapping phrases to a domain predicate) and is therefore of little use when moving to a different domain.

In this work, we attempt to develop a domain independent approach to semantic parsing. We do it by developing a layer of representation that is applicable to multiple domains. Specifically, we add an intermediate layer capturing shallow semantic relations between the input sentence constituents. Unlike semantic parsing which maps the input to a closed set of symbols, this layer can be used to identify general predicate-argument structures in the input sentence. The following example demonstrates the key idea behind our representation – two sentences from two different domains have a similar intermediate structure.

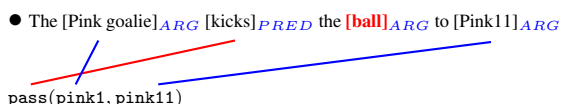
**Example 1.** *Domains with similar intermediate structures*

- The [Pink goalie]<sub>ARG</sub> [kicks]<sub>PRED</sub> to [Pink11]<sub>ARG</sub>  
  
pass(pink1, pink11)
- [She]<sub>ARG</sub> [walks]<sub>PRED</sub> to the [kitchen]<sub>ARG</sub>  
  
go(sister, kitchen)

In this case, the constituents of the first sentence (from the Robocup domain (Chen and Mooney, 2008)), are assigned domain-independent predicate-argument labels (e.g., the word corresponding to a logical function is identified as a *PRED*). Note that it does not use any domain specific information, for example, the *PRED* label assigned to the word “kicks” indicates that this word is the predicate of the sentence, not a specific domain predicate (e.g., *pass*(·)). The intermediate layer can be reused across domains. The logical output associated with the second sentence is taken from a different domain, using a different set of output symbols, however it shares the same predicate-argument structure.

Despite the idealized example, in practice,

leveraging this information is challenging, as the logical structure is assumed to only weakly correspond to the domain-independent structure, a correspondence which may change in different domains. The mismatch between the domain independent (linguistic) structure and logical structures typically stems from technical considerations, as the domain logical language is designed according to an application-specific logic and not according to linguistic considerations. This situation is depicted in the following example, in which one of the domain-independent labels is omitted.



In order to overcome this difficulty, we suggest a flexible model that is able to leverage the supervision provided in one domain to learn an abstract intermediate layer, and show empirically that it learns a robust model, improving results significantly in a second domain.

## 2 Semantic Interpretation Model

Our model consists of both domain-dependent (mapping between text and a closed set of symbols) and domain independent (abstract predicate-argument structures) information. We formulate the joint interpretation process as a structured prediction problem, mapping a NL input sentence ( $\mathbf{x}$ ), to its highest ranking interpretation and abstract structure ( $\mathbf{y}$ ). The decision is quantified using a linear objective, which uses a vector  $w$ , mapping features to weights and a feature function  $\Phi$  which maps the output decision to a feature vector. The output interpretation  $\mathbf{y}$  is described using a subset of first order logic, consisting of typed constants (e.g., robotic soccer player), functions capturing relations between entities, and their properties (e.g.,  $\text{pass}(x, y)$ , where  $\text{pass}$  is a function symbol and  $x, y$  are typed arguments). We use data taken from two grounded domains, describing robotic soccer events and household situations.

We begin by formulating the domain-specific process. We follow (Goldwasser et al., 2011; Clarke et al., 2010) and formalize semantic inference as an Integer Linear Program (ILP). Due to space consideration, we provide a brief description (see (Clarke et al., 2010) for more details). We then proceed to augment this model with domain-independent information, and connect the two models by constraining the ILP model.

## 2.1 Domain-Dependent Model

Interpretation is composed of several decisions, capturing mapping of input tokens to logical fragments (first order) and their composition into larger fragments (second). We encode a first-order decision as  $\alpha_{cs}$ , a binary variable indicating that constituent  $c$  is aligned with the logical symbol  $s$ . A second-order decision  $\beta_{cs,dt}$ , is encoded as a binary variable indicating that the symbol  $t$  (associated with constituent  $d$ ) is an argument of a function  $s$  (associated with constituent  $c$ ). The overall inference problem (Eq. 1) is as follows:

$$F_{\mathbf{w}}(\mathbf{x}) = \arg \max_{\alpha, \beta} \sum_{c \in \mathbf{x}} \sum_{s \in D} \alpha_{cs} \cdot \mathbf{w}^T \Phi_1(\mathbf{x}, c, s) + \sum_{c, d \in \mathbf{x}} \sum_{s, t \in D} \beta_{cs, dt} \cdot \mathbf{w}^T \Phi_2(\mathbf{x}, c, s, d, t) \quad (1)$$

We restrict the possible assignments to the decision variables, forcing the resulting output formula to be *syntactically* legal, for example by restricting active  $\beta$ -variables to be type consistent, and forcing the resulting functional composition to be acyclic and fully connected (we refer the reader to (Clarke et al., 2010) for more details). We take advantage of the flexible ILP framework and encode these restrictions as global constraints.

**Features** We use two types of feature, first-order  $\Phi_1$  and second-order  $\Phi_2$ .  $\Phi_1$  depends on lexical information: each mapping of a lexical item  $c$  to a domain symbol  $s$  generates a feature. In addition each combination of a lexical item  $c$  and an symbol type generates a feature.

$\Phi_2$  captures a pair of symbols and their alignment to lexical items. Given a second-order decision  $\beta_{cs, dt}$ , a feature is generated considering the normalized distance between the head words in the constituents  $c$  and  $d$ . Another feature is generated for every composition of symbols (ignoring the alignment to the text).

## 2.2 Domain-Independent Information

We enhance the decision process with information that abstracts over the attributes of specific domains by adding an intermediate layer consisting of the predicate-argument structure of the sentence. Consider the mappings described in Example 1. Instead of relying on the mapping between *Pink goalie* and `pink1`, this model tries to identify an `ARG` using different means. For example, the fact that it is preceded by a determiner, or capitalized provide useful cues. We do not assume any language specific knowledge and use features that help capture these cues.

This information is used to assist the overall learning process. We assume that these labels correspond to a binding to some logical symbol, and encode it as a constraint forcing the relations between the two models. Moreover, since learning this layer is a by-product of the learning process (as it does not use any labeled data) forcing the connection between the decisions is the mechanism that drives learning this model.

Our domain-independent layer bears some similarity to other semantic tasks, most notably Semantic-Role Labeling (SRL) introduced in (Gildea and Jurafsky, 2002), in which identifying the predicate-argument structure is considered a preprocessing step, prior to assigning argument labels. Unlike SRL, which aims to identify linguistic structures alone, in our framework these structures capture both natural-language and domain-language considerations.

**Domain-Independent Decision Variables** We add two new types of decisions abstracting over the domain-specific decisions. We encode the new decisions as  $\gamma_c$  and  $\delta_{cd}$ . The first ( $\gamma$ ) captures local information helping to determine if a given constituent  $c$  is likely to have a label (i.e.,  $\gamma_c^P$  for predicate or  $\gamma_c^A$  for argument). The second ( $\delta$ ) considers higher level structures, quantifying decisions over both the labels of the constituents  $c, d$  as a predicate-argument pair. Note, a given word  $c$  can be labeled as PRED or ARG if  $\gamma_c$  and  $\delta_{cd}$  are active.

**Model’s Features** We use the following features: (1) **Local Decisions**  $\Phi_3(\gamma(c))$  use a feature indicating if  $c$  is capitalized, a set of features capturing the context of  $c$  (window of size 2), such as determiner and quantifier occurrences. Finally we use a set of features capturing the suffix letters of  $c$ , these features are useful in identifying verb patterns. Features indicate if  $c$  is mapped to an ARG or PRED. (2) **Global Decision**  $\Phi_4(\delta(c, d))$ : a feature indicating the relative location of  $c$  compared to  $d$  in the input sentence. Additional features indicate properties of the relative location, such as if the word appears initially or finally in the sentence.

**Combined Model** In order to consider both types of information we augment our decision model with the new variables, resulting in the following objective function (Eq. 2).

$$F_{\mathbf{w}}(\mathbf{x}) = \arg \max_{\alpha, \beta} \sum_{c \in \mathbf{x}} \sum_{s \in D} \alpha_{cs} \cdot \mathbf{w}_1^T \Phi_1(\mathbf{x}, c, s) + \sum_{c, d \in \mathbf{x}} \sum_{s, t \in D} \sum_{i, j} \beta_{cs^i, dt^j} \cdot \mathbf{w}_2^T \Phi_2(\mathbf{x}, c, s^i, d, t^j) + \sum_{c \in \mathbf{x}} \gamma_c \cdot \mathbf{w}_3^T \Phi_3(\mathbf{x}, c) + \sum_{c, d \in \mathbf{x}} \delta_{cd} \cdot \mathbf{w}_4^T \Phi_4(\mathbf{x}, c, d) \quad (2)$$

For notational convenience we decompose the weight vector  $\mathbf{w}$  into four parts,  $\mathbf{w}_1, \mathbf{w}_2$  for features of (first, second) order domain-dependent decisions, and similarly for the independent ones. In addition, we also add new constraints tying these new variables to semantic interpretation :  $\forall c \in x (\gamma_c \rightarrow \alpha_{c,s^1} \vee \alpha_{c,s^2} \vee \dots \vee \alpha_{c,s^n})$   
 $\forall c \in x, \forall d \in x (\delta_{c,d} \rightarrow \beta_{c,s^1, dt^1} \vee \beta_{c,s^2, dt^1} \vee \dots \vee \beta_{c,s^n, dt^n})$   
 (where  $n$  is the length of  $x$ ).

### 2.3 Learning the Combined Model

The supervision to the learning process is given via data consisting of pairs of sentences and (domain specific) semantic interpretation. Given that we have introduced additional variables that capture the more abstract predicate-argument structure of the text, we need to induce these as latent variables. Our decision model maps an input sentence  $x$ , into a logical output  $y$  and predicate-argument structure  $h$ . We are only supplied with training data pertaining to the input ( $x$ ) and output ( $y$ ). We use a variant of the latent structure perceptron to learn in these settings<sup>2</sup>.

## 3 Experimental Settings

**Situated Language** This dataset, introduced in (Bordes et al., 2010), describes situations in a simulated world. The dataset consists of triplets of the form - ( $x, u, y$ ), where  $x$  is a NL sentence describing a situation (e.g., “*He goes to the kitchen*”),  $u$  is a world state consisting of grounded relations (e.g., loc(John, Kitchen)) description, and  $y$  is a logical interpretation corresponding to  $x$ .

The original dataset was used for *concept tagging*, which does not include a compositional aspect. We automatically generated the full logical structure by mapping the constants to function arguments. We generated additional function symbols of the same relation, but of different arity when needed<sup>3</sup>. Our new dataset consists of 25 relation symbols (originally 15). In our experiments we used a set of 5000 of the training triplets.

**Robocup** The Robocup dataset, originally introduced in (Chen and Mooney, 2008), describes robotic soccer events. The dataset was collected for the purpose of constructing semantic parsers from ambiguous supervision and consists of both “noisy” and gold labeled data. The noisy dataset

<sup>2</sup>Details omitted, see (Chang et al., 2010) for more details.

<sup>3</sup>For example, a unary relation symbol for “*He plays*”, and a binary for “*He plays with a ball*”.



| System                         | Training Procedure   |
|--------------------------------|--|
| <b>DOM-INIT</b>                | $w_1$ : Noisy probabilistic model, described below.  |
| <b>PRED-ARG<sub>S</sub></b>    | <b>Only</b> $w_3, w_4$ Trained over the Situ. dataset.   |
| <b>COMBINED<sub>RL</sub></b>   | $w_1, w_2, w_3, w_4$ : learned from Robocup gold   |
| <b>COMBINED<sub>RI+S</sub></b> | $w_3, w_4$ : learned from the Situ. dataset, $w_1$ uses the DOM-INIT Robocup model.                      |
| <b>COMBINED<sub>RL+S</sub></b> | $w_3, w_4$ : Initially learned over the Situ. dataset, updated jointly with $w_1, w_2$ over Robocup gold |

Table 1: Evaluated System descriptions.

was constructed by temporally aligning a stream of soccer events occurring during a robotic soccer match with human commentary describing the game. This dataset consists of pairs  $(x, \{y_0, y_k\})$ ,  $x$  is a sentence and  $\{y_0, y_k\}$  is a set of events (logical formulas). One of these events is assumed to correspond to the comment, however this is not guaranteed. The gold labeled data consists of pairs  $(x, y)$ . The data was collected from four Robocup games. In our experiments we follow other works and use 4-fold cross validation, training over 3 games and testing over the remaining game. We evaluate the Accuracy of the parser over the test game data.<sup>4</sup> Due to space considerations, we refer the reader to (Chen and Mooney, 2008) for further details about this dataset.

**Semantic Interpretation Tasks** We consider two of the tasks described in (Chen and Mooney, 2008) (1) *Semantic Parsing* requires generating the correct logical form given an input sentence. (2) *Matching*, given a NL sentence and a set of several possible interpretation candidates, the system is required to identify the correct one. In all systems, the source for domain-independent information is the Situated domain, and the results are evaluated over the Robocup domain.

**Experimental Systems** We tested several variations, all solving Eq. 2, however different resources were used to obtain Eq. 2 parameters (see sec. 2.2). Tab. 1 describes the different variations. We used the noisy Robocup dataset to initialize **DOM-INIT**, a noisy probabilistic model, constructed by taking statistics over the noisy robocup data and computing  $p(y|x)$ . Given the training set  $\{(x, \{y_1, \dots, y_k\})\}$ , every word in  $x$  is aligned to every symbol in every  $y$  that is aligned with it. The probability of a matching  $(x, y)$  is computed as the product:  $\prod_{i=1}^n p(y_i|x_i)$ , where  $n$  is the number of symbols appearing in  $y$ , and  $x_i, y_i$  is the word

<sup>4</sup>In our model accuracy is equivalent to F-measure.

| System                     | Matching     | Parsing      |
|----------------------------|--------------|--------------|
| PRED-ARG <sub>S</sub>      | 0.692        | –            |
| DOM-INIT                   | 0.823        | 0.357        |
| COMBINED <sub>RI+S</sub>   | <b>0.905</b> | <b>0.627</b> |
| (BÖRSCHINGER ET AL., 2011) | –            | 0.86         |
| (KIM AND MOONEY, 2010)     | 0.885        | 0.742        |

Table 2: Results for the matching and parsing tasks. Our system performs well on the matching task **without any domain information**. Results for both parsing and matching tasks show that using domain-independent information improves results dramatically.

level matching to a logical symbol. *Note that this model uses lexical information only.*

## 4 Knowledge Transfer Experiments

We begin by studying the role of domain-independent information when very little domain information is available. Domain-independent information is learned from the situated domain and domain-specific information (Robocup) available is the simple probabilistic model (**DOM-INIT**). This model can be considered as a noisy probabilistic lexicon, without any domain-specific compositional information, *which is only available through domain-independent information*.

The results, summarized in Table 2, show that in both tasks domain-independent information is extremely useful and can make up for missing domain information. Most notably, performance for the matching task using only domain independent information (**PRED-ARG<sub>S</sub>**) was surprisingly good, with an accuracy of 0.69. Adding domain-specific lexical information (**COMBINED<sub>RI+S</sub>**) pushes this result to over 0.9, currently the highest for this task – achieved without domain specific learning.

The second set of experiments study whether using domain independent information, when relevant (gold) domain-specific training data is available, improves learning. In this scenario, the domain-independent model is updated according to training data available for the Robocup domain. We compare two system over varying amounts of training data (25, 50, 200 training samples and the full set of 3 Robocup games), one bootstrapped using the Situ. domain (**COMBINED<sub>RL+S</sub>**) and one relying on the Robocup training data alone (**COMBINED<sub>RL</sub>**). The results, summarized in table 3, consistently show that transferring domain independent information is helpful, and helps push the learned models beyond the supervision offered by the relevant domain training data. Our final system, trained over the entire dataset achieves a

| System   | # training | Parsing            |
|--|------------|--------------------|
| COMBINED <sub>RL+S</sub> (COMBINED <sub>RL</sub> ) | 25         | 0.16 (0.03)        |
| COMBINED <sub>RL+S</sub> (COMBINED <sub>RL</sub> ) | 50         | 0.323 (0.16)       |
| COMBINED <sub>RL+S</sub> (COMBINED <sub>RL</sub> ) | 200        | 0.385 (0.36)       |
| COMBINED <sub>RL+S</sub> (COMBINED <sub>RL</sub> ) | full game  | <b>0.86</b> (0.79) |
| (CHEN ET AL., 2010)                                | full game  | 0.81               |

**Table 3:** Evaluating our model in a learning settings. The domain-independent information is used to bootstrap learning from the Robocup domain. Results show that this information improves performance significantly, especially when little data is available

score of 0.86, significantly outperforming (Chen et al., 2010), a competing supervised model. It achieves similar results to (Börschinger et al., 2011), the current state-of-the-art for the parsing task over this dataset. The system used in (Börschinger et al., 2011) learns from ambiguous training data and achieves this score by using global information. We hypothesize that it can be used by our model and leave it for future work.

## 5 Conclusions

In this paper, we took a first step towards a new kind of generalization in semantic parsing: constructing a model that is able to generalize to a new domain defined over a different set of symbols. Our approach adds an additional hidden layer to the semantic interpretation process, capturing shallow but domain-independent semantic information, which can be shared by different domains. Our experiments consistently show that domain-independent knowledge can be transferred between domains. We describe two settings; in the first, where only noisy lexical-level domain-specific information is available, we observe that the model learned in the other domain can be used to make up for the missing compositional information. For example, in the matching task, even when no domain information is available, identifying the abstract predicate argument structure provides sufficient discriminatory power to identify the correct event in over 69% of the times.

In the second setting domain-specific examples are available. The learning process can still utilize the transferred knowledge, as it provides scaffolding for the latent learning process, resulting in a significant improvement in performance.

## 6 Acknowledgement

The authors would like to thank Julia Hockenmaier, Gerald DeJong, Raymond Mooney and the anonymous reviewers for their efforts and insightful comments.

Most of this work was done while the first author was at the University of Illinois. The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. In addition, this material is based on research sponsored by DARPA under agreement number FA8750-13-2-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, or the U.S. Government.

## References

- A. Bordes, N. Usunier, R. Collobert, and J. Weston. 2010. Towards understanding situated natural language. In *AISTATS*.
- B. Börschinger, B. K. Jones, and M. Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *EMNLP*.
- M. Chang, D. Goldwasser, D. Roth, and V. Srikumar. 2010. Discriminative learning over constrained latent representations. In *NAACL*.
- D. Chen and R. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *ICML*.
- D. L. Chen, J. Kim, and R. J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.
- J. Clarke, D. Goldwasser, M. Chang, and D. Roth. 2010. Driving semantic parsing from the world’s response. In *CoNLL*.
- D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*.
- D. Goldwasser, R. Reichart, J. Clarke, and D. Roth. 2011. Confidence driven unsupervised semantic parsing. In *ACL*.
- J. Kim and R. J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *COLING*.
- T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, , and M. Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *EMNLP*.
- Y.W. Wong and R. Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*.
- L. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*.

# A Structured Distributional Semantic Model for Event Co-reference

Kartik Goyal\*    Sujay Kumar Jauhar\*    Huiying Li\*  
Mrinmaya Sachan\*    Shashank Srivastava\*    Eduard Hovy

Language Technologies Institute  
School of Computer Science  
Carnegie Mellon University

{kartikgo, sjauhar, huiyingl, mrinmays, shashans, hovy}@cs.cmu.edu

## Abstract

In this paper we present a novel approach to modelling distributional semantics that represents meaning as distributions over relations in syntactic neighborhoods. We argue that our model approximates meaning in compositional configurations more effectively than standard distributional vectors or bag-of-words models. We test our hypothesis on the problem of judging event coreferentiality, which involves compositional interactions in the predicate-argument structure of sentences, and demonstrate that our model outperforms both state-of-the-art window-based word embeddings as well as simple approaches to compositional semantics previously employed in the literature.

## 1 Introduction

Distributional Semantic Models (DSM) are popular in computational semantics. DSMs are based on the hypothesis that the meaning of a word or phrase can be effectively captured by the distribution of words in its neighborhood. They have been successfully used in a variety of NLP tasks including information retrieval (Manning et al., 2008), question answering (Tellex et al., 2003), word-sense discrimination (Schütze, 1998) and disambiguation (McCarthy et al., 2004), semantic similarity computation (Wong and Raghavan, 1984; McCarthy and Carroll, 2003) and selectional preference modeling (Erk, 2007).

A shortcoming of DSMs is that they ignore the syntax within the context, thereby reducing the distribution to a bag of words. Composing the

distributions for “Lincoln”, “Booth”, and “killed” gives the same result regardless of whether the input is “Booth killed Lincoln” or “Lincoln killed Booth”. But as suggested by Pantel and Lin (2000) and others, modeling the distribution over preferential attachments for each syntactic relation separately yields greater expressive power. Thus, to remedy the bag-of-words failing, we extend the generic DSM model to several relation-specific distributions over syntactic neighborhoods. In other words, one can think of the Structured DSM (SDSM) representation of a word/phrase as several vectors defined over the same vocabulary, each vector representing the word’s selectional preferences for its various syntactic arguments.

We argue that this representation not only captures individual word semantics more effectively than the standard DSM, but is also better able to express the semantics of compositional units. We prove this on the task of judging event coreference.

Experimental results indicate that our model achieves greater predictive accuracy on the task than models that employ weaker forms of composition, as well as a baseline that relies on state-of-the-art window based word embeddings. This suggests that our formalism holds the potential of greater expressive power in problems that involve underlying semantic compositionality.

## 2 Related Work

Next, we relate and contrast our work to prior research in the fields of Distributional Vector Space Models, Semantic Compositionality and Event Co-reference Resolution.

### 2.1 DSMs and Compositionality

The underlying idea that “a word is characterized by the company it keeps” was expressed by Firth

---

\*Equally contributing authors

(1957). Several works have defined approaches to modelling context-word distributions anchored on a target word, topic, or sentence position. Collectively these approaches are called Distributional Semantic Models (DSMs).

While DSMs have been very successful on a variety of tasks, they are not an effective model of semantics as they lack properties such as compositionality or the ability to handle operators such as negation. In order to model a stronger form of semantics, there has been a recent surge in studies that phrase the problem of DSM compositionality as one of vector composition. These techniques derive the meaning of the combination of two words  $a$  and  $b$  by a single vector  $c = f(a, b)$ . Mitchell and Lapata (2008) propose a framework to define the composition  $c = f(a, b, r, K)$  where  $r$  is the relation between  $a$  and  $b$ , and  $K$  is the additional knowledge used to define composition. While this framework is quite general, the actual models considered in the literature tend to disregard  $K$  and  $r$  and mostly perform component-wise addition and multiplication, with slight variations, of the two vectors. To the best of our knowledge the formulation of composition we propose is the first to account for both  $K$  and  $r$  within this compositional framework.

Dinu and Lapata (2010) and Séaghdha and Korhonen (2011) introduced a probabilistic model to represent word meanings by a latent variable model. Subsequently, other high-dimensional extensions by Rudolph and Giesbrecht (2010), Baroni and Zamparelli (2010) and Grefenstette et al. (2011), regression models by Guevara (2010), and recursive neural network based solutions by Socher et al. (2012) and Collobert et al. (2011) have been proposed. However, these models do not efficiently account for structure.

Pantel and Lin (2000) and Erk and Padó (2008) attempt to include syntactic context in distributional models. A quasi-compositional approach was attempted in Thater et al. (2010) by a combination of first and second order context vectors. But they do not explicitly construct phrase-level meaning from words which limits their applicability to real world problems. Furthermore, we also include structure into our method of composition. Prior work in structure aware methods to the best of our knowledge are (Weisman et al., 2012) and (Baroni and Lenci, 2010). However, these methods do not explicitly model composition.

## 2.2 Event Co-reference Resolution

While automated resolution of entity coreference has been an actively researched area (Haghighi and Klein, 2009; Stoyanov et al., 2009; Raghunathan et al., 2010), there has been relatively little work on event coreference resolution. Lee et al. (2012) perform joint cross-document entity and event coreference resolution using the two-way feedback between events and their arguments. We, on the other hand, attempt a slightly different problem of making co-referentiality judgements on event-coreference candidate pairs.

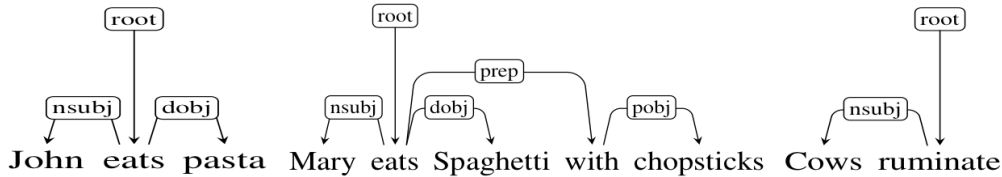
## 3 Structured Distributional Semantics

In this paper, we propose an approach to incorporate structure into distributional semantics (more details in Goyal et al. (2013)). The word distributions drawn from the context defined by a set of relations anchored on the target word (or phrase) form a set of vectors, namely a matrix for the target word. One axis of the matrix runs over all the relations and the other axis is over the distributional word vocabulary. The cells store word counts (or PMI scores, or other measures of word association). Note that collapsing the rows of the matrix provides the standard dependency based distributional representation.

### 3.1 Building Representation: The PropStore

To build a lexicon of SDSM matrices for a given vocabulary we first construct a proposition knowledge base (the PropStore) created by parsing the Simple English Wikipedia. Dependency arcs are stored as 3-tuples of the form  $\langle w_1, r, w_2 \rangle$ , denoting an occurrence of words  $w_1$ , word  $w_2$  related by  $r$ . We also store sentence indices for triples as this allows us to achieve an intuitive technique to achieve compositionality. In addition to the words' surface-forms, the PropStore also stores their POS tags, lemmas, and Wordnet supersenses. This helps to generalize our representation when surface-form distributions are sparse.

The PropStore can be used to query for the expectations of words, supersenses, relations, etc., around a given word. In the example in Figure 1, the query  $(SST(W_1) = \text{verb.consumption}, ?, \text{dobj})$  i.e. "what is consumed" might return expectations [pasta:1, spaghetti:1, mice:1 ...]. Relations and POS tags are obtained using a dependency parser Tratz and Hovy (2011), supersense tags using sst-light Ciaramita and Altun (2006), and lemmas us-



- 1) { (John/NNP/john/Noun.person , nsubj, eats/VBG/eat/verb.consumption ),  
 (eats/VBG/eat/verb.consumption, dobj, pasta/NN/pasta/noun.food) }
- 2) { (Mary/NNP/mary/Noun.person), nsubj, (eats/VBG/eat/verb.consumption) ... }
- 3) { (Cows/NNP/cow/Noun.animal),nsubj,(ruminates/VBG/ruminates/verb.consumption) }

Figure 1: Sample sentences & triples

ing Wordnet Fellbaum (1998).

### 3.2 Mimicking Compositionality

For representing intermediate multi-word phrases, we extend the above word-relation matrix symbolism in a bottom-up fashion using the PropStore. The combination hinges on the intuition that when lexical units combine to form a larger syntactically connected phrase, the representation of the phrase is given by its own distributional neighborhood within the embedded parse tree. The distributional neighborhood of the net phrase can be computed using the PropStore given syntactic relations anchored on its parts. For the example in Figure 1, we can compose  $SST(w_1) = \text{Noun.person}$  and  $\text{Lemma}(W_1) = \text{eat}$  appearing together with a *nsubj* relation to obtain expectations around “people eat” yielding [pasta:1, spaghetti:1 ...] for the *object* relation, [room:2, restaurant:1 ...] for the *location* relation, etc. Larger phrasal queries can be built to answer queries like “What do people in China eat with?”, “What do cows do?”, etc. All of this helps us to account for both relation  $r$  and knowledge  $K$  obtained from the PropStore within the compositional framework  $c = f(a, b, r, K)$ .

The general outline to obtain a composition of two words is given in Algorithm 1, which returns the distributional expectation around the composed unit. Note that the entire algorithm can conveniently be written in the form of database queries to our PropStore.

---

#### Algorithm 1 ComposePair( $w_1, r, w_2$ )

---

- $M_1 \leftarrow \text{queryMatrix}(w_1)$  (1)
  - $M_2 \leftarrow \text{queryMatrix}(w_2)$  (2)
  - SentIDs  $\leftarrow M_1(r) \cap M_2(r)$  (3)
  - return  $((M_1 \cap \text{SentIDs}) \cup (M_2 \cap \text{SentIDs}))$  (4)
- 

For the example “noun.person nsubj eat”, steps

(1) and (2) involve querying the PropStore for the individual tokens, noun.person and eat. Let the resulting matrices be  $M_1$  and  $M_2$ , respectively. In step (3), SentIDs (sentences where the two words appear with the specified relation) are obtained by taking the intersection between the *nsubj* component vectors of the two matrices  $M_1$  and  $M_2$ . In step (4), the entries of the original matrices  $M_1$  and  $M_2$  are intersected with this list of common SentIDs. Finally, the resulting matrix for the composition of the two words is simply the union of all the relationwise intersected sentence IDs. Intuitively, through this procedure, we have computed the expectation around the words  $w_1$  and  $w_2$  when they are connected by the relation “r”.

Similar to the two-word composition process, given a parse subtree  $T$  of a phrase, we obtain its matrix representation of empirical counts over word-relation contexts (described in Algorithm 2). Let the  $E = \{e_1 \dots e_n\}$  be the set of edges in  $T$ ,  $e_i = (w_{i1}, r_i, w_{i2}) \forall i = 1 \dots n$ .

---

#### Algorithm 2 ComposePhrase( $T$ )

---

- SentIDs  $\leftarrow$  All Sentences in corpus
  - for**  $i = 1 \rightarrow n$  **do**
  - $M_{i1} \leftarrow \text{queryMatrix}(w_{i1})$
  - $M_{i2} \leftarrow \text{queryMatrix}(w_{i2})$
  - SentIDs  $\leftarrow$  SentIDs  $\cap (M_{i1}(r_i) \cap M_{i2}(r_i))$
  - end for**
  - return  $((M_{11} \cap \text{SentIDs}) \cup (M_{12} \cap \text{SentIDs})$   
 $\dots \cup (M_{n1} \cap \text{SentIDs}) \cup (M_{n2} \cap \text{SentIDs}))$
- 

The phrase representations becomes sparser as phrase length increases. For this study, we restrict phrasal query length to a maximum of three words.

### 3.3 Event Coreferentiality

Given the SDSM formulation and assuming no sparsity constraints, it is possible to calculate

SDSM matrices for composed concepts. However, are these correct? Intuitively, if they truly capture semantics, the two SDSM matrix representations for “Booth assassinated Lincoln” and “Booth shot Lincoln with a gun” should be (almost) the same. To test this hypothesis we turn to the task of predicting whether two event mentions are coreferent or not, even if their surface forms differ. It may be noted that this task is different from the task of full event coreference and hence is not directly comparable to previous experimental results in the literature. Two mentions generally refer to the same event when their respective actions, agents, patients, locations, and times are (almost) the same. Given the non-compositional nature of determining equality of locations and times, we represent each event mention by a triple  $\mathbf{E} = (e, a, p)$  for the event, agent, and patient.

In our corpus, most event mentions are verbs. However, when nominalized events are encountered, we replace them by their verbal forms. We use SRL Collobert et al. (2011) to determine the agent and patient arguments of an event mention. When SRL fails to determine either role, its empirical substitutes are obtained by querying the PropStore for the most likely word expectations for the role. It may be noted that the SDSM representation relies on syntactic dependency relations. Hence, to bridge the gap between these relations and the composition of semantic role participants of event mentions we empirically determine those syntactic relations which most strongly co-occur with the semantic relations connecting events, agents and patients. The triple  $(e, a, p)$  is thus the composition of the triples  $(a, relationset_{agent}, e)$  and  $(p, relationset_{patient}, e)$ , and hence a complex object. To determine equality of this complex composed representation we generate three levels of progressively simplified event constituents for comparison:

**Level 1: Full Composition:**

$$M_{full} = ComposePhrase(e, a, p).$$

**Level 2: Partial Composition:**

$$M_{part:EA} = ComposePair(e, r, a)$$

$$M_{part:EP} = ComposePair(e, r, p).$$

**Level 3: No Composition:**

$$M_E = queryMatrix(e)$$

$$M_A = queryMatrix(a)$$

$$M_P = queryMatrix(p)$$

To judge coreference between events  $\mathbf{E1}$  and  $\mathbf{E2}$ , we compute pair-

wise similarities  $\text{Sim}(M1_{full}, M2_{full})$ ,  $\text{Sim}(M1_{part:EA}, M2_{part:EA})$ , etc., for each level of the composed triple representation. Furthermore, we vary the computation of similarity by considering different levels of granularity (lemma, SST), various choices of distance metric (Euclidean, Cityblock, Cosine), and score normalization techniques (Row-wise, Full, Column-collapsed). This results in 159 similarity-based features for every pair of events, which are used to train a classifier to decide conference.

## 4 Experiments

We evaluate our method on two datasets and compare it against four baselines, two of which use window based distributional vectors and two that employ weaker forms of composition.

### 4.1 Datasets

**IC Event Coreference Corpus:** The dataset (Hovy et al., 2013), drawn from 100 news articles about violent events, contains manually created annotations for 2214 pairs of co-referent and non-coreferent events each. Where available, events’ semantic role-fillers for *agent* and *patient* are annotated as well. When missing, empirical substitutes were obtained by querying the PropStore for the preferred word attachments.

**EventCorefBank (ECB) corpus:** This corpus (Bejan and Harabagiu, 2010) of 482 documents from Google News is clustered into 45 topics, with event coreference chains annotated over each topic. The event mentions are enriched with semantic roles to obtain the canonical event structure described above. Positive instances are obtained by taking pairwise event mentions within each chain, and negative instances are generated from pairwise event mentions across chains, but within the same topic. This results in 11039 positive instances and 33459 negative instances.

### 4.2 Baselines

To establish the efficacy of our model, we compare SDSM against a purely window-based baseline (DSM) trained on the same corpus. In our experiments we set a window size of seven words. We also compare SDSM against the window-based embeddings trained using a recursive neural network (SENNA) (Collobert et al., 2011) on both datasets. SENNA embeddings are state-of-the-art for many NLP tasks. The second baseline uses

|       | IC Corpus    |              |              |              | ECB Corpus   |              |              |              |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|       | Prec         | Rec          | F-1          | Acc          | Prec         | Rec          | F-1          | Acc          |
| SDSM  | <b>0.916</b> | 0.929        | <b>0.922</b> | <b>0.906</b> | 0.901        | 0.401        | <b>0.564</b> | <b>0.843</b> |
| Senna | 0.850        | 0.881        | 0.865        | 0.835        | 0.616        | <b>0.408</b> | 0.505        | 0.791        |
| DSM   | 0.743        | 0.843        | 0.790        | 0.740        | 0.854        | 0.378        | 0.524        | 0.830        |
| MVC   | 0.756        | <b>0.961</b> | 0.846        | 0.787        | <b>0.914</b> | 0.353        | 0.510        | 0.831        |
| AVC   | 0.753        | 0.941        | 0.837        | 0.777        | 0.901        | 0.373        | 0.528        | 0.834        |

Table 1: Cross-validation Performance on IC and ECB dataset

SENNA to generate level 3 similarity features for events’ individual words (agent, patient and action). As our final set of baselines, we extend two simple techniques proposed by (Mitchell and Lapata, 2008) that use element-wise addition and multiplication operators to perform composition. We extend it to our matrix representation and build two baselines AVC (element-wise addition) and MVC (element-wise multiplication).

### 4.3 Discussion

Among common classifiers, decision-trees (J48) yielded best results in our experiments. Table 1 summarizes our results on both datasets.

The results reveal that the SDSM model consistently outperforms DSM, SENNA embeddings, and the MVC and AVC models, both in terms of F-1 score and accuracy. The IC corpus comprises of domain specific texts, resulting in high lexical overlap between event mentions. Hence, the scores on the IC corpus are consistently higher than those on the ECB corpus.

The improvements over DSM and SENNA embeddings, support our hypothesis that syntax lends greater expressive power to distributional semantics in compositional configurations. Furthermore, the increase in predictive accuracy over MVC and AVC shows that our formulation of composition of two words based on the relation binding them yields a stronger form of compositionality than simple additive and multiplicative models.

Next, we perform an ablation study to determine the most predictive features for the task of event coreferentiality. The forward selection procedure reveals that the most informative attributes are the level 2 compositional features involving the agent and the action, as well as their individual level 3 features. This corresponds to the intuition that the agent and the action are the principal determiners for identifying events. Features involving the patient and level 1 features are least

useful. This is probably because features involving full composition are sparse, and not as likely to provide statistically significant evidence. This may change as our PropStore grows in size.

## 5 Conclusion and Future Work

We outlined an approach that introduces structure into distributed semantic representations gives us an ability to compare the identity of two representations derived from supposedly semantically identical phrases with different surface realizations. We employed the task of event coreference to validate our representation and achieved significantly higher predictive accuracy than several baselines.

In the future, we would like to extend our model to other semantic tasks such as paraphrase detection, lexical substitution and recognizing textual entailment. We would also like to replace our syntactic relations to semantic relations and explore various ways of dimensionality reduction to solve this problem.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by the following grants: NSF grant IIS-1143703, NSF award IIS-1147810, DARPA grant FA87501220342.

## References

- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721, December.
- Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical*

- Methods in Natural Language Processing*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1412–1422, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 594–602, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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.
- Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1162–1172, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Katrin Erk. 2007. A simple, similarity-based model for selectional preferences.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- John R. Firth. 1957. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32.
- Kartik Goyal, Sujay Kumar Jauhar, Mrinmaya Sachan, Shashank Srivastava, Huiying Li, and Eduard Hovy. 2013. A structured distributional semantic model : Integrating structure with semantics. In *Proceedings of the 1st Continuous Vector Space Models and their Compositionality Workshop at the conference of ACL 2013*.
- Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 125–134, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '10, pages 33–37, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1152–1161, Stroudsburg, PA, USA. Association for Computational Linguistics.
- E.H. Hovy, T. Mitamura, M.F. Verdejo, J. Araki, and A. Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. In *Proceedings of the 1st Events Workshop at the conference of the HLT-NAACL 2013*.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- Diana McCarthy and John Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Comput. Linguist.*, 29(4):639–654, December.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.
- Patrick Pantel and Dekang Lin. 2000. Word-for-word glossing with contextually similar words. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 78–85, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10,



- pages 492–501, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sebastian Rudolph and Eugenie Giesbrecht. 2010. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 907–916, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Hinrich Schütze. 1998. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123.
- Diarmuid Ó Séaghdha and Anna Korhonen. 2011. Probabilistic models of similarity in syntactic context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1047–1057, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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 *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 656–664, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Stefanie Tellex, Boris Katz, Jimmy J. Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR*, pages 41–47.
- Stefan Thater, Hagen Fürstenu, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 948–957, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1257–1268, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Hila Weisman, Jonathan Berant, Idan Szpektor, and Ido Dagan. 2012. Learning verb inference rules from linguistically-motivated evidence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 194–204, Stroudsburg, PA, USA. Association for Computational Linguistics.
- S. K. M. Wong and Vijay V. Raghavan. 1984. Vector space model of information retrieval: a reevaluation. In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '84, pages 167–185, Swinton, UK. British Computer Society.

# Text Classification from Positive and Unlabeled Data using Misclassified Data Correction

Fumiyo Fukumoto and Yoshimi Suzuki and Suguru Matsuyoshi

Interdisciplinary Graduate School of Medicine and Engineering

University of Yamanashi, Kofu, 400-8511, JAPAN

{fukumoto, ysuzuki, sugurum}@yamanashi.ac.jp

## Abstract

This paper addresses the problem of dealing with a collection of labeled training documents, especially annotating negative training documents and presents a method of text classification from positive and unlabeled data. We applied an error detection and correction technique to the results of positive and negative documents classified by the Support Vector Machines (SVM). The results using Reuters documents showed that the method was comparable to the current state-of-the-art biased-SVM method as the F-score obtained by our method was 0.627 and biased-SVM was 0.614.

## 1 Introduction

Text classification using machine learning (ML) techniques with a small number of labeled data has become more important with the rapid increase in volume of online documents. Quite a lot of learning techniques *e.g.*, semi-supervised learning, self-training, and active learning have been proposed. Blum *et al.* proposed a semi-supervised learning approach called the Graph Mincut algorithm which uses a small number of positive and negative examples and assigns values to unlabeled examples in a way that optimizes consistency in a nearest-neighbor sense (Blum *et al.*, 2001). Cabrera *et al.* described a method for self-training text categorization using the Web as the corpus (Cabrera *et al.*, 2009). The method extracts unlabeled documents automatically from the Web and applies an enriched self-training for constructing the classifier.

Several authors have attempted to improve classification accuracy using only positive and unlabeled data (Yu *et al.*, 2002; Ho *et al.*, 2011). Liu *et al.* proposed a method called biased-SVM that

uses soft-margin SVM as the underlying classifiers (Liu *et al.*, 2003). Elkan and Noto proposed a theoretically justified method (Elkan and Noto, 2008). They showed that under the assumption that the labeled documents are selected randomly from the positive documents, a classifier trained on positive and unlabeled documents predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive. They reported that the results were comparable to the current state-of-the-art biased SVM method. The methods of Liu *et al.* and Elkan *et al.* model a region containing most of the available positive data. However, these methods are sensitive to the parameter values, especially the small size of labeled data presents special difficulties in tuning the parameters to produce optimal results.

In this paper, we propose a method for eliminating the need for manually collecting training documents, especially annotating negative training documents based on supervised ML techniques. Our goal is to eliminate the need for manually collecting training documents, and hopefully achieve classification accuracy from positive and unlabeled data as high as that from labeled positive and labeled negative data. Like much previous work on semi-supervised ML, we apply SVM to the positive and unlabeled data, and add the classification results to the training data. The difference is that before adding the classification results, we applied the MisClassified data Detection and Correction (MCDC) technique to the results of SVM learning in order to improve classification accuracy obtained by the final classifiers.

## 2 Framework of the System

The MCDC method involves category error correction, *i.e.*, correction of misclassified candidates, while there are several strategies for automatically detecting lexical/syntactic errors in corpora (Abney *et al.*, 1999; Eskin, 2000; Dickinson and

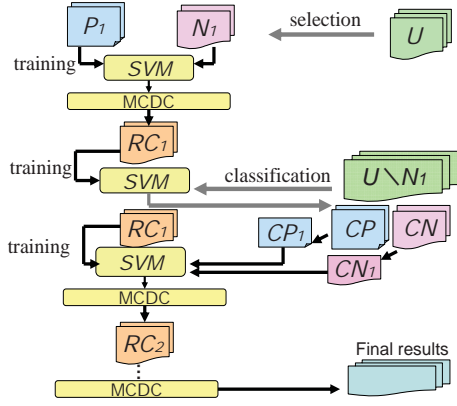


Figure 1: Overview of the system

Meurers., 2005; Boyd et al., 2008) or categorical data errors (Akoglu et al., 2013). The method first detects error candidates. As error candidates, we focus on support vectors (SVs) extracted from the training documents by SVM. Training by SVM is performed to find the optimal hyperplane consisting of SVs, and only the SVs affect the performance. Thus, if some training document reduces the overall performance of text classification because of an outlier, we can assume that the document is a SV.

Figure 1 illustrates our system. First, we randomly select documents from unlabeled data ( $U$ ) where the number of documents is equal to that of the initial positive training documents ( $P_1$ ). We set these selected documents to negative training documents ( $N_1$ ), and apply SVM to learn classifiers. Next, we apply the MCDC technique to the results of SVM learning. For the result of correction ( $RC_1$ )<sup>1</sup>, we train SVM classifiers, and classify the remaining unlabeled data ( $U \setminus N_1$ ). For the result of classification, we randomly select positive ( $CP_1$ ) and negative ( $CN_1$ ) documents classified by SVM and add to the SVM training data ( $RC_1$ ). We re-train SVM classifiers with the training documents, and apply the MCDC. The procedure is repeated until there are no unlabeled documents judged to be either positive or negative. Finally, the test data are classified using the final classifiers. In the following subsections, we present the MCDC procedure shown in Figure 2. It consists of three steps: extraction of misclassified candidates, estimation of error reduction, and correction of misclassified candidates.

<sup>1</sup>The manually annotated positive examples are not corrected.

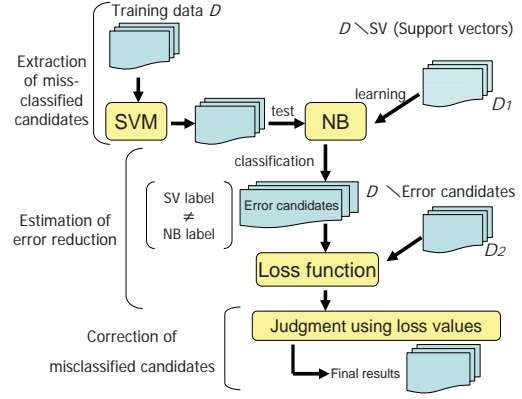


Figure 2: The MCDC procedure

## 2.1 Extraction of misclassified candidates

Let  $D$  be a set of training documents and  $\mathbf{x}_k \in \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  be a SV of negative or positive documents obtained by SVM. We remove  $\cup_{k=1}^m \mathbf{x}_k$  from the training documents  $D$ . The resulting  $D \setminus \cup_{k=1}^m \mathbf{x}_k$  is used for training Naive Bayes (NB) (McCallum, 2001), leading to a classification model. This classification model is tested on each  $\mathbf{x}_k$ , and assigns a positive or negative label. If the label is different from that assigned to  $\mathbf{x}_k$ , we declare  $\mathbf{x}_k$  an error candidate.

## 2.2 Estimation of error reduction

We detect misclassified data from the extracted candidates by estimating error reduction. The estimation of error reduction is often used in active learning. The earliest work is the method of Roy and McCallum (Roy and McCallum, 2001). They proposed a method that directly optimizes expected future error by log-loss or 0-1 loss, using the entropy of the posterior class distribution on a sample of unlabeled documents. We used their method to detect misclassified data. Specifically, we estimated future error rate by log-loss function. It uses the entropy of the posterior class distribution on a sample of the unlabeled documents. A loss function is defined by Eq (1).

$$E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}} = -\frac{1}{|X|} \sum_{x \in X} \sum_{y \in Y} P(y|x) \times \log(\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}(y|x)). \quad (1)$$

Eq (1) denotes the expected error of the learner.  $P(y | x)$  denotes the true distribution of output classes  $y \in Y$  given inputs  $x$ .  $X$  denotes a

set of test documents.  $\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}(y | x)$  shows the learner’s prediction, and  $D_2$  denotes the training documents  $D$  except for the error candidates  $\cup_{k=1}^l \mathbf{x}_k$ . If the value of Eq (1) is sufficiently small, the learner’s prediction is close to the true output distribution.

We used bagging to reduce variance of  $P(y | x)$  as it is unknown for each test document  $x$ . More precisely, from the training documents  $D$ , a different training set consisting of positive and negative documents is created<sup>2</sup>. The learner then creates a new classifier from the training documents. The procedure is repeated  $m$  times<sup>3</sup>, and the final class posterior for an instance is taken to be the unweighted average of the class posteriori for each of the classifiers.

### 2.3 Correction of misclassified candidates

For each error candidate  $\mathbf{x}_k$ , we calculated the expected error of the learner,  $E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_{k\_old})}}$  and  $E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_{k\_new})}}$  by using Eq (1). Here,  $y_{k\_old}$  refers to the original label assigned to  $\mathbf{x}_k$ , and  $y_{k\_new}$  is the resulting category label estimated by NB classifiers. If the value of the latter is smaller than that of the former, we declare the document  $x_k$  to be misclassified, *i.e.*, the label  $y_{k\_old}$  is an error, and its true label is  $y_{k\_new}$ . Otherwise, the label of  $\mathbf{x}_k$  is  $y_{k\_old}$ .

## 3 Experiments

### 3.1 Experimental setup

We chose the 1996 Reuters data (Reuters, 2000) for evaluation. After eliminating unlabeled documents, we divided these into three. The data (20,000 documents) extracted from 20 Aug to 19 Sept is used as training data indicating positive and unlabeled documents. We set the range of  $\delta$  from 0.1 to 0.9 to create a wide range of scenarios, where  $\delta$  refers to the ratio of documents from the positive class first selected from a fold as the positive set. The rest of the positive and negative documents are used as unlabeled data. We used categories assigned to more than 100 documents in the training data as it is necessary to examine a wide range of  $\delta$  values. These categories are 88 in all. The data from 20 Sept to 19 Nov is used

<sup>2</sup>We set the number of negative documents extracted randomly from the unlabeled documents to the same number of positive training documents.

<sup>3</sup>We set the number of  $m$  to 100 in the experiments.

as a test set  $X$ , to estimate true output distribution. The remaining data consisting 607,259 from 20 Nov 1996 to 19 Aug 1997 is used as a test data for text classification. We obtained a vocabulary of 320,935 unique words after eliminating words which occur only once, stemming by a part-of-speech tagger (Schmid, 1995), and stop word removal. The number of categories per documents is 3.21 on average. We used the SVM-Light package (Joachims, 1998)<sup>4</sup>. We used a linear kernel and set all parameters to their default values.

We compared our method, MCDC with three baselines: (1) SVM, (2) Positive Example-Based Learning (PEBL) proposed by (Yu et al., 2002), and (3) biased-SVM (Liu et al., 2003). We chose PEBL because the convergence procedure is very similar to our framework. Biased-SVM is the state-of-the-art SVM method, and often used for comparison (Elkan and Noto, 2008). To make comparisons fair, all methods were based on a linear kernel. We randomly selected 1,000 positive and 1,000 negative documents classified by SVM and added to the SVM training data in each iteration<sup>5</sup>. For biased-SVM, we used training data and classified test documents directly. We empirically selected values of two parameters, “ $c$ ” (trade-off between training error and margin) and “ $j$ ”, *i.e.*, cost (cost-factor, by which training errors on positive examples) that optimized the F-score obtained by classification of test documents.

The positive training data in SVM are assigned to the target category. The negative training data are the remaining data except for the documents that were assigned to the target category, *i.e.*, this is the ideal method as we used all the training data with positive/negative labeled documents. The number of positive training data in other three methods depends on the value of  $\delta$ , and the rest of the positive and negative documents were used as unlabeled data.

### 3.2 Text classification

Classification results for 88 categories are shown in Figure 3. Figure 3 shows micro-averaged F-score against the  $\delta$  value. As expected, the results obtained by SVM were the best among all  $\delta$  values. However, this is the ideal method that requires 20,000 documents labeled positive/negative, while other methods including our

<sup>4</sup><http://svmlight.joachims.org>

<sup>5</sup>We set the number of documents up to 1,000.

| Level (# of Cat)  |       | SVM   |             | PEBL  |                  | Biased-SVM |             | MCDC  |                 |
|-------------------|-------|-------|-------------|-------|------------------|------------|-------------|-------|-----------------|
|                   |       | Cat   | F           | Cat   | F (Iter)         | Cat        | F (Iter)    | Cat   | F (Iter)        |
| Top (22)          | Best  | GSPO  | .955        | GSPO  | .802 (26)        | CCAT       | .939        | GSPO  | .946 (9)        |
|                   | Worst | GODD  | .099        | GODD  | .079 (6)         | GODD       | .038        | GODD  | .104 (4)        |
|                   | Avg   |       | <b>.800</b> |       | <b>.475 (19)</b> |            | <b>.593</b> |       | <b>.619 (8)</b> |
| Second (32)       | Best  | M14   | .870        | E71   | .848 (7)         | M14        | .869        | M14   | .875 (9)        |
|                   | Worst | C16   | .297        | E14   | .161 (14)        | C16        | .148        | C16   | .150 (3)        |
|                   | Avg   |       | <b>.667</b> |       | <b>.383 (22)</b> |            | <b>.588</b> |       | <b>.593 (7)</b> |
| Third (33)        | Best  | M141  | .878        | C174  | .792 (27)        | M141       | .887        | M141  | .885 (8)        |
|                   | Worst | G152  | .102        | C331  | .179 (16)        | G155       | .130        | C331  | .142 (6)        |
|                   | Avg   |       | <b>.717</b> |       | <b>.313 (18)</b> |            | <b>.518</b> |       | <b>.557 (8)</b> |
| Fourth (1)        | –     | C1511 | <b>.738</b> | C1511 | <b>.481 (16)</b> | C1511      | <b>.737</b> | C1511 | <b>.719 (4)</b> |
| Micro Avg F-score |       |       | .718        |       | .428 (19)        |            | .614        |       | .627 (8)        |

Table 1: Classification performance ( $\delta = 0.7$ )

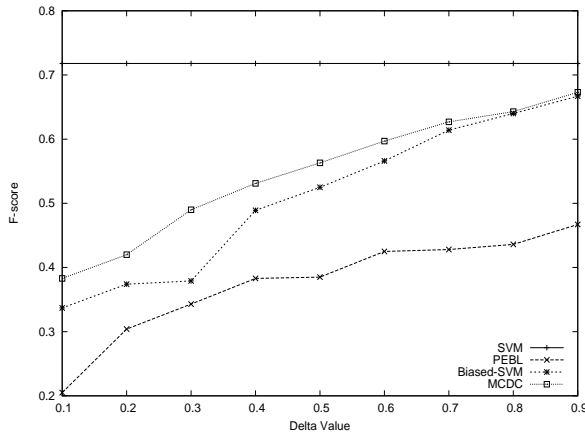


Figure 3: F-score against the value of  $\delta$

method used only positive and unlabeled documents. Overall performance obtained by MCDC was better for those obtained by PEBL and biased-SVM methods in all  $\delta$  values, especially when the positive set was small, *e.g.*,  $\delta = 0.3$ , the improvement of MCDC over biased-SVM and PEBL was significant.

Table 1 shows the results obtained by each method with a  $\delta$  value of 0.7. “Level” indicates each level of the hierarchy and the numbers in parentheses refer to the number of categories. “Best” and “Worst” refer to the best and the lowest F-scores in each level of a hierarchy, respectively. “Iter” in PEBL indicates the number of iterations until the number of negative documents is zero in the convergence procedure. Similarly, “Iter” in the MCDC indicates the number of iterations until no unlabeled documents are judged to be either positive or negative. As can be seen clearly from Table 1, the results with MCDC were better than those obtained by PEBL in each level of the hierarchy. Similarly, the results were bet-

| $\delta$ | SV      | Ec     | Err    | Correct |      |      |
|----------|---------|--------|--------|---------|------|------|
|          |         |        |        | Prec    | Rec  | F    |
| 0.3      | 227,547 | 54,943 | 79,329 | .693    | .649 | .670 |
| 0.7      | 141,087 | 34,944 | 42,385 | .712    | .673 | .692 |

Table 2: Miss-classified data correction results

ter than those of biased-SVM except for the fourth level, “C1511”(Annual results). The average numbers of iterations with MCDC and PEBL were 8 and 19 times, respectively. In biased-SVM, it is necessary to run SVM many times, as we searched “*c*” and “*j*”. In contrast, MCDC does not require such parameter tuning.

### 3.3 Correction of misclassified candidates

Our goal is to achieve classification accuracy from only positive documents and unlabeled data as high as that from labeled positive and negative data. We thus applied a miss-classified data detection and correction technique for the classification results obtained by SVM. Therefore, it is important to examine the accuracy of miss-classified correction. Table 2 shows detection and correction performance against all categories. “SV” shows the total number of SVs in 88 categories in all iterations. “Ec” refers to the total number of extracted error candidates. “Err” denotes the number of documents classified incorrectly by SVM and added to the training data, *i.e.*, the number of documents that should be assigned correctly by the correction procedure. “Prec” and “Rec” show the precision and recall of correction, respectively.

Table 2 shows that precision was better than recall with both  $\delta$  values, as the precision obtained by  $\gamma$  value = 0.3 and 0.7 were 4.4% and 3.9% improvement against recall values, respectively. These observations indicated that the error candidates extracted by our method were appropriately

corrected. In contrast, there were still other documents that were miss-classified but not extracted as error candidates. We extracted error candidates using the results of SVM and NB classifiers. Ensemble of other techniques such as boosting and kNN for further efficacy gains seems promising to try with our method.

#### 4 Conclusion

The research described in this paper involved text classification using positive and unlabeled data. Miss-classified data detection and correction technique was incorporated in the existing classification technique. The results using the 1996 Reuters corpora showed that the method was comparable to the current state-of-the-art biased-SVM method as the F-score obtained by our method was 0.627 and biased-SVM was 0.614. Future work will include feature reduction and investigation of other classification algorithms to obtain further advantages in efficiency and efficacy in manipulating real-world large corpora.

#### References

- S. Abney, R. E. Schapire, and Y. Singer. 1999. Boosting Applied to Tagging and PP Attachment. In *Proc. of the Joint SIGDAT Conference on EMNLP and Very Large Corpora*, pages 38–45.
- L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos. 2013. Fast and Reliable Anomaly Detection in Categorical Data. In *Proc. of the CIKM*, pages 415–424.
- A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy. 2001. Learning from Labeled and Unlabeled Data using Graph Mincuts. In *Proc. of the 18th ICML*, pages 19–26.
- A. Boyd, M. Dickinson, and D. Meurers. 2008. On Detecting Errors in Dependency Treebanks. *Research on Language and Computation*, 6(2):113–137.
- R. G. Cabrera, M. M. Gomez, P. Rosso, and L. V. Pineda. 2009. Using the Web as Corpus for Self-Training Text Categorization. *Information Retrieval*, 12(3):400–415.
- M. Dickinson and W. D. Meurers. 2005. Detecting Errors in Discontinuous Structural Annotation. In *Proc. of the ACL'05*, pages 322–329.
- C. Elkan and K. Noto. 2008. Learning Classifiers from Only Positive and Unlabeled Data. In *Proc. of the KDD'08*, pages 213–220.
- E. Eskin. 2000. Detecting Errors within a Corpus using Anomaly Detection. In *Proc. of the 6th ANLP Conference and the 1st Meeting of the NAACL*, pages 148–153.
- C. H. Ho, M. H. Tsai, and C. J. Lin. 2011. Active Learning and Experimental Design with SVMs. In *Proc. of the JMLR Workshop on Active Learning and Experimental Design*, pages 71–84.
- T. Joachims. 1998. SVM Light Support Vector Machine. In *Dept. of Computer Science Cornell University*.
- B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. 2003. Building Text Classifiers using Positive and Unlabeled Examples. In *Proc. of the ICDM'03*, pages 179–188.
- A. K. McCallum. 2001. Multi-label Text Classification with a Mixture Model Trained by EM. In *Revised Version of Paper Appearing in AAAI'99 Workshop on Text Learning*, pages 135–168.
- Reuters. 2000. *Reuters Corpus Volume1 English Language*. 1996-08-20 to 1997-08-19 Release Date 2000-11-03 Format Version 1.
- N. Roy and A. K. McCallum. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proc. of the 18th ICML*, pages 441–448.
- H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*, pages 47–50.
- H. Yu, H. Han, and K. C-C. Chang. 2002. PEBL: Positive Example based Learning for Web Page Classification using SVM. In *Proc. of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, pages 239–248.

# Character-to-Character Sentiment Analysis in Shakespeare's Plays

Eric T. Nalisnick   Henry S. Baird  
Dept. of Computer Science and Engineering  
Lehigh University  
Bethlehem, PA 18015, USA  
{etn212, hsb2}@lehigh.edu

## Abstract

We present an automatic method for analyzing sentiment dynamics between characters in plays. This literary format's structured dialogue allows us to make assumptions about who is participating in a conversation. Once we have an idea of who a character is speaking to, the sentiment in his or her speech can be attributed accordingly, allowing us to generate lists of a character's enemies and allies as well as pinpoint scenes critical to a character's emotional development. Results of experiments on Shakespeare's plays are presented along with discussion of how this work can be extended to unstructured texts (i.e. novels).

## 1 Introduction

Insightful analysis of literary fiction often challenges trained human readers let alone machines. In fact, some humanists believe literary analysis is so closely tied to the human condition that it is impossible for computers to perform. In his book *Reading Machines: Toward an Algorithmic Criticism*, Stephen Ramsay (2011) states:

Tools that can adjudicate the hermeneutical parameters of human reading experiences...stretch considerably beyond the most ambitious fantasies of artificial intelligence.

Antonio Roque (2012) has challenged Ramsay's claim, and certainly there has been successful work done in the computational analysis and modeling of narratives, as we will review in the next section. However, we believe that most previous work (except possibly (Elsner, 2012)) has failed to directly address the root cause of Ramsay's skepticism: can computers extract the emotions encoded in a narrative? For example, can the love

that Shakespeare's Juliet feels for Romeo be computationally tracked? Empathizing with characters along their journeys to emotional highs and lows is often what makes a narrative compelling for a reader, and therefore we believe mapping these journeys is the first step in capturing the human reading experience.

Unfortunately but unsurprisingly, computational modeling of the emotional relationships described in natural language text remains a daunting technical challenge. The reason this task is so difficult is that emotions are indistinct and often subtly conveyed, especially in text with literary merit. Humans typically achieve no greater than 80% accuracy in sentiment classification experiments involving product reviews (Pang et al., 2002) (Gammon, 2004). Similar experiments on fiction texts would presumably yield even higher error rates.

In order to attack this open problem and make further progress towards refuting Ramsay's claim, we turn to shallow statistical approaches. Sentiment analysis (Pang and Lee, 2008) has been successfully applied to mine social media data for emotional responses to events, public figures, and consumer products just by using emotion lexicons—lists that map words to polarity values (+1 for positive sentiment, -1 for negative) or valence values that try to capture degrees of polarity. In the following paper, we describe our attempts to use modern sentiment lexicons and dialogue structure to algorithmically track and model—with no domain-specific customization—the emotion dynamics between characters in Shakespeare's plays.<sup>1</sup>

## 2 Sentiment Analysis and Related Work

Sentiment analysis (SA) is now widely used commercially to infer user opinions from product reviews and social-media messages (Pang and Lee,

<sup>1</sup>XML versions provided by Jon Bosak: <http://www.ibiblio.org/xml/examples/shakespeare/>

2008). Traditional machine learning techniques on n-grams, parts of speech, and other bag of words features can be used when the data is labeled (e.g. IMDB’s user reviews are labeled with one to ten stars, which are assumed to correlate with the text’s polarity) (Pang et al., 2002). But text annotated with its true sentiments is hard to come by so often labels must be obtained via crowdsourcing.

Knowledge-based methods (which also typically rely on crowdsourcing) provide an alternative to using labeled data (Andreevskaia and Bergler, 2007). These methods are driven by sentiment lexicons, fixed lists associating words with “valences” (signed integers representing positive and negative feelings) (Kim and Hovy, 2004). Some lexicons allow for analysis of specific emotions by associating words with degrees of fear, joy, surprise, anger, anticipation, etc. (Strapparava and Valitutti, 2004) (Mohammad and Turney, 2008). Unsurprisingly, methods which, like these, lack deep understanding often work more reliably as the length of the input text increases.

Turning our attention now to automatic semantic analysis of fiction, it seems that narrative modeling and summarization has been the most intensively studied application. Chambers and Jurafsky (2009) described a system that can learn (without supervision) the sequence of events described in a narrative, and Elson and McKeown (2009) created a platform that can symbolically represent and reason over narratives.

Narrative structure has also been studied by representing character interactions as networks. Mutton (2004) adapted methods for extracting social networks from Internet Relay Chat (IRC) to mine Shakespeare’s plays for their networks. Extending this line of work to novels, Elson and McKeown (2010) developed a reliable method for speech attribution in unstructured texts, and then used this method to successfully extract social networks from Victorian novels (Elson et al., 2010)(Agarwal et al., 2012).

While structure is undeniably important, we believe analyzing a narrative’s emotions is essential to capturing the ‘reading experience,’ which is a view others have held. Alm and Sproat (2005) analyzed *Brothers Grimm* fairy tales for their ‘emotional trajectories,’ finding emotion typically increases as a story progresses. Mohammad (2011) scaled-up their work by using a crowdsourced emotion lexicon to track emotion dynam-

ics over the course of many novels and plays, including Shakespeare’s. In the most recent work we are aware of, Elsner (2012) analyzed emotional trajectories at the character level, showing how Miss Elizabeth Bennet’s emotions change over the course of *Pride and Prejudice*.

### 3 Character-to-Character Sentiment Analysis

| Character   | Hamlet's Sentiment Valence Sum |
|-------------|--------------------------------|
| Guildestern | 31                             |
| Polonius    | 25                             |
| Gertrude    | 24                             |
| Horatio     | 12                             |
| Ghost       | 8                              |
| Marcellus   | 7                              |
| Osric       | 7                              |
| Bernardo    | 2                              |
| Laertes     | -1                             |
| Ophelia     | -5                             |
| Rosencrantz | -12                            |
| Claudius    | -27                            |

Figure 1: The characters in *Hamlet* are ranked by Hamlet’s sentiment towards them. Expectedly, Claudius draws the most negative emotion.

We attempt to further Elsner’s line of work by leveraging text structure (as Mutton and Elson did) and knowledge-based SA to track the emotional trajectories of interpersonal relationships rather than of a whole text or an isolated character. To extract these relationships, we mined for character-to-character sentiment by summing the valence values (provided by the *AFINN* sentiment lexicon (Nielsen, 2011)) over each instance of continuous speech and then assumed that sentiment was directed towards the character that spoke immediately before the current speaker. This assumption doesn’t always hold; it is not uncommon to find a scene in which two characters are expressing feelings about someone offstage. Yet our initial results on Shakespeare’s plays show that the instances of face-to-face dialogue produce a strong enough signal to generate sentiment rankings that match our expectations.

For example, Hamlet’s sentiment rankings upon the conclusion of his play are shown in Figure 1. Not surprisingly, Claudius draws the most negative sentiment from Hamlet, receiving a score of -27. On the other hand, Gertrude is very well liked by Hamlet (+24), which is unexpected (at least to



us) since Hamlet suspects that his mother was involved in murdering King Hamlet.

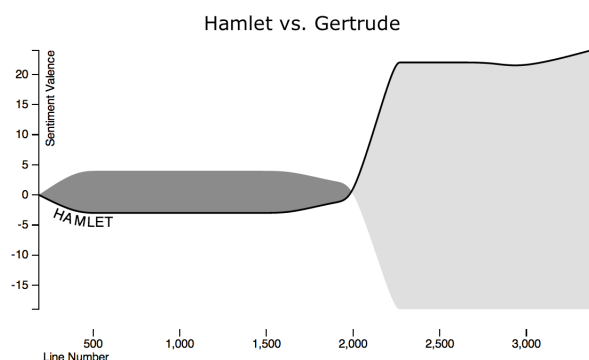


Figure 2: The above chart tracks how Gertrude’s and Hamlet’s sentiment towards one another changes over the course of the play. Hamlet’s sentiment for Gertrude is denoted by the black line, and Gertrude’s for Hamlet is marked by the opposite boundary of the dark/light gray area. The drastic change in *Act III Scene IV: The Queen’s Closet* is consistent with the scene’s plot events.

### 3.1 Peering into the Queen’s Closet

To gain more insight into this mother-son relationship, we examined how their feelings towards one another change over the course of the play. Figure 2 shows the results of dynamic character-to-character sentiment analysis on Gertrude and Hamlet. The running total of Hamlet’s sentiment valence toward Gertrude is tracked by the black line, and Gertrude’s feelings toward her son are tracked by the opposite boundary of the light/dark gray area. The line graph shows a dramatic swing in sentiment around line 2,250, which corresponds to Act iii, Scene iv.

In this scene, entitled *The Queen’s Closet*, Hamlet confronts his mother about her involvement in King Hamlet’s death. Gertrude is shocked at the accusation, revealing she never suspected Hamlet’s father was murdered. King Hamlet’s ghost even points this out to his son: “But, look, amazement on thy mother sits” (3.4.109). Hamlet then comes to the realization that his mother had no involvement in the murder and probably married Claudius more so to preserve stability in the state. As a result, Hamlet’s affection towards his mother grows, as exhibited in the sentiment jump from -1 to 22. But this scene has the opposite affect on Gertrude: she sees her son murder an innocent man (Polonius) and talk to an invisible presence

(she cannot see King Hamlet’s ghost). Gertrude is coming to the understanding that Hamlet is not just depressed but possibly mad and on a revenge mission. Because of Gertrude’s realization, it is only natural that her sentiment undergoes a sharply negative change (1 to -19).

### 3.2 Analyzing Shakespeare’s Most Famous Couples

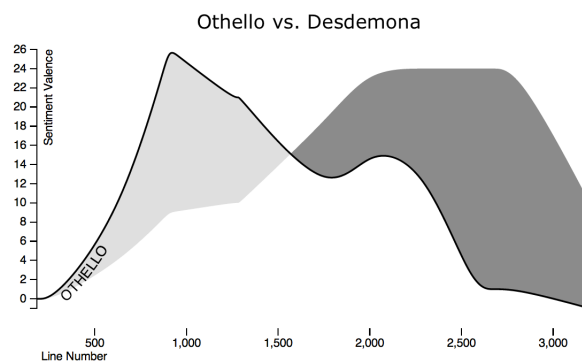


Figure 3: Othello’s sentiment for Desdemona is denoted by the black line, and Desdemona’s for Othello is marked by the opposite boundary of the dark/light gray area. As expected, the line graph shows Othello has very strong positive emotion towards his new wife at the beginning of the play, but this positivity quickly degrades as Othello falls deeper and deeper into Iago’s deceit.

After running this automatic analysis on all of Shakespeare’s plays, not all the results examined were as enlightening as the Hamlet vs. Gertrude example. Instead, the majority supported our already held interpretations. We will now present what the technique revealed about three of Shakespeare’s best known relationships. Figure 3 shows Othello vs. Desdemona sentiment dynamics. We clearly see Othello’s love for his new bride climaxes in the first third of the play and then rapidly degrades due to Iago’s deceit while Desdemona’s feelings for Othello stay positive until the very end of the play when it is clear Othello’s love for her has become poisoned. For an example of a contrasting relationship, Figure 4 shows Romeo vs. Juliet. As expected, the two exhibit rapidly increasing positive sentiment for each other that only tapers when the play takes a tragic course in the latter half. Lastly, Figure 5 shows Petruchio vs. Katharina (from *The Taming of the Shrew*). The phases of Petruchio’s courtship can be seen: first he is neutral to her, then ‘tames’ her with a

period of negative sentiment, and finally she embraces him, as shown by the increasingly positive sentiment exhibited in both directions.

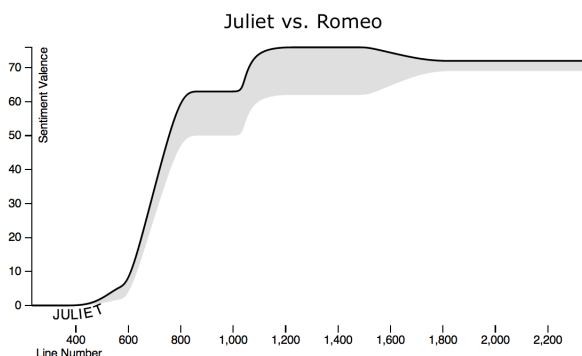


Figure 4: Juliet’s sentiment for Romeo is denoted by the black line, and Romeo’s for Juliet is marked by the opposite boundary of the gray area. Aligning with our expectations, both characters exhibit strong positive sentiment towards the other throughout the play.

Unfortunately, we do not have room in this paper to discuss further examples, but a visualization of sentiment dynamics between any pair of characters in any of Shakespeare’s plays can be seen at [www.lehigh.edu/~etn212/ShakespeareExplorer.html](http://www.lehigh.edu/~etn212/ShakespeareExplorer.html).

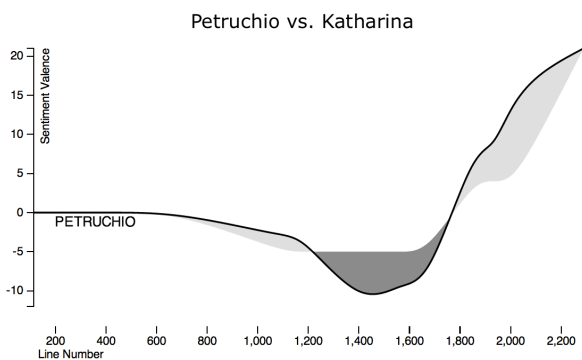


Figure 5: Petruchio’s sentiment for Katharina is denoted by the black line, and Katharina’s for Petruchio is marked by the opposite boundary of the dark/light gray area. The period from line 1200 to line 1700, during which Petruchio exhibits negative sentiment, marks where he is ‘taming’ the ‘shrew.’

#### 4 Future Work

While this paper presents experiments on just Shakespeare’s plays, note that the described technique can be extended to *any* work of fiction writ-

ten since the Elizabethan Period. The sentiment lexicon we used, *AFINN*, is designed for modern English; thus, it should only provide better analysis on works written after Shakespeare’s. Furthermore, character-to-character analysis should be able to be applied to novels (and other unstructured fiction) if Elson and McKeown’s (2010) speaker attribution technique is first run on the work.

Not only can these techniques be extended to novels but also be made more precise. For instance, the assumption that the current speaker’s sentiment is directed toward the previous speaker is rather naive. A speech could be analyzed for context clues that signal that the character speaking is not talking about someone present but about someone out of the scene. The sentiment could then be redirected to the not-present character. Furthermore, detecting subtle rhetorical features such as irony and deceit would markedly improve the accuracy of the analysis on some plays. For example, our character-to-character analysis fails to detect that Iago hates Othello because Iago gives his commander constant lip service in order to manipulate him—only revealing his true feelings at the play’s conclusion.

#### 5 Conclusions

As demonstrated, shallow, un-customized sentiment analysis can be used in conjunction with text structure to analyze interpersonal relationships described within a play and output an interpretation that matches reader expectations. This character-to-character sentiment analysis can be done statically as well as dynamically to possibly pinpoint influential moments in the narrative (which is how we noticed the importance of Hamlet’s *Act 3, Scene 4* to the Hamlet-Gertrude relationship). Yet, we believe the most noteworthy aspect of this work lies not in the details of our technique but rather in the demonstration that detailed emotion dynamics can be extracted with simplistic approaches—which in turn gives promise to the future work of robust analysis of interpersonal relationships in short stories and novels.

#### References

A. Agarwal, A. Corvalan, J. Jensen, and O. Rambow. 2012. Social network analysis of alice in wonderland. *NAACL-HLT 2012*, page 88.

- Cecilia Ovesdotter Alm and Richard Sproat. 2005. Emotional sequencing and development in fairy tales. In *Affective Computing and Intelligent Interaction*, pages 668–674. Springer.
- Alina Andreevskaia and Sabine Bergler. 2007. Clac and clac-nb: knowledge-based and corpus-based approaches to sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07*, pages 117–120, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics.
- Micha Elsner. 2012. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL '12*, pages 634–644, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David K Elson and Kathleen R McKeown. 2009. Extending and evaluating a platform for story understanding. In *Proceedings of the AAAI 2009 Spring Symposium on Intelligent Narrative Technologies II*.
- D.K. Elson and K.R. McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of AAAI*.
- D.K. Elson, N. Dames, and K.R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147. Association for Computational Linguistics.
- Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Soo-Min Kim and Eduard 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.
- Saif M Mohammad and Peter D Turney. 2008. Crowdsourcing the creation of a word–emotion association lexicon.
- S. Mohammad. 2011. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 105–114. Association for Computational Linguistics.
- P. Mutton. 2004. Inferring and visualizing social networks on internet relay chat. In *Information Visualization, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 35–43. IEEE.
- F. Å. Nielsen. 2011. AFINN, March.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- 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, EMNLP '02*, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Stephen Ramsay. 2011. *Reading Machines: Toward an Algorithmic Criticism*. University of Illinois Press.
- Antonio Roque. 2012. Towards a computational approach to literary text analysis. *NAACL-HLT 2012*, page 97.
- C. Strapparava and A. Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of LREC*, volume 4, pages 1083–1086.

# A Novel Text Classifier Based on Quantum Computation

Ding Liu, Xiaofang Yang, Minghu Jiang

Laboratory of Computational Linguistics, School of Humanities,  
Tsinghua University, Beijing, China

Dingliu\_thu@126.com xfyang.thu@gmail.com

jiang.mh@mail.tsinghua.edu.cn

## Abstract

In this article, we propose a novel classifier based on quantum computation theory. Different from existing methods, we consider the classification as an evolutionary process of a physical system and build the classifier by using the basic quantum mechanics equation. The performance of the experiments on two datasets indicates feasibility and potentiality of the quantum classifier.

## 1 Introduction

Taking modern natural science into account, the quantum mechanics theory (QM) is one of the most famous and profound theory which brings a world-shaking revolution for physics. Since QM was born, it has been considered as a significant part of theoretic physics and has shown its power in explaining experimental results. Furthermore, some scientists believe that QM is the final principle of physics even the whole natural science. Thus, more and more researchers have expanded the study of QM in other fields of science, and it has affected almost every aspect of natural science and technology deeply, such as quantum computation.

The principle of quantum computation has also affected a lot of scientific researches in computer science, specifically in computational modeling, cryptography theory as well as information theory. Some researchers have employed the principle and technology of quantum computation to improve the studies on Machine Learning (ML) (Aïmeur et al., 2006; Aïmeur et al., 2007; Chen et al., 2008; Gambs, 2008; Horn and Gottlieb, 2001; Nasios and Bors, 2007), a field which studies theories and constructions of systems that can learn from data, among which classification is a typical task. Thus, we attempted to

build a computational model based on quantum computation theory to handle classification tasks in order to prove the feasibility of applying the QM model to machine learning.

In this article, we present a method that considers the classifier as a physical system amenable to QM and treat the entire process of classification as the evolutionary process of a closed quantum system. According to QM, the evolution of quantum system can be described by a unitary operator. Therefore, the primary problem of building a quantum classifier (QC) is to find the correct or optimal unitary operator. We applied classical optimization algorithms to deal with the problem, and the experimental results have confirmed our theory.

The outline of this paper is as follows. First, the basic principle and structure of QC is introduced in section 2. Then, two different experiments are described in section 3. Finally, section 4 concludes with a discussion.

## 2 Basic principle of quantum classifier

As we mentioned in the introduction, the major principle of quantum classifier (QC) is to consider the classifier as a physical system and the whole process of classification as the evolutionary process of a closed quantum system. Thus, the evolution of the quantum system can be described by a unitary operator (unitary matrix), and the remaining job is to find the correct or optimal unitary operator.

### 2.1 Architecture of quantum classifier

The architecture and the whole procedure of data processing of QC are illustrated in Figure 1. As is shown, the key aspect of QC is the optimization part where we employ the optimization algorithm to find an optimal unitary operator  $U'$ .

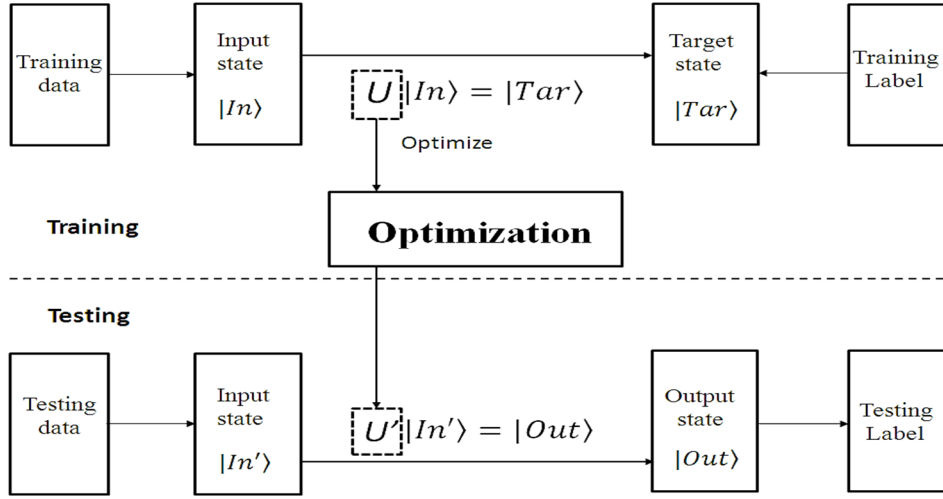


Figure 1. Architecture of quantum classifier

The detailed information about each phase of the process will be explained thoroughly in the following sections.

## 2.2 Encode input state and target state

In quantum mechanics theory, the state of a physical system can be described as a superposition of the so called eigenstates which are orthogonal. Any state, including the eigenstate, can be represented by a complex number vector. We use Dirac's bracket notation to formalize the data as equation 1:

$$|\phi\rangle = \sum_n C_n |E_n\rangle \quad (1)$$

where  $|\phi\rangle$  denotes a state and  $C_n \in \mathbb{C}$  is a complex number with  $C_n = \langle E_n | \phi \rangle$  being the projection of  $|\phi\rangle$  on the eigenstate  $|E_n\rangle$ . According to quantum theory,  $C_n$  denotes the probability amplitude. Furthermore, the probability of  $|\phi\rangle$  collapsing on  $|E_n\rangle$  is  $P(E_n) = \frac{|C_n|^2}{\sum_n |C_n|^2}$ .

Based on the hypothesis that QC can be considered as a quantum system, the input data should be transformed to an available format in quantum theory — the complex number vector. According to Euler's formula, a complex number  $z$  can be denoted as  $z = r e^{i\theta}$  with  $r \geq 0, \theta \in \mathbb{R}$ . Equation 1, thus, can be written as:

$$|\phi\rangle = \sum_n r_n e^{i\theta_n} |E_n\rangle \quad (2)$$

where  $r_n$  and  $\theta_n$  denote the module and the phase of the complex coefficient respectively.

For different applications, we employ different approaches to determine the value of  $r_n$  and  $\theta_n$ . Specifically, in our experiment, we assigned the term frequency, a feature frequently used in text classification to  $r_n$ , and treated the phase  $\theta_n$  as a constant, since we found the phase makes little contribution to the classification.

For each data sample  $sample_k$ , we calculate the corresponding input complex number vector by equation 3, which is illustrated in detail in Figure 2.

$$|\phi_k\rangle = \sum_{j=1}^m r_{jk} \cdot e^{i\theta} |E_j\rangle \quad (3)$$

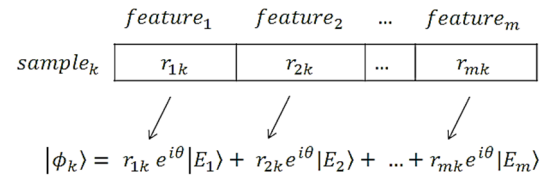


Figure 2. Process of calculating the input state

Each eigenstate  $|E_j\rangle$  denotes the corresponding *feature*<sub>*j*</sub>, resulting in *m* eigenstates for all the samples.

As is mentioned above, the evolutionary process of a closed physical system can be described by a unitary operator, depicted by a matrix as in equation 4:

$$|\phi'\rangle = U|\phi\rangle \quad (4)$$

where  $|\phi'\rangle$  and  $|\phi\rangle$  denote the final state and the initial state respectively. The approach to determine the unitary operator will be discussed in

section 2.3. We encode the target state in the similar way. Like the Vector Space Model(VSM), we use a label matrix to represent each class as in Figure 3.

|           | $Class_1$ | $Class_2$ | ... | $Class_w$ |
|-----------|-----------|-----------|-----|-----------|
| $Label_1$ | $L_{11}$  | $L_{12}$  | ... | $L_{1w}$  |
| $Label_2$ | $L_{21}$  | $L_{22}$  | ... | $L_{2w}$  |
| ...       | ...       | ...       | ... | ...       |
| $Label_w$ | $L_{w1}$  | $L_{w2}$  | ... | $L_{ww}$  |

Figure 3. Label matrix

For each input sample  $sample_k$ , we generate the corresponding target complex number vector according to equation 5:

$$|\varphi_k\rangle = \sum_{j=1}^n L_{jk} \cdot e^{i\theta} |E_j\rangle \quad (5)$$

where each eigenstate  $|E_j\rangle$  represents the corresponding  $Label_j$ , resulting in  $w$  eigenstates for all the labels. Totally, we need  $m + w$  eigenstates, including features and labels.

### 2.3 Finding the Hamiltonian matrix and the Unitary operator

As is mentioned in the first section, finding a unitary operator to describe the evolutionary process is the vital step in building a QC. As a basic quantum mechanics theory, a unitary operator can be represented by a unitary matrix with the property  $U^\dagger = U^{-1}$ , and a unitary operator can also be written as equation 6:

$$U = e^{-\frac{iH}{\hbar}t} \quad (6)$$

where  $H$  is the Hamiltonian matrix and  $\hbar$  is the reduced Planck constant. Moreover, the Hamiltonian  $H$  is a Hermitian matrix with the property  $U^\dagger = (U^t)^* = U$ . The remaining job, therefore, is to find an optimal Hamiltonian matrix.

Since  $H$  is a Hermitian matrix, we only need to determine  $(m + w)^2$  free real parameters, provided that the dimension of  $H$  is  $(m+w)$ . Thus, the problem of determining  $H$  can be regarded as a classical optimization problem, which can be resolved by various optimization algorithms (Chen and Kudlek, 2001). An error function is defined as equation 7:

$$err(H) = \frac{1}{\sum_{(\phi_t, \phi_i) \in T} |\langle \phi_t^k | \phi_o^k \rangle|} \quad (7)$$

where  $T$  is a set of training pairs with  $\phi_t$ ,  $\phi_i$ , and  $\phi_o$  denoting the target, input, and output state respectively, and  $\phi_o$  is determined by  $\phi_i$  as equation 8:

$$|\phi_o\rangle = e^{-\frac{iH}{\hbar}t} |\phi_i\rangle \quad (8)$$

In the optimization phase, we employed several optimization algorithm, including BFGS, Generic Algorithm, and a multi-objective optimization algorithm SQP (sequential quadratic programming) to optimize the error function. In our experiment, the SQP method performed best outperformed the others.

## 3 Experiment

We tested the performance of QC on two different datasets. In section 3.1, the Reuters-21578 dataset was used to train a binary QC. We compared the performance of QC with several classical classification methods, including Support Vector Machine (SVM) and K-nearest neighbor (KNN). In section 3.2, we evaluated the performance on multi-class classification using an oral conversation datasets and analyzed the results.

### 3.1 Reuters-21578

The Reuters dataset we tested contains 3,964 texts belonging to “earnings” category and 8,938 texts belonging to “others” categories. In this classification task, we selected the features by calculating the  $\chi^2$  score of each term from the “earnings” category (Manning and Schütze, 2002).

For the convenience of counting, we adopted 3,900 “earnings” documents and 8,900 “others” documents and divided them into two groups: the training pool and the testing sets. Since we focused on the performance of QC trained by small-scale training sets in our experiment, we each selected 1,000 samples from the “earnings” and the “others” category as our training pool and took the rest of the samples (2,900 “earnings” and 7,900 “others” documents) as our testing sets. We randomly selected training samples from the training pool ten times to train QC, SVM, and KNN classifier respectively and then verified the three trained classifiers on the testing sets, the results of which are illustrated in Figure 4. We noted that the QC performed better than both KNN and SVM on small-scale training sets, when the number of training samples is less than 50.

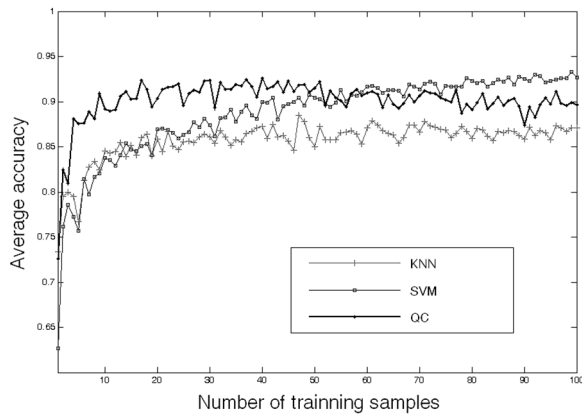


Figure 4. Classification accuracy for Reuters-21578 datasets

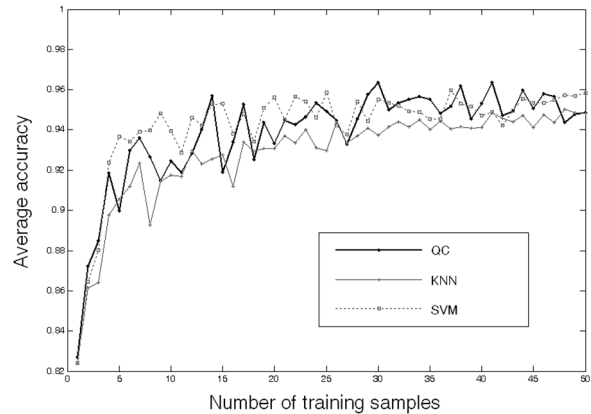


Figure 5. Classification accuracy for oral conversation datasets

Generally speaking, the QC trained by a large training set may not always has an ideal performance. Whereas some single training sample pair led to a favorable result when we used only one sample from each category to train the QC. Actually, some single samples could lead to an accuracy of more than 90%, while some others may produce an accuracy lower than 30%. Therefore, the most significant factor for QC is the quality of the training samples rather than the quantity.

### 3.2 Oral conversation datasets

Besides the binary QC, we also built a multi-class version and tested its performance on an oral conversation dataset which was collected by the Laboratory of Computational Linguistics of Tsinghua university. The dataset consisted of 1,000 texts and were categorized into 5 classes, each containing 200 texts. We still took the term frequency as the feature, the dimension of which exceeded 1,000. We, therefore, utilized the primary component analysis (PCA) to reduce the high dimension of the features in order to decrease the computational complexity. In this experiment, we chose the top 10 primary components of the outcome of PCA, which contained nearly 60% information of the original data. Again, we focused on the performance of QC trained by small-scale training sets. We selected 100 samples from each class to construct the training pool and took the rest of the data as the testing sets. Same to the experiment in section 3.1, we randomly selected the training samples from the training pool ten times to train QC, SVM, and KNN classifier respectively and verified the models on the testing sets, the results of which are shown in Figure 5.

## 4 Discussion

We present here our model of text classification and compare it with SVM and KNN on two datasets. We find that it is feasible to build a supervised learning model based on quantum mechanics theory. Previous studies focus on combining quantum method with existing classification models such as neural network (Chen et al., 2008) and kernel function (Nasios and Bors, 2007) aiming to improve existing models to work faster and more efficiently. Our work, however, focuses on developing a novel method which explores the relationship between machine learning model with physical world, in order to investigate these models by physical rule which describe our universe. Moreover, the QC performs well in text classification compared with SVM and KNN and outperforms them on small-scale training sets. Additionally, the time complexity of QC depends on the optimization algorithm and the amounts of features we adopt. Generally speaking, simulating quantum computing on classical computer always requires more computation resources, and we believe that quantum computer will tackle the difficulty in the forthcoming future. Actually, Google and NASA have launched a quantum computing AI lab this year, and we regard the project as an exciting beginning.

Future studies include: We hope to find a more suitable optimization algorithm for QC and a more reasonable physical explanation towards the “quantum nature” of the QC. We hope our attempt will shed some light upon the application of quantum theory into the field of machine learning.

## Acknowledgments

This work was supported by the National Natural Science Foundation in China (61171114), State Key Lab of Pattern Recognition open foundation, CAS. Tsinghua University Self-determination Research Project (20111081023 & 20111081010) and Human & liberal arts development foundation (2010WKHQ009)

## References

- Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. 2006. Machine Learning in a Quantum World. *Canadian AI 2006*
- Esma Aïmeur, Gilles Brassard and Sébastien Gambs. 2007. Quantum Clustering Algorithms. *Proceedings of the 24 th International Conference on Machine Learning*
- Joseph C.H. Chen and Manfred Kudlek. 2001. Duality of Syntex and Semantics – From the View Point of Brain as a Quantum Computer. *Proceedings of Recent Advances in NLP*
- Joseph C.H. Chen. 2001. Quantum Computation and Natural Language Processing. University of Hamburg, Germany. Ph.D. thesis
- Joseph C.H. Chen. 2001. A Quantum Mechanical Approach to Cognition and Representation. *Consciousness and its Place in Nature, Toward a Science of Consciousness*.
- Cheng-Hung Chen, Cheng-Jian Lin and Chin-Teng Lin. 2008. An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation. *Soft Comput*, 12:567–583.
- Fumiyo Fukumoto and Yoshimi Suzuki. 2002. Manipulating Large Corpora for Text Classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*
- Sébastien Gambs. 2008. Quantum classification, arXiv:0809.0444
- Lov K. Grover. 1997. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Physical Review Letters*, 79,325–328
- David Horn and Assaf Gottlieb. 2001. The Method of Quantum Clustering. *Proceedings of Advances in Neural Information Processing Systems* .
- Christopher D. Manning and Hinrich Schütze. 2002. Foundations of Statistical Natural Language Processing. *MIT Press. Cambridge, Massachusetts, USA*.
- Nikolaos Nasios and Adrian G. Bors. 2007. Kernel-based classification using quantum mechanics. *Pattern Recognition*, 40:875–889
- Hartmut Neven and Vasil S. Denchev. 2009. Training a Large Scale Classifier with the Quantum Adiabatic Algorithm. arXiv:0912.0779v1
- Michael A. Nielsen and Isaac L. Chuang. 2000. Quantum Computation and Quantum Information, *Cambridge University Press, Cambridge, UK*.
- Masahide Sasaki and Alberto Carlini. 2002. Quantum learning and universal quantum matching machine. *Physical Review*, A 66, 022303
- Dan Ventura. 2002. Pattern classification using a quantum system. Proceedings of the Joint Conference on Information Sciences.



# Re-embedding Words

**Igor Labutov**

Cornell University  
iil4@cornell.edu

**Hod Lipson**

Cornell University  
hod.lipson@cornell.edu

## Abstract

We present a fast method for re-purposing existing semantic word vectors to improve performance in a supervised task. Recently, with an increase in computing resources, it became possible to learn rich word embeddings from massive amounts of unlabeled data. However, some methods take days or weeks to learn good embeddings, and some are notoriously difficult to train. We propose a method that takes as input an existing embedding, some labeled data, and produces an embedding in the same space, but with a better predictive performance in the supervised task. We show improvement on the task of sentiment classification with respect to several baselines, and observe that the approach is most useful when the training set is sufficiently small.

## 1 Introduction

Incorporating the vector representation of a word as a feature, has recently been shown to benefit performance in several standard NLP tasks such as language modeling (Bengio et al., 2003; Mnih and Hinton, 2009), POS-tagging and NER (Collobert et al., 2011), parsing (Socher et al., 2010), as well as in sentiment and subjectivity analysis tasks (Maas et al., 2011; Yessenalina and Cardie, 2011). Real-valued word vectors mitigate sparsity by “smoothing” relevant semantic insight gained during the unsupervised training over the rare and unseen terms in the training data. To be effective, these word-representations — and the process by which they are assigned to the words (i.e. embedding) — should capture the semantics relevant to the task. We might, for example, consider *dramatic* (term  $X$ ) and *pleasant* (term  $Y$ ) to correlate with a review of a good movie (task  $A$ ), while finding them of opposite polarity in the context of a

dating profile (task  $B$ ). Consequently, good vectors for  $X$  and  $Y$  should yield an inner product close to 1 in the context of task  $A$ , and  $-1$  in the context of task  $B$ . Moreover, we may already have on our hands embeddings for  $X$  and  $Y$  obtained from yet another (possibly unsupervised) task ( $C$ ), in which  $X$  and  $Y$  are, for example, orthogonal. If the embeddings for task  $C$  happen to be learned from a much larger dataset, it would make sense to reuse task  $C$  embeddings, but adapt them for task  $A$  and/or task  $B$ . We will refer to task  $C$  and its embeddings as the *source task* and the *source embeddings*, and task  $A/B$ , and its embeddings as the *target task* and the *target embeddings*.

Traditionally, we would learn the embeddings for the target task jointly with whatever unlabeled data we may have, in an instance of semi-supervised learning, and/or we may leverage labels from multiple other related tasks in a multi-task approach. Both methods have been applied successfully (Collobert and Weston, 2008) to learn task-specific embeddings. But while joint training is highly effective, a downside is that a large amount of data (and processing time) is required a-priori. In the case of deep neural embeddings, for example, training time can number in days. On the other hand, learned embeddings are becoming more abundant, as much research and computing effort is being invested in learning word representations using large-scale deep architectures trained on web-scale corpora. Many of said embeddings are published and can be harnessed in their raw form as additional features in a number of supervised tasks (Turian et al., 2010). It would, thus, be advantageous to learn a task-specific embedding directly from another (source) embedding.

In this paper we propose a fast method for re-embedding words from a source embedding  $S$  to a target embedding  $T$  by performing unconstrained optimization of a convex objective. Our objective is a linear combination of the dataset’s log-

likelihood under the target embedding and the Frobenius norm of the distortion matrix — a matrix of component-wise differences between the target and the source embeddings. The latter acts as a regularizer that penalizes the Euclidean distance between the source and target embeddings. The method is much faster than joint training and yields competitive results with several baselines.

## 2 Related Work

The most relevant to our contribution is the work by Maas *et.al* (2011), where word vectors are learned specifically for sentiment classification. Embeddings are learned in a semi-supervised fashion, and the components of the embedding are given an explicit probabilistic interpretation. Their method produces state-of-the-art results, however, optimization is non-convex and takes approximately 10 hours on 10 machines<sup>1</sup>. Naturally, our method is significantly faster because it operates in the space of an existing embedding, and does not require a large amount of training data a-priori.

Collobert and Weston (2008), in their seminal paper on deep architectures for NLP, propose a multilayer neural network for learning word embeddings. Training of the model, depending on the task, is reported to be between an hour and three days. While the obtained embeddings can be “fine-tuned” using backpropogation for a supervised task, like all multilayer neural network training, optimization is non-convex, and is sensitive to the dimensionality of the hidden layers.

In machine learning literature, joint semi-supervised embedding takes form in methods such as the LaplacianSVM (LapSVM) (Belkin et al., 2006) and Label Propagation (Zhu and Ghahramani, 2002), to which our approach is related. These methods combine a discriminative learner with a non-linear manifold learning technique in a joint objective, and apply it to a combined set of labeled and unlabeled examples to improve performance in a supervised task. (Weston et al., 2012) take it further by applying this idea to deep-learning architectures. Our method is different in that the (potentially) massive amount of unlabeled data is not required a-priori, but only the resultant embedding.

<sup>1</sup>as reported by author in private correspondence. The runtime can be improved using recently introduced techniques, see (Collobert et al., 2011)

## 3 Approach

Let  $\Phi_S, \Phi_T \in \mathbb{R}^{|V| \times K}$  be the source and target embedding matrices respectively, where  $K$  is the dimension of the word vector space, identical in the source and target embeddings, and  $V$  is the set of embedded words, given by  $V_S \cap V_T$ . Following this notation,  $\phi_i$  — the  $i^{th}$  row in  $\Phi$  — is the respective vector representation of word  $w_i \in V$ . In what follows, we first introduce our supervised objective, then combine it with the proposed regularizer and learn the target embedding  $\Phi_T$  by optimizing the resulting joint convex objective.

### 3.1 Supervised model

We model each document  $d_j \in D$  (a movie review, for example) as a collection of words  $w_{ij}$  (i.i.d samples). We assign a sentiment label  $s_j \in \{0, 1\}$  to each document (converting the star rating to a binary label), and seek to optimize the conditional likelihood of the labels  $(s_j)_{j \in \{1, \dots, |D|\}}$ , given the embeddings and the documents:

$$p(s_1, \dots, s_{|D|} | D; \Phi_T) = \prod_{d_j \in D} \prod_{w_i \in d_j} p(s_j | w_i; \Phi_T)$$

where  $p(s_j = 1 | w_i, \Phi_T)$  is the probability of assigning a positive label to document  $j$ , given that  $w_i \in d_j$ . As in (Maas et al., 2011), we use logistic regression to model the conditional likelihood:

$$p(s_j = 1 | w_i; \Phi_T) = \frac{1}{1 + \exp(-\psi^T \phi_i)}$$

where  $\psi \in \mathbb{R}^{K+1}$  is a regression parameter vector with an included bias component. Maximizing the log-likelihood directly (for  $\psi$  and  $\Phi_T$ ), especially on small datasets, will result in severe overfitting, as learning will tend to commit neutral words to either polarity. Classical regularization will mitigate this effect, but can be improved further by introducing an external embedding in the regularizer. In what follows, we describe *re-embedding regularization*— employing existing (source) embeddings to bias word vector learning.

### 3.2 Re-embedding regularization

To leverage rich semantic word representations, we employ an external *source* embedding and incorporate it in the regularizer on the supervised objective. We use Euclidean distance between the source and the target embeddings as the regular-

ization loss. Combined with the supervised objective, the resulting log-likelihood becomes:

$$\operatorname{argmax}_{\psi, \Phi_T} \sum_{d_j \in D} \sum_{w_i \in d_j} \log p(s_j | w_i; \Phi_T) - \lambda \|\Delta\Phi\|_F^2 \quad (1)$$

where  $\Delta\Phi = \Phi_T - \Phi_S$ ,  $\|\cdot\|_F$  is a Frobenius norm, and  $\lambda$  is a trade-off parameter. There are almost no restrictions on  $\Phi_S$ , except that it must match the desired target vector space dimension  $K$ . The objective is convex in  $\psi$  and  $\Phi_T$ , thus, yielding a unique target re-embedding. We employ L-BFGS algorithm (Liu and Nocedal, 1989) to find the optimal target embedding.

### 3.3 Classification with word vectors

To classify documents, re-embedded word vectors can now be used to construct a document-level feature vector for a supervised learning algorithm of choice. Perhaps the most direct approach is to compute a weighted linear combination of the embeddings for words that appear in the document to be classified, as done in (Maas et al., 2011) and (Blacoe and Lapata, 2012). We use the document’s binary bag-of-words vector  $v_j$ , and compute the document’s vector space representation through the matrix-vector product  $\Phi_T v_j$ . The resulting  $K + 1$ -dimensional vector is then cosine-normalized and used as a feature vector to represent the document  $d_j$ .

## 4 Experiments

**Data:** For our experiments, we employ a large, recently introduced IMDB movie review dataset (Maas et al., 2011), in place of the smaller dataset introduced in (Pang and Lee, 2004) more commonly used for sentiment analysis. The dataset (50,000 reviews) is split evenly between training and testing sets, each containing a balanced set of highly polar ( $\geq 7$  and  $\leq 4$  stars out of 10) reviews.

**Source embeddings:** We employ three external embeddings (obtained from (Turian et al., 2010)) induced using the following models: 1) hierarchical log-bilinear model (HLBL) (Mnih and Hinton, 2009) and two neural network-based models – 2) Collobert and Weston’s (C&W) deep-learning architecture, and 3) Huang *et al.*’s polysemous neural language model (HUANG) (Huang et al., 2012). C&W and HLBL were induced using a 37M-word newswire text (Reuters Corpus 1). We also induce a Latent Semantic Analysis (LSA) based embedding from the subset of the English project Gutenberg collection of approximately 100M words. No

pre-processing (stemming or stopword removal), beyond case-normalization is performed in either the external or LSA-based embedding. For HLBL, C&W and LSA embeddings, we use two variants of different dimensionality: 50 and 200. In total, we obtain seven source embeddings: HLBL-50, HLBL-200, C&W-50, C&W-200, HUANG-50, LSA-50, LSA-200.

**Baselines:** We generate two baseline embeddings – NULL and RANDOM. NULL is a set of zero vectors, and RANDOM is a set of uniformly distributed random vectors with a unit L2-norm. NULL and RANDOM are treated as source vectors and re-embedded in the same way. The NULL baseline is equivalent to regularizing on the target embedding without the source embedding. As additional baselines, we use each of the 7 source embeddings directly as a target without re-embedding.

**Training:** For each source embedding matrix  $\Phi_S$ , we compute the optimal target embedding matrix  $\Phi_T$  by maximizing Equation 1 using the L-BFGS algorithm. 20 % of the training set (5,000 documents) is withheld for parameter ( $\lambda$ ) tuning. We use LIBLINEAR (Fan et al., 2008) logistic regression module to classify document-level embeddings (computed from the  $\Phi_T v_j$  matrix-vector product). Training (re-embedding and document classification) on 20,000 documents and a 16,000 word vocabulary takes approximately 5 seconds on a 3.0 GHz quad-core machine.

## 5 Results and Discussion

The main observation from the results is that our method improves performance for smaller training sets ( $\leq 5000$  examples). The reason for the performance boost is expected – classical regularization of the supervised objective reduces overfitting. However, comparing to the NULL and RANDOM baseline embeddings, the performance is improved noticeably (note that a percent difference of 0.1 corresponds to 20 correctly classified reviews) for word vectors that incorporate the source embedding in the regularizer, than those that do not (NULL), and those that are based on the random source embedding (RANDOM). We hypothesize that the external embeddings, generated from a significantly larger dataset help “smooth” the word-vectors learned from a small labeled dataset alone. Further observations include:

| Features                             | Number of training examples |              |              |                         |              |              |
|--------------------------------------|-----------------------------|--------------|--------------|-------------------------|--------------|--------------|
|                                      |                             |              |              | + Bag-of-words features |              |              |
|                                      | .5K                         | 5K           | 20K          | .5K                     | 5K           | 20K          |
| <b>A. Re-embeddings (our method)</b> |                             |              |              |                         |              |              |
| HLBL-50                              | 74.01                       | 79.89        | 80.94        | 78.90                   | 84.88        | 85.42        |
| HLBL-200                             | 74.33                       | 80.14        | 81.05        | 79.22                   | 85.05        | 85.95        |
| C&W-50                               | 74.52                       | 79.81        | 80.48        | 78.92                   | 84.89        | 85.87        |
| <b>C&amp;W-200</b>                   | <b>74.80</b>                | <b>80.25</b> | <b>81.15</b> | <b>79.34</b>            | <b>85.28</b> | <b>86.15</b> |
| HUANG-50                             | 74.29                       | 79.90        | 79.91        | 79.03                   | 84.89        | 85.61        |
| LSA-50                               | 72.83                       | 79.67        | 80.67        | 78.71                   | 83.44        | 84.73        |
| LSA-200                              | 73.70                       | 80.03        | 80.91        | 79.12                   | 84.83        | 85.31        |
| <b>B. Baselines</b>                  |                             |              |              |                         |              |              |
| RANDOM-50 w/ re-embedding            | 72.90                       | 79.12        | 80.21        | 78.29                   | 84.01        | 84.87        |
| RANDOM-200 w/ re-embedding           | 72.93                       | 79.20        | 80.29        | 78.31                   | 84.08        | 84.91        |
| NULL w/ re-embedding                 | 72.92                       | 79.18        | 80.24        | 78.29                   | 84.10        | 84.98        |
| HLBL-200 w/o re-embedding            | 67.88                       | 72.60        | 73.10        | 79.02                   | 83.83        | 85.83        |
| <b>C&amp;W-200 w/o re-embedding</b>  | <b>68.17</b>                | <b>72.72</b> | <b>73.38</b> | <b>79.30</b>            | <b>85.15</b> | <b>86.15</b> |
| HUANG-50 w/o re-embedding            | 67.89                       | 72.63        | 73.12        | 79.13                   | 84.94        | 85.99        |
| <b>C. Related methods</b>            |                             |              |              |                         |              |              |
| Joint training (Maas, 2011)          | —                           | —            | 84.65        | —                       | —            | 88.90        |
| Bag of Words SVM                     | —                           | —            | —            | 79.17                   | 84.97        | 86.14        |

Table 1: Classification accuracy for the sentiment task (IMDB movie review dataset (Maas et al., 2011)). Subtable A compares performance of the re-embedded vocabulary, induced from a given source embedding. Subtable B contains a set of baselines: *X-w/o re-embedding* indicates using a source embedding *X* directly without re-embedding.

**Training set size:** We note that with a sufficient number of training instances for each word in the test set, additional knowledge from an external embedding does little to improve performance.

**Source embeddings:** We find C&W embeddings to perform best for the task of sentiment classification. These embeddings were found to perform well in other NLP tasks as well (Turian et al., 2010).

**Embedding dimensionality:** We observe that for HLBL, C&W and LSA source embeddings (for all training set sizes), 200 dimensions outperform 50. While a smaller number of dimensions has been shown to work better in other tasks (Turian et al., 2010), re-embedding words may benefit from a larger initial dimension of the word vector space. We leave the testing of this hypothesis for future work.

**Additional features:** Across all embeddings, appending the document’s binary bag-of-words representation increases classification accuracy.

## 6 Future Work

While “semantic smoothing” obtained from introducing an external embedding helps to improve performance in the sentiment classification task, the method does not help to re-embed words that do not appear in the training set to begin with. Returning to our example, if we found *dramatic* and *pleasant* to be “far” in the original (source) embedding space, but re-embed them such that they are “near” (for the task of movie review sentiment

### BORING

source: lethal, lifestyles, masterpiece ...  
target: **idiotic, soft-core, gimmicky**

### BAD

source: past, developing, lesser, ...  
target: **ill, madonna, low, ...**

### DEPRESSING

source: versa, redemption, townsfolk ...  
target: **hate, pressured, unanswered, ...**

### BRILLIANT

source: high-quality, obsession, hate ...  
target: **all-out, bold, smiling ...**

Table 2: A representative set of words from the 20 closest-ranked (cosine-distance) words to (*boring*, *bad*, *depressing*, *brilliant*) extracted from the *source* and *target* (C&W-200) embeddings. Source embeddings give higher rank to words that are related, but not necessarily indicative of sentiment, e.g. *brilliant* and *obsession*. Target words tend to be tuned and ranked higher based on movie-sentiment-based relations.

classification, for example), then we might expect words such as *melodramatic*, *powerful*, *striking*, *enjoyable* to be re-embedded nearby as well, even if they did not appear in the training set. The objective for this optimization problem can be posed by requiring that the distance between every pair of words in the source and target embeddings is preserved as much as possible, i.e.  $\min(\hat{\phi}_i\hat{\phi}_j - \phi_i\phi_j)^2 \quad \forall i, j$  (where, with some abuse of notation,  $\phi$  and  $\hat{\phi}$  are the source and target embeddings respectively). However, this objective is no longer convex in the embeddings. Global re-embedding constitutes our ongoing work and may pose an interesting challenge to the community.

## 7 Conclusion

We presented a novel approach to adapting existing word vectors for improving performance in a text classification task. While we have shown promising results in a single task, we believe that the method is general enough to be applied to a range of supervised tasks and source embeddings. As sophistication of unsupervised methods grows, scaling to ever-more massive datasets, so will the representational power and coverage of induced word vectors. Techniques for leveraging the large amount of unsupervised data, but indirectly through word vectors, can be instrumental in cases where the data is not directly available, training time is valuable and a set of easy low-dimensional “plug-and-play” features is desired.

## 8 Acknowledgements

This work was supported in part by the NSF CDI Grant ECCS 0941561 and the NSF Graduate fellowship. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the sponsoring organizations. The authors would like to thank Thorsten Joachims and Bishan Yang for helpful and insightful discussions.

## References

- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.
- William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.
- 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.
- 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.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.
- Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.
- 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. Association for Computational Linguistics.
- Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. *Advances in neural information processing systems*, 21:1081–1088.
- 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.
- Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*.
- 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*, pages 384–394. Association for Computational Linguistics.
- Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer.
- Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 172–182. Association for Computational Linguistics.
- Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, Carnegie Mellon University.

# LABR: A Large Scale Arabic Book Reviews Dataset

**Mohamed Aly**

Computer Engineering Department  
Cairo University  
Giza, Egypt  
mohamed@mohamedaly.info

**Amir Atiya**

Computer Engineering Department  
Cairo University  
Giza, Egypt  
amir@alumni.caltech.edu

## Abstract

We introduce LABR, the largest sentiment analysis dataset to-date for the Arabic language. It consists of over 63,000 book reviews, each rated on a scale of 1 to 5 stars. We investigate the properties of the dataset, and present its statistics. We explore using the dataset for two tasks: *sentiment polarity classification* and *rating classification*. We provide standard splits of the dataset into training and testing, for both polarity and rating classification, in both balanced and unbalanced settings. We run baseline experiments on the dataset to establish a benchmark.

## 1 Introduction

The internet is full of platforms where users can express their opinions about different subjects, from movies and commercial products to books and restaurants. With the explosion of social media, this has become easier and more prevalent than ever. Mining these troves of unstructured text has become a very active area of research with lots of applications. **Sentiment Classification** is among the most studied tasks for processing opinions (Pang and Lee, 2008; Liu, 2010). In its basic form, it involves classifying a piece of opinion, e.g. a movie or book review, into either having a *positive* or *negative* sentiment. Another form involves predicting the actual rating of a review, e.g. predicting the number of stars on a scale from 1 to 5 stars.

Most of the current research has focused on building sentiment analysis applications for the English language (Pang and Lee, 2008; Liu, 2010; Korayem et al., 2012), with much less work on other languages. In particular, there has been little work on sentiment analysis in Arabic (Abasi et al., 2008; Abdul-Mageed et al., 2011;

Abdul-Mageed et al., 2012; Abdul-Mageed and Diab, 2012b; Korayem et al., 2012), and very few, considerably small-sized, datasets to work with (Rushdi-Saleh et al., 2011b; Rushdi-Saleh et al., 2011a; Abdul-Mageed and Diab, 2012a; Elarnaoty et al., 2012). In this work, we try to address the lack of large-scale Arabic sentiment analysis datasets in this field, in the hope of sparking more interest in research in Arabic sentiment analysis and related tasks. Towards this end, we introduce **LABR**, the **L**arge-scale **A**rabic **B**ook **R**eview dataset. It is a set of over 63K book reviews, each with a rating of 1 to 5 stars.

We make the following contributions: (1) We present the largest Arabic sentiment analysis dataset to-date (up to our knowledge); (2) We provide standard splits for the dataset into training and testing sets. This will make comparing different results much easier. The dataset and the splits are publicly available at [www.mohamedaly.info/datasets](http://www.mohamedaly.info/datasets); (3) We explore the structure and properties of the dataset, and perform baseline experiments for two tasks: *sentiment polarity classification* and *rating classification*.

## 2 Related Work

A few Arabic sentiment analysis datasets have been collected in the past couple of years, we mention the relevant two sets:

**OCA** Opinion Corpus for Arabic (Rushdi-Saleh et al., 2011b) contains 500 movie reviews in Arabic, collected from forums and websites. It is divided into 250 positive and 250 negative reviews, although the division is not standard in that there is no rating for *neutral* reviews i.e. for 10-star rating systems, ratings above and including 5 are considered positive and those below 5 are considered negative.

**AWATIF** is a multi-genre corpus for Modern Standard Arabic sentiment analysis (Abdul-

|                          |           |
|--------------------------|-----------|
| Number of reviews        | 63,257    |
| Number of users          | 16,486    |
| Avg. reviews per user    | 3.84      |
| Median reviews per user  | 2         |
| Number of books          | 2,131     |
| Avg. reviews per book    | 29.68     |
| Median reviews per book  | 6         |
| Median tokens per review | 33        |
| Max tokens per review    | 3,736     |
| Avg. tokens per review   | 65        |
| Number of tokens         | 4,134,853 |
| Number of sentences      | 342,199   |

Table 1: **Important Dataset Statistics.**

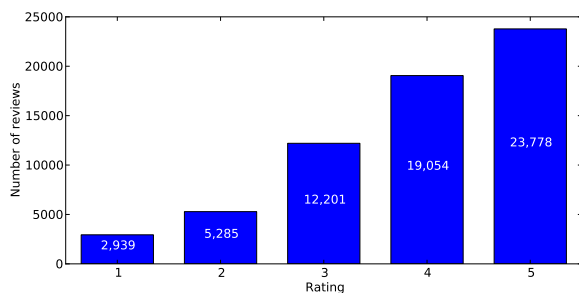


Figure 1: **Reviews Histogram.** The plot shows the number of reviews for each rating.

Mageed and Diab, 2012a). It consists of about 2855 sentences of news wire stories, 5342 sentences from Wikipedia talk pages, and 2532 threaded conversations from web forums.

### 3 Dataset Collection

We downloaded over 220,000 reviews from the book readers social network [www.goodreads.com](http://www.goodreads.com) during the month of March 2013. These reviews were from the first 2143 books in the list of *Best Arabic Books*. After harvesting the reviews, we found out that over 70% of them were not in Arabic, either because some non-Arabic books exist in the list, or because of existing translations of some of the books in other languages. After filtering out the non-Arabic reviews, and performing several pre-processing steps to clean up HTML tags and other unwanted content, we ended up with 63,257 Arabic reviews.

### 4 Dataset Properties

The dataset contains 63,257 reviews that were submitted by 16,486 users for 2,131 different books.

| Task                       |   | Training Set | Test Set |
|----------------------------|---|--------------|----------|
| 1. Polarity Classification | B | 13,160       | 3,288    |
|                            | U | 40,845       | 10,211   |
| 2. Rating Classification   | B | 11,760       | 2,935    |
|                            | U | 50,606       | 12,651   |

Table 2: **Training and Test sets.** **B** stands for balanced, and **U** stands for Unbalanced.

Table 1 contains some important facts about the dataset and Fig. 1 shows the number of reviews for each rating. We consider as *positive* reviews those with ratings 4 or 5, and *negative* reviews those with ratings 1 or 2. Reviews with rating 3 are considered neutral and not included in the polarity classification. The number of positive reviews is much larger than that of negative reviews. We believe this is because the books we got reviews for were the most popular books, and the top rated ones had many more reviews than the the least popular books.

The average user provided 3.84 reviews with the median being 2. The average book got almost 30 reviews with the median being 6. Fig. 2 shows the number of reviews per user and book. As shown in the Fig. 2c, most books and users have few reviews, and vice versa. Figures 2a-b show a box plot of the number of reviews per user and book. We notice that books (and users) tend to have (give) positive reviews than negative reviews, where the median number of positive reviews per book is 5 while that for negative reviews is only 2 (and similarly for reviews per user).

Fig. 3 shows the statistics of tokens and sentences. The reviews were tokenized and “rough” sentence counts were computed (by looking for punctuation characters). The average number of tokens per review is 65.4, the average number of sentences per review is 5.4, and the average number of tokens per sentence is 12. Figures 3a-b show that the distribution is similar for positive and negative reviews. Fig. 3c shows a plot of the frequency of the tokens in the vocabulary in a log-log scale, which conforms to Zipf’s law (Manning and Schütze, 2000).

## 5 Experiments

We explored using the dataset for two tasks: (a) **Sentiment polarity classification**: where the goal is to predict if the review is *positive* i.e. with rating 4 or 5, or is *negative* i.e. with rating 1 or 2; and (b)

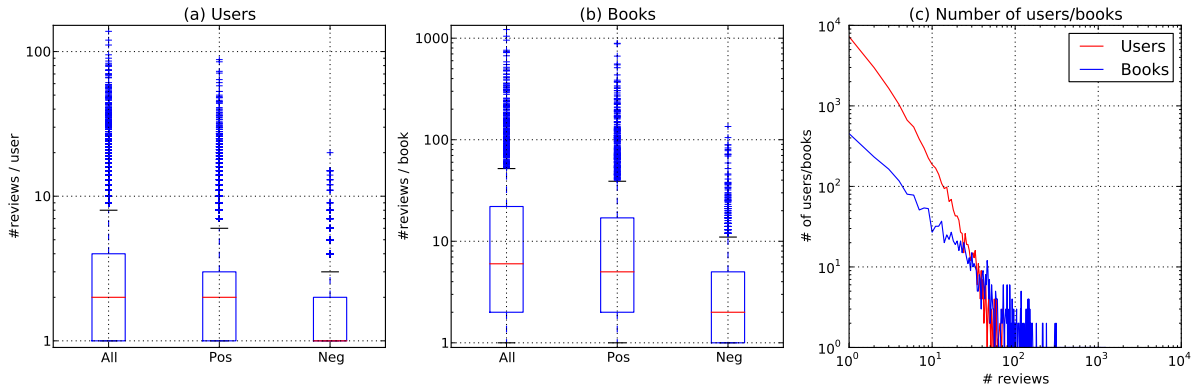


Figure 2: **Users and Books Statistics.** (a) Box plot of the number of reviews per user for all, positive, and negative reviews. The *red* line denotes the median, and the edges of the box the *quartiles*. (b) the number of reviews per book for all, positive, and negative reviews. (c) the number of books/users with a given number of reviews.

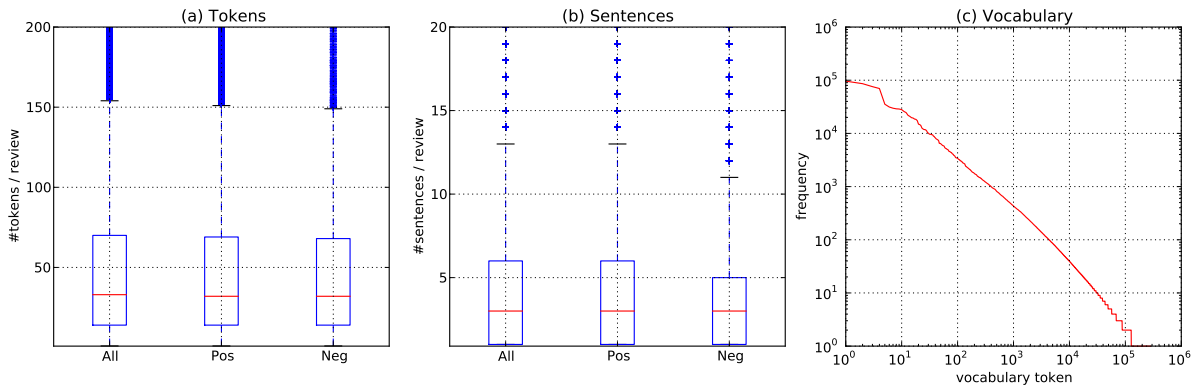


Figure 3: **Tokens and Sentences Statistics.** (a) the number of tokens per review for all, positive, and negative reviews. (b) the number of sentences per review. (c) the frequency distribution of the vocabulary tokens.

**Rating classification:** where the goal is to predict the rating of the review on a scale of 1 to 5.

To this end, we divided the dataset into separate training and test sets, with a ratio of 8:2. We do this because we already have enough training data, so there is no need to resort to cross-validation (Pang et al., 2002). To avoid the bias of having more positive than negative reviews, we explored two settings: (a) a *balanced* split where the number of reviews from *every* class is the same, and is taken to be the size of the smallest class (where larger classes are down-sampled); (b) an *unbalanced* split where the number of reviews from every class is unrestricted, and follows the distribution shown in Fig. 1. Table 2 shows the number of reviews in the training and test sets for each of the two tasks for the balanced and unbalanced splits, while Fig. 4 shows the breakdown of these num-

bers per class.

Tables 3-4 show results of the experiments for both tasks in both balanced/unbalanced settings. We tried different features: unigrams, bigrams, and trigrams with/without tf-idf weighting. For classifiers, we used Multinomial Naive Bayes, Bernoulli Naive Bayes (for binary counts), and Support Vector Machines. We report two measures: the *total* classification accuracy (percentage of correctly classified test examples) and *weighted* F1 measure (Manning and Schütze, 2000). All experiments were implemented in Python using scikit-learn (Pedregosa et al., 2011) and Qalsadi (available at [pypi.python.org/pypi/qalsadi](http://pypi.python.org/pypi/qalsadi)).

We notice that: (a) The total accuracy and weighted F1 are quite correlated and go hand-in-hand. (b) Task 1 is much easier than task 2, which is expected. (c) The unbalanced setting seems eas-



| Features | Tf-Idf | Balanced             |               |               | Unbalanced    |               |                      |
|----------|--------|----------------------|---------------|---------------|---------------|---------------|----------------------|
|          |        | MNB                  | BNB           | SVM           | MNB           | BNB           | SVM                  |
| 1g       | No     | 0.801 / 0.801        | 0.807 / 0.807 | 0.766 / 0.766 | 0.887 / 0.879 | 0.889 / 0.876 | 0.880 / 0.877        |
|          | Yes    | 0.809 / 0.808        | 0.529 / 0.417 | 0.801 / 0.801 | 0.838 / 0.765 | 0.838 / 0.766 | 0.903 / 0.895        |
| 1g+2g    | No     | 0.821 / 0.821        | 0.821 / 0.821 | 0.789 / 0.789 | 0.893 / 0.877 | 0.891 / 0.873 | 0.892 / 0.888        |
|          | Yes    | 0.822 / 0.822        | 0.513 / 0.368 | 0.818 / 0.818 | 0.838 / 0.765 | 0.837 / 0.763 | <b>0.910 / 0.901</b> |
| 1g+2g+3g | No     | 0.821 / 0.821        | 0.823 / 0.823 | 0.786 / 0.786 | 0.889 / 0.869 | 0.886 / 0.863 | 0.893 / 0.888        |
|          | Yes    | <b>0.827 / 0.827</b> | 0.511 / 0.363 | 0.821 / 0.820 | 0.838 / 0.765 | 0.837 / 0.763 | <b>0.910 / 0.901</b> |

Table 3: **Task 1: Polarity Classification Experimental Results.** *1g* means using the unigram model, *1g+2g* is using unigrams + bigrams, and *1g+2g+3g* is using trigrams. *Tf-Idf* indicates whether tf-idf weighting was used or not. *MNB* is Multinomial Naive Bayes, *BNB* is Bernoulli Naive Bayes, and *SVM* is the Support Vector Machine. The numbers represent *total accuracy / weighted F1* measure. See Sec. 5.

| Features | Tf-Idf | Balanced             |               |               | Unbalanced    |               |                      |
|----------|--------|----------------------|---------------|---------------|---------------|---------------|----------------------|
|          |        | MNB                  | BNB           | SVM           | MNB           | BNB           | SVM                  |
| 1g       | No     | 0.393 / 0.392        | 0.395 / 0.396 | 0.367 / 0.365 | 0.465 / 0.445 | 0.464 / 0.438 | 0.460 / 0.454        |
|          | Yes    | 0.402 / 0.405        | 0.222 / 0.128 | 0.387 / 0.384 | 0.430 / 0.330 | 0.379 / 0.229 | 0.482 / 0.472        |
| 1g+2g    | No     | 0.407 / 0.408        | 0.418 / 0.421 | 0.383 / 0.379 | 0.487 / 0.460 | 0.487 / 0.458 | 0.472 / 0.466        |
|          | Yes    | 0.419 / 0.423        | 0.212 / 0.098 | 0.411 / 0.407 | 0.432 / 0.325 | 0.379 / 0.217 | 0.501 / 0.490        |
| 1g+2g+3g | No     | 0.405 / 0.408        | 0.417 / 0.420 | 0.384 / 0.381 | 0.487 / 0.457 | 0.484 / 0.452 | 0.474 / 0.467        |
|          | Yes    | <b>0.426 / 0.431</b> | 0.211 / 0.093 | 0.410 / 0.407 | 0.431 / 0.322 | 0.379 / 0.216 | <b>0.503 / 0.491</b> |

Table 4: **Task 2: Rating Classification Experimental Results.** See Table 3 and Sec. 5.

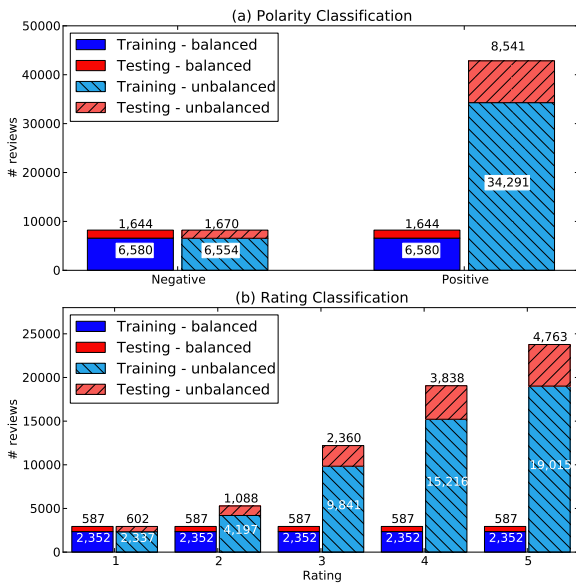


Figure 4: **Training-Test Splits.** (a) Histogram of the number of training and test reviews for the polarity classification task for *balanced* (solid) and *unbalanced* (hatched) cases. (b) The same for the rating classification task. In the balanced set, all classes have the same number of reviews as the smallest class, which is done by down-sampling the larger classes.

ier than the balanced one. This might be because the unbalanced sets contain more training examples to make use of. (d) SVM does much better in the unbalanced setting, while MNB is slightly better than SVM in the balanced setting. (e) Using more ngrams helps, and especially combined with tf-idf weighting, as all the best scores are with tf-idf.

## 6 Conclusion and Future Work

In this work we presented the largest Arabic sentiment analysis dataset to-date. We explored its properties and statistics, provided standard splits, and performed several baseline experiments to establish a benchmark. Although we used very simple features and classifiers, task 1 achieved quite good results (~90% accuracy) but there is much room for improvement in task 2 (~50% accuracy). We plan next to work more on the dataset to get sentence-level polarity labels, and to extract Arabic sentiment lexicon and explore its potential. Furthermore, we also plan to explore using Arabic-specific and more powerful features.

## References

- Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*.
- Muhammad Abdul-Mageed and Mona Diab. 2012a. Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*.
- Muhammad Abdul-Mageed and Mona Diab. 2012b. Toward building a large-scale arabic sentiment lexicon. In *Proceedings of the 6th International Global Word-Net Conference*.
- Muhammad Abdul-Mageed, Mona Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard arabic. In *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.
- Muhammad Abdul-Mageed, Sandra Kübler, and Mona Diab. 2012. Samar: A system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*.
- Mohamed Elarnaoty, Samir AbdelRahman, and Aly Fahmy. 2012. A machine learning approach for opinion holder extraction in arabic language. *arXiv preprint arXiv:1206.1011*.
- Mohammed Korayem, David Crandall, and Muhammad Abdul-Mageed. 2012. Subjectivity and sentiment analysis of arabic: A survey. In *Advanced Machine Learning Technologies and Applications*.
- Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*.
- Christopher D. Manning and Hinrich Schütze. 2000. *Foundations of Statistical Natural Language Processing*. MIT Press.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135.
- B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *EMNLP*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- M. Rushdi-Saleh, M. Martín-Valdivia, L. Ureña-López, and J. Perea-Ortega. 2011a. Bilingual experiments with an arabic-english corpus for opinion mining. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*.
- M. Rushdi-Saleh, M. Martín-Valdivia, L. Ureña-López, and J. Perea-Ortega. 2011b. Oca: Opinion corpus for arabic. *Journal of the American Society for Information Science and Technology*.

# Generating Recommendation Dialogs by Extracting Information from User Reviews

Kevin Reschke, Adam Vogel, and Dan Jurafsky

Stanford University

Stanford, CA, USA

{kreschke, acvogel, jurafsky}@stanford.edu

## Abstract

Recommendation dialog systems help users navigate e-commerce listings by asking questions about users' preferences toward relevant domain attributes. We present a framework for generating and ranking fine-grained, highly relevant questions from user-generated reviews. We demonstrate our approach on a new dataset just released by Yelp, and release a new sentiment lexicon with 1329 adjectives for the restaurant domain.

## 1 Introduction

Recommendation dialog systems have been developed for a number of tasks ranging from product search to restaurant recommendation (Chai et al., 2002; Thompson et al., 2004; Bridge et al., 2005; Young et al., 2010). These systems learn user requirements through spoken or text-based dialog, asking questions about particular attributes to filter the space of relevant documents.

Traditionally, these systems draw questions from a small, fixed set of attributes, such as cuisine or price in the restaurant domain. However, these systems overlook an important element in users' interactions with online product listings: user-generated reviews. Huang et al. (2012) show that information extracted from user reviews greatly improves user experience in visual search interfaces. In this paper, we present a dialog-based interface that takes advantage of review texts. We demonstrate our system on a new challenge corpus of 11,537 businesses and 229,907 user reviews released by the popular review website Yelp<sup>1</sup>, focusing on the dataset's 4724 restaurants and bars (164,106 reviews).

This paper makes two main contributions. First, we describe and qualitatively evaluate a frame-

work for generating new, highly-relevant questions from user review texts. The framework makes use of techniques from topic modeling and sentiment-based aspect extraction to identify fine-grained attributes for each business. These attributes form the basis of a new set of questions that the system can ask the user.

Second, we use a method based on information-gain for dynamically ranking candidate questions during dialog production. This allows our system to select the most informative question at each dialog step. An evaluation based on simulated dialogs shows that both the ranking method and the automatically generated questions improve recall.

## 2 Generating Questions from Reviews

### 2.1 Subcategory Questions

Yelp provides each business with category labels for top-level cuisine types like *Japanese*, *Coffee & Tea*, and *Vegetarian*. Many of these top-level categories have natural subcategories (e.g., *ramen* vs. *sushi*). By identifying these subcategories, we enable questions which probe one step deeper than the top-level category label.

To identify these subcategories, we run Latent Dirichlet Analysis (LDA) (Blei et al., 2003) on the reviews of each set of businesses in the twenty most common top-level categories, using 10 topics and concatenating all of a business's reviews into one document.<sup>2</sup> Several researchers have used sentence-level documents to model topics in reviews, but these tend to generate topics about fine-grained aspects of the sort we discuss in Section 2.2 (Jo and Oh, 2011; Brody and Elhadad, 2010). We then manually labeled the topics, discarding junk topics and merging similar topics. Table 1 displays sample extracted subcategories.

Using these topic models, we assign a business

<sup>1</sup>[https://www.yelp.com/dataset\\_challenge/](https://www.yelp.com/dataset_challenge/)

<sup>2</sup>We use the Topic Modeling Toolkit implementation: <http://nlp.stanford.edu/software/tmt>

| Category       | Topic Label   | Top Words  |
|----------------|---|--|
| Italian        | pizza<br>traditional<br>bistro<br>deli                                  | crust sauce pizza garlic sausage slice salad<br>pasta sauce delicious ravioli veal dishes gnocchi<br>bruschetta patio salad valet delicious brie panini<br>sandwich deli salad pasta delicious grocery meatball  |
| American (New) | brew pub<br>grill<br>bar<br>bistro<br>brunch<br>burger<br>mediterranean | beers peaks ale brewery patio ipa brew<br>steak salad delicious sliders ribs tots drinks<br>drinks vig bartender patio uptown dive karaoke<br>drinks pretzel salad fondue patio sandwich windsor<br>sandwich brunch salad delicious pancakes patio<br>burger fries sauce beef potato sandwich delicious<br>pita hummus jungle salad delicious mediterranean wrap |
| Delis          | italian<br>new york<br>bagels<br>mediterranean<br>sandwiches            | deli sandwich meats cannoli cheeses authentic sausage<br>deli beef sandwich pastrami corned fries waitress<br>bagel sandwiches toasted lox delicious donuts yummy<br>pita lemonade falafel hummus delicious salad bakery<br>sandwich subs sauce beef tasty meats delicious   |
| Japanese       | sushi<br>teppanyaki<br>teriyaki<br>ramen                                | sushi kyoto zen rolls tuna sashimi spicy<br>sapporo chef teppanyaki sushi drinks shrimp fried<br>teriyaki sauce beef bowls veggies spicy grill<br>noodles udon dishes blossom delicious soup ramen   |

Table 1: A sample of subcategory topics with hand-labels and top words.

to a subcategory based on the topic with highest probability in that business’s topic distribution. Finally, we use these subcategory topics to generate questions for our recommender dialog system. Each top-level category corresponds to a single question whose potential answers are the set of subcategories: e.g., “What type of Japanese cuisine do you want?”

## 2.2 Questions from Fine-Grained Aspects

Our second source for questions is based on aspect extraction in sentiment summarization (Blair-Goldensohn et al., 2008; Brody and Elhadad, 2010). We define an aspect as any noun-phrase which is targeted by a sentiment predicate. For example, from the sentence “The place had **great atmosphere**, but the **service** was **slow**.” we extract two aspects: *+atmosphere* and *-service*.

Our aspect extraction system has two steps. First we develop a domain specific sentiment lexicon. Second, we apply syntactic patterns to identify NPs targeted by these sentiment predicates.

### 2.2.1 Sentiment Lexicon

**Coordination Graph** We generate a list of domain-specific sentiment adjectives using graph propagation. We begin with a seed set combining PARADIGM+ (Jo and Oh, 2011) with ‘strongly subjective’ adjectives from the OpinionFinder lexicon (Wilson et al., 2005), yielding 1342 seeds. Like Brody and Elhadad (2010), we then construct a coordination graph that links adjectives modifying the same noun, but to increase precision we

require that the adjectives also be conjoined by *and* (Hatzivassiloglou and McKeown, 1997). This reduces problems like propagating positive sentiment to *orange* in *good orange chicken*. We marked adjectives that follow *too* or lie in the scope of negation with special prefixes and treated them as distinct lexical entries.

**Sentiment Propagation** Negative and positive seeds are assigned values of 0 and 1 respectively. All other adjectives begin at 0.5. Then a standard propagation update is computed iteratively (see Eq. 3 of Brody and Elhadad (2010)).

In Brody and Elhadad’s implementation of this propagation method, seed sentiment values are fixed, and the update step is repeated until the non-seed values converge. We found that three modifications significantly improved precision. First, we omit candidate nodes that don’t link to at least two positive or two negative seeds. This eliminated spurious propagation caused by one-off parsing errors. Second, we run the propagation algorithm for fewer iterations (two iterations for negative terms and one for positive terms). We found that additional iterations led to significant error propagation when neutral (*italian*) or ambiguous (*thick*) terms were assigned sentiment.<sup>3</sup> Third, we update both non-seed and seed adjectives. This allows us to learn, for example, that the negative seed *decadent* is positive in the restaurant domain.

Table 2 shows a sample of sentiment adjectives

<sup>3</sup>Our results are consistent with the recent finding of Whitney and Sarkar (2012) that cautious systems are better when bootstrapping from seeds.

| Negative Sentiment   |
|--|
| institutional, underwhelming, not_nice, burn-tish, unidentifiable, inefficient, not_attentive, grotesque, confused, trashy, insufferable, grandiose, not_pleasant, timid, degrading, laughable, under-seasoned, dismayed, torn |
| Positive Sentiment   |
| decadent, satisfied, lovely, stupendous, sizable, nutritious, intense, peaceful, not_expensive, elegant, rustic, fast, affordable, efficient, congenial, rich, not_too_heavy, wholesome, bustling, lush                        |

Table 2: Sample of Learned Sentiment Adjectives

derived by this graph propagation method. The final lexicon has 1329 adjectives<sup>4</sup>, including 853 terms not in the original seed set. The lexicon is available for download.<sup>5</sup>

**Evaluative Verbs** In addition to this adjective lexicon, we take 56 evaluative verbs such as *love* and *hate* from *admire*-class VerbNet predicates (Kipper-Schuler, 2005).

### 2.2.2 Extraction Patterns

To identify noun-phrases which are targeted by predicates in our sentiment lexicon, we develop hand-crafted extraction patterns defined over syntactic dependency parses (Blair-Goldensohn et al., 2008; Somasundaran and Wiebe, 2009) generated by the Stanford parser (Klein and Manning, 2003). Table 3 shows a sample of the aspects generated by these methods.

**Adj + NP** It is common practice to extract any NP modified by a sentiment adjective. However, this simple extraction rule suffers from precision problems. First, reviews often contain sentiment toward irrelevant, non-business targets (*Wayne* is the target of *excellent job* in (1)). Second, hypothetical contexts lead to spurious extractions. In (2), the extraction *+service* is clearly wrong—in fact, the opposite sentiment is being expressed.

- (1) Wayne did an **excellent job** addressing our needs and giving us our options.
- (2) Nice and airy atmosphere, but **service** could be more **attentive** at times.

<sup>4</sup>We manually removed 26 spurious terms which were caused by parsing errors or propagation to a neutral term.

<sup>5</sup><http://nlp.stanford.edu/projects/yelp.shtml>

We address these problems by filtering out sentences in hypothetical contexts cued by *if*, *should*, *could*, or a question mark, and by adopting the following, more conservative extractions rules:

- i) [BIZ + *have* + adj. + NP] Sentiment adjective modifies NP, main verb is *have*, subject is business name, *it*, *they*, *place*, or absent. (E.g., *This place has some really great yogurt and toppings*).
- ii) [NP + *be* + adj.] Sentiment adjective linked to NP by *be*—e.g., *Our pizza was much too jalapeno-y*.

**“Good For” + NP** Next, we extract aspects using the pattern BIZ + positive adj. + *for* + NP, as in *It’s perfect for a date night*. Examples of extracted aspects include *+lunch*, *+large groups*, *+drinks*, and *+quick lunch*.

**Verb + NP** Finally, we extract NPs that appear as direct object to one of our evaluative verbs (e.g., *We loved the fried chicken*).

### 2.2.3 Aspects as Questions

We generate questions from these extracted aspects using simple templates. For example, the aspect *+burritos* yields the question: *Do you want a place with good burritos?*

## 3 Question Selection for Dialog

To utilize the questions generated from reviews in recommendation dialogs, we first formalize the dialog optimization task and then offer a solution.

### 3.1 Problem Statement

We consider a version of the Information Retrieval Dialog task introduced by Kopeček (1999). Businesses  $b \in B$  have associated *attributes*, coming from a set  $Att$ . These attributes are a combination of Yelp categories and our automatically extracted aspects described in Section 2. Attributes  $att \in Att$  take values in a finite domain  $\text{dom}(att)$ . We denote the subset of businesses with an attribute  $att$  taking value  $val \in \text{dom}(att)$ , as  $B|_{att=val}$ . Attributes are functions from businesses to subsets of values:  $att : B \rightarrow \mathcal{P}(\text{dom}(att))$ . We model a user *information need*  $I$  as a set of attribute/value pairs:  $I = \{(att_1, val_1), \dots, (att_{|I|}, val_{|I|})\}$ .

Given a set of businesses and attributes, a *recommendation agent*  $\pi$  selects an attribute to ask

|   |   |
|---|---|
| <b>Chinese:</b><br>+beef +egg roll +sour soup +orange chicken<br>+noodles +crab puff +egg drop soup<br>+dim sum +fried rice +honey chicken<br><b>Japanese:</b><br>+rolls +sushi rolls +wasabi +sushi bar +salmon<br>+chicken katsu +crunch +green tea +sake selection<br>+oysters +drink menu +sushi selection +quality | <b>Mexican:</b><br>+salsa bar +burritos +fish tacos +guacamole<br>+enchiladas +hot sauce +carne asade +breakfast burritos<br>+horchata +green salsa +tortillas +quesadillas<br><b>American (New)</b><br>+environment +drink menu +bar area +cocktails +brunch<br>+hummus +mac and cheese +outdoor patio +seating area<br>+lighting +brews +sangria +cheese plates |
|---|---|

Table 3: Sample of the most frequent positive aspects extracted from review texts.

**Input:** Information need  $I$

Set of businesses  $B$

Set of attributes  $\text{Att}$

Recommendation agent  $\pi$

Dialog length  $K$

**Output:** Dialog history  $H$

Recommended businesses  $B$

Initialize dialog history  $H = \emptyset$

**for**  $step = 0$ ;  $step < K$ ;  $step++$  **do**

Select an attribute:  $att = \pi(B, H)$

Query user for the answer:  $val = I(att)$

Restrict set of businesses:  $B = B|_{att=val}$

Append answer:  $H = H \cup \{(att, val)\}$

**end**

Return  $(H, B)$

**Algorithm 1:** Procedure for evaluating a recommendation agent

the user about, then uses the answer value to narrow the set of businesses to those with the desired attribute value, and selects another query. Algorithm 1 presents this process more formally. The recommendation agent can use both the set of businesses  $B$  and the history of question and answers  $H$  from the user to select the next query. Thus, formally a recommendation agent is a function  $\pi : B \times H \rightarrow \text{Att}$ . The dialog ends after a fixed number of queries  $K$ .

### 3.2 Information Gain Agent

The *information gain recommendation agent* chooses questions to ask the user by selecting question attributes that maximize the entropy of the resulting document set, in a manner similar to decision tree learning (Mitchell, 1997). Formally, we define a function  $infogain : \text{Att} \times \mathcal{P}(B) \rightarrow \mathbb{R}$ :

$$infogain(att, B) = - \sum_{vals \in \mathcal{P}(\text{dom}(att))} \frac{|B_{att=vals}|}{|B|} \log \frac{|B_{att=vals}|}{|B|}$$

The agent then selects questions  $att \in \text{Att}$  that maximize the information gain with respect to the

set of businesses satisfying the dialog history  $H$ :

$$\pi(B, H) = \arg \max_{att \in \text{Att}} infogain(att, B|_H)$$

## 4 Evaluation

### 4.1 Experimental Setup

We follow the standard approach of using the attributes of an individual business as a simulation of a user’s preferences (Chung, 2004; Young et al., 2010). For every business  $b \in B$  we form an information need composed of all of  $b$ ’s attributes:

$$I_b = \bigcup_{\{att \in \text{Att} | att(b) \neq \emptyset\}} (att, att(b))$$

To evaluate a recommendation agent, we use the *recall* metric, which measures how well an information need is satisfied. For each information need  $I$ , let  $B_I$  be the set of businesses that satisfy the questions of an agent. We define the recall of the set of businesses with respect to the information need as

$$\text{recall}(B_I, I) = \frac{\sum_{b \in B_I} \sum_{(att, val) \in I} \mathbb{1}[val \in att(b)]}{|B_I| |I|}$$

We average recall across all information needs, yielding *average recall*.

We compare against a *random agent* baseline that selects attributes  $att \in \text{Att}$  uniformly at random at each time step. Other recommendation dialog systems such as Young et al. (2010) select questions from a small fixed hierarchy, which is not applicable to our large set of attributes.

### 4.2 Results

Figure 1 shows the average recall for the random agent versus the information gain agent with varying sets of attributes. ‘Top-level’ repeatedly queries the user’s top-level category preferences, ‘Subtopic’ additionally uses our topic modeling subcategories, and ‘All’ uses these plus the aspects extracted from reviews. We see that for sufficiently long dialogs, ‘All’ outperforms the other systems. The ‘Subtopic’ and ‘Top-level’ systems plateau after a few dialog steps once they’ve asked

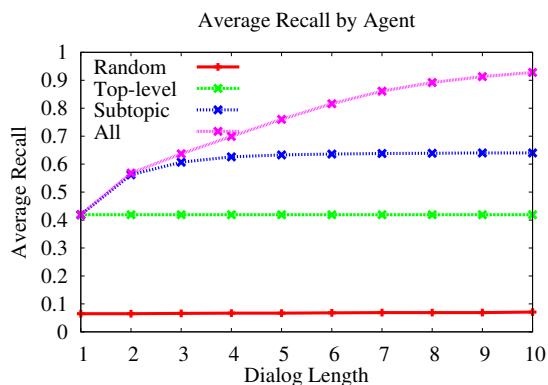


Figure 1: Average recall for each agent.

all useful questions. For instance, most businesses only have one or two top-level categories, so after the system has identified the top-level category that the user is interested in, it has no more good questions to ask. Note that the information gain agent starts dialogs with the top-level and appropriate subcategory questions, so it is only for longer dialogs that the fine-grained aspects boost performance.

Below we show a few sample output dialogs from our ‘All’ information gain agent.

- Q:** What kind of place do you want?  
**A:** American (New)  
**Q:** What kind of American (New) do you want: bar, bistro, standard, burgers, brew pub, or brunch?  
**A:** bistro  
**Q:** Do you want a place with a good patio?  
**A:** Yes
- Q:** What kind of place do you want?  
**A:** Chinese  
**Q:** What kind of Chinese place do you want: buffet, dim sum, noodles, pan Asian, Panda Express, sit down, or veggie?  
**A:** sit down  
**Q:** Do you want a place with a good lunch special?  
**A:** Yes
- Q:** What kind of place do you want?  
**A:** Mexican  
**Q:** What kind of Mexican place do you want: dinner, taqueria, margarita bar, or tortas?  
**A:** Margarita bar  
**Q:** Do you want a place with a good patio?

**A:** Yes

## 5 Conclusion

We presented a system for extracting large sets of attributes from user reviews and selecting relevant attributes to ask questions about. Using topic models to discover subtypes of businesses, a domain-specific sentiment lexicon, and a number of new techniques for increasing precision in sentiment aspect extraction yields attributes that give a rich representation of the restaurant domain. We have made this 1329-term sentiment lexicon for the restaurant domain available as useful resource to the community. Our information gain recommendation agent gives a principled way to dynamically combine these diverse attributes to ask relevant questions in a coherent dialog. Our approach thus offers a new way to integrate the advantages of the curated hand-built attributes used in statistical slot and filler dialog systems, and the distributionally induced, highly relevant categories built by sentiment aspect extraction systems.

## 6 Acknowledgments

Thanks to the anonymous reviewers and the Stanford NLP group for helpful suggestions. The authors also gratefully acknowledge the support of the Nuance Foundation, the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-13-2-0040, ONR grants N00014-10-1-0109 and N00014-13-1-0287 and ARO grant W911NF-07-1-0216, and the Center for Advanced Study in the Behavioral Sciences.

## References

- Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. 2005. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320.
- Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews.

- In *Proceedings of HLT NAACL 2010*, pages 804–812.
- Joyce Chai, Veronika Horvath, Nicolas Nicolov, Margo Stys, A Kambhatla, Wlodek Zadrozny, and Prem Melville. 2002. Natural language assistant - a dialog system for online product recommendation. *AI Magazine*, 23:63–75.
- Grace Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of ACL 2004*, pages 63–70.
- Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of EACL 1997*, pages 174–181.
- Jeff Huang, Oren Etzioni, Luke Zettlemoyer, Kevin Clark, and Christian Lee. 2012. Revminer: An extractive interface for navigating reviews on a smartphone. In *Proceedings of UIST 2012*.
- Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 815–824.
- Karin Kipper-Schuler. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon.
- Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings ACL 2003*, pages 423–430.
- I. Kopeček. 1999. Modeling of the information retrieval dialogue systems. In *Proceedings of the Workshop on Text, Speech and Dialogue-TSD 99, Lectures Notes in Artificial Intelligence 1692*, pages 302–307. Springer-Verlag.
- Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.
- Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of ACL 2009*, pages 226–234.
- Cynthia A. Thompson, Mehmet H. Goeker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research (JAIR)*, 21:393–428.
- Max Whitney and Anoop Sarkar. 2012. Bootstrapping via graph propagation. In *Proceedings of the ACL 2012*, pages 620–628, Jeju Island, Korea.
- Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 on Interactive Demonstrations*, pages 34–35.
- Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, April.



# Exploring Sentiment in Social Media: Bootstrapping Subjectivity Clues from Multilingual Twitter Streams

**Svitlana Volkova**

CLSP

Johns Hopkins University  
Baltimore, MD

svitlana@jhu.edu

**Theresa Wilson**

HLTCOE

Johns Hopkins University  
Baltimore, MD

taw@jhu.edu

**David Yarowsky**

CLSP

Johns Hopkins University  
Baltimore, MD

yarowsky@cs.jhu.edu

## Abstract

We study subjective language in social media and create Twitter-specific lexicons via bootstrapping sentiment-bearing terms from multilingual Twitter streams. Starting with a domain-independent, high-precision sentiment lexicon and a large pool of unlabeled data, we bootstrap Twitter-specific sentiment lexicons, using a small amount of labeled data to guide the process. Our experiments on English, Spanish and Russian show that the resulting lexicons are effective for sentiment classification for many under-explored languages in social media.

## 1 Introduction

The language that people use to express opinions and sentiment is extremely diverse. This is true for well-formed data, such as news and reviews, and it is particularly true for data from social media. Communication in social media is informal, abbreviations and misspellings abound, and the person communicating is often trying to be funny, creative, and entertaining. Topics change rapidly, and people invent new words and phrases.

The dynamic nature of social media together with the extreme diversity of subjective language has implications for any system with the goal of analyzing sentiment in this domain. General, domain-independent sentiment lexicons have low coverage. Even models trained specifically on social media data may degrade somewhat over time as topics change and new sentiment-bearing terms crop up. For example, the word “occupy” would not have been indicative of sentiment before 2011.

Most of the previous work on sentiment lexicon construction relies on existing natural language

processing tools, e.g., syntactic parsers (Wiebe, 2000), information extraction (IE) tools (Riloff and Wiebe, 2003) or rich lexical resources such as WordNet (Esuli and Sebastiani, 2006). However, such tools and lexical resources are not available for many languages spoken in social media. While English is still the top language in Twitter, it is no longer the majority. Thus, the applicability of these approaches is limited. Any method for analyzing sentiment in microblogs or other social media streams must be easily adapted to (1) many low-resource languages, (2) the dynamic nature of social media, and (3) working in a streaming mode with limited or no supervision.

Although bootstrapping has been used for learning sentiment lexicons in other domains (Turney and Littman, 2002; Banea et al., 2008), it has not yet been applied to learning sentiment lexicons for microblogs. In this paper, we present an approach for bootstrapping subjectivity clues from Twitter data, and evaluate our approach on English, Spanish and Russian Twitter streams. Our approach:

- handles the informality, creativity and the dynamic nature of social media;
- does not rely on language-dependent tools;
- scales to the hundreds of new under-explored languages and dialects in social media;
- classifies sentiment in a streaming mode.

To bootstrap subjectivity clues from Twitter streams we rely on three main assumptions:

- i. sentiment-bearing terms of similar orientation tend to co-occur at the tweet level (Turney and Littman, 2002);
- ii. sentiment-bearing terms of opposite orientation do not co-occur at the tweet level (Gamon and Aue, 2005);
- iii. the co-occurrence of domain-specific and domain-independent subjective terms serves as a signal of subjectivity.

## 2 Related Work

Mihalcea et al. (2012) classifies methods for bootstrapping subjectivity lexicons into two types: corpus-based and dictionary-based.

Dictionary-based methods rely on existing lexical resources to bootstrap sentiment lexicons. Many researchers have explored using relations in WordNet (Miller, 1995), e.g., Esuli and Sabastiani (2006), Andreevskaia and Bergler (2006) for English, Rao and Ravichandran (2009) for Hindi and French, and Perez-Rosas et al. (2012) for Spanish. Mohammad et al. (2009) use a thesaurus to aid in the construction of a sentiment lexicon for English. Other works (Clematide and Klenner, 2010; Abdul-Mageed et al., 2011) automatically expands and evaluates German and Arabic lexicons. However, the lexical resources that dictionary-based methods need, do not yet exist for the majority of languages in social media. There is also a mismatch between the formality of many language resources, such as WordNet, and the extremely informal language of social media.

Corpus-based methods extract subjectivity and sentiment lexicons from large amounts of unlabeled data using different similarity metrics to measure the relatedness between words. Hatzivassiloglou and McKeown (1997) were the first to explore automatically learning the polarity of words from corpora. Early work by Wiebe (2000) identifies clusters of subjectivity clues based on their distributional similarity, using a small amount of data to bootstrap the process. Turney (2002) and Velikovich et al. (2010) bootstrap sentiment lexicons for English from the web by using Pointwise Mutual Information (PMI) and graph propagation approach, respectively. Kaji and Kitsuregawa (2007) propose a method for building sentiment lexicon for Japanese from HTML pages. Banea et al. (2008) experiment with Lexical Semantic Analysis (LSA) (Dumais et al., 1988) to bootstrap a subjectivity lexicon for Romanian. Kanayama and Nasukawa (2006) bootstrap subjectivity lexicons for Japanese by generating subjectivity candidates based on word co-occurrence patterns.

In contrast to other corpus-based bootstrapping methods, we evaluate our approach on multiple languages, specifically English, Spanish, and Russian. Also, as our approach relies only on the availability of a bilingual dictionary for translating an English subjectivity lexicon and crowdsourcing for help in selecting seeds, it is more scalable and

better able to handle the informality and the dynamic nature of social media. It also can be effectively used to bootstrap sentiment lexicons for any language for which a bilingual dictionary is available or can be automatically induced from parallel corpora.

## 3 Data

For the experiments in this paper, we use three sets of data for each language: 1M *unlabeled* tweets (BOOT) for bootstrapping Twitter-specific lexicons, 2K labeled tweets for development data (DEV), and 2K labeled tweets for evaluation (TEST). DEV is used for parameter tuning while bootstrapping, and TEST is used to evaluating the quality of the bootstrapped lexicons.

We take English tweets from the corpus constructed by Burger et al. (2011) which contains 2.9M tweets (excluding retweets) from 184K users.<sup>1</sup> English tweets are identified automatically using a compression-based language identification (LID) tool (Bergsma et al., 2012). According to LID, there are 1.8M (63.6%) English tweets, which we randomly sample to create BOOT, DEV and TEST sets for English. Unfortunately, Burger’s corpus does not include Russian and Spanish data on the same scale as English. Therefore, for other languages we construct a new Twitter corpus by downloading tweets from followers of region-specific news and media feeds.

Sentiment labels for tweets in DEV and TEST sets for all languages are obtained using Amazon Mechanical Turk. For each tweet we collect annotations from five workers and use majority vote to determine the final label for the tweet. Snow et al. (2008) show that for a similar task, labeling emotion and valence, on average four non-expert labelers are needed to achieve an expert level of annotation. Table 1 gives the distribution of tweets over sentiment labels for the development and test sets for English (E-DEV, E-TEST), Spanish (S-DEV, S-TEST), and Russian (R-DEV, R-TEST). Below are examples of tweets in Russian with English translations labeled with sentiment:

- Positive: В планах вкусный завтрак и куча фильмов (Planning for delicious breakfast and lots of movies);
- Negative: Хочу сдохнуть, и я это сделаю (I want to die and I will do that);

<sup>1</sup>They provided the tweet IDs, and we used the Twitter Corpus Tools to download the tweets.

| Data   | Positive | Neg | Both | Neutral |
|--------|----------|-----|------|---------|
| E-DEV  | 617      | 357 | 202  | 824     |
| E-TEST | 596      | 347 | 195  | 862     |
| S-DEV  | 358      | 354 | 86   | 1,202   |
| S-TEST | 317      | 387 | 93   | 1203    |
| R-DEV  | 452      | 463 | 156  | 929     |
| R-TEST | 488      | 380 | 149  | 983     |

Table 1: Sentiment label distribution in development DEV and test TEST datasets across languages.

- **Both:** Хочется написать грубее про фильм но не буду. Хотя актеры хороши (I want to write about the movie rougher but I will not. Although the actors are good);
- **Neutral:** Почему умные мысли приходят только ночью? (Why clever thoughts come only at night?).

## 4 Lexicon Bootstrapping

To create a Twitter-specific sentiment lexicon for a given language, we start with a general-purpose, high-precision sentiment lexicon<sup>2</sup> and bootstrap from the unlabeled data (BOOT) using the labeled development data (DEV) to guide the process.

### 4.1 High-Precision Subjectivity Lexicons

For English we seed the bootstrapping process with the strongly subjective terms from the MPQA lexicon<sup>3</sup> (Wilson et al., 2005). These terms have been previously shown to be high-precision for recognizing subjective sentences (Riloff and Wiebe, 2003).

For the other languages, the subjective seed terms are obtained by translating English seed terms using a bilingual dictionary, and then collecting judgments about term subjectivity from Mechanical Turk. Terms that truly are strongly subjective in translation are used for seed terms in the new language, with term polarity projected from the English. Finally, we expand the lexicons with plurals and inflectional forms for adverbs, adjectives and verbs.

### 4.2 Bootstrapping Approach

To bootstrap, first the new lexicon  $L_{B(0)}$  is seeded with the strongly subjective terms from the original lexicon  $L_I$ . On each iteration  $i \geq 1$ , tweets in the unlabeled data are labeled using the lexicon

from the previous iteration,  $L_{B(i-1)}$ . If a tweet contains one or more terms from  $L_{B(i-1)}$  it is considered subjective, otherwise objective. The polarity of subjective tweets is determined in a similar way: if the tweet contains  $\geq 1$  positive terms, taking into account the negation, it is considered negative; if it contains  $\geq 1$  negative terms, taking into account the negation, it is considered positive.<sup>4</sup> If it contains both positive and negative terms, it is considered to be both. Then, for every term not in  $L_{B(i-1)}$  that has a frequency  $\geq \theta_{freq}$ , the probability of that term being subjective is calculated as shown in Algorithm 1 line 10. The top  $\theta_k$  terms with a subjective probability  $\geq \theta_{pr}$  are then added to  $L_{B(i)}$ . The polarity of new terms is determined based on the probability of the term appearing in positive or negative tweets as shown in line 18.<sup>5</sup> The bootstrapping process terminates when there are no more new terms meeting the criteria to add.

---

#### Algorithm 1 BOOTSTRAP ( $\sigma, \theta_{pr}, \theta_{freq}, \theta_{topK}$ )

---

```

1:  $iter = 0, \sigma = 0.5, L_B(\vec{\theta}) \leftarrow L_I(\sigma)$ 
2: while ( $stop \neq true$ ) do
3:    $L_B^{iter}(\vec{\theta}) \leftarrow \emptyset, \Delta L_B^{iter}(\vec{\theta}) \leftarrow \emptyset$ 
4:   for each new term  $w \in \{V \setminus L_B(\vec{\theta})\}$  do
5:     for each tweet  $t \in T$  do
6:       if  $w \in t$  then
7:         UPDATE  $c(w, L_B(\vec{\theta})), c(w, L_B^{pos}(\vec{\theta})), c(w)$ 
8:       end if
9:     end for
10:     $p^{subj}(w) \leftarrow \frac{c(w, L_B(\vec{\theta}))}{c(w)}$ 
11:     $p^{pos}(w) \leftarrow \frac{c(w, L_B^{pos}(\vec{\theta}))}{c(w, L_B(\vec{\theta}))}$ 
12:     $L_B^{iter}(\vec{\theta}) \leftarrow w, p^{subj}(w), p^{pol}(w)$ 
13:  end for
14:  SORT  $L_B^{iter}(\vec{\theta})$  by  $p^{subj}(w)$ 
15:  while ( $K \leq \theta_{topK}$ ) do
16:    for each new term  $w \in L_B^{iter}(\vec{\theta})$  do
17:      if [ $p^{subj}(w) \geq \theta_{pr}$  and  $c_w \geq \theta_{freq}$ ] then
18:        if [ $p^{pos}(w) \geq 0.5$ ] then
19:           $w^{pol} \leftarrow positive$ 
20:        else
21:           $w^{pol} \leftarrow negative$ 
22:        end if
23:         $\Delta L_B^{iter}(\vec{\theta}) \leftarrow \Delta L_B^{iter}(\vec{\theta}) + w^{pol}$ 
24:      end if
25:    end for
26:     $K = K + 1$ 
27:  end while
28:  if [ $\Delta L_B^{iter}(\vec{\theta}) == 0$ ] then
29:     $stop \leftarrow true$ 
30:  end if
31:   $L_B(\vec{\theta}) \leftarrow L_B(\vec{\theta}) + \Delta L_B^{iter}(\vec{\theta})$ 
32:   $iter = iter + 1$ 
33: end while

```

---

<sup>2</sup>Other works on generating domain-specific sentiment lexicons e.g., from blog data (Jijkoun et al., 2010) also start with a general, domain-specific lexicon.

<sup>3</sup><http://www.cs.pitt.edu/mpqa/>

<sup>4</sup>If there is a negation in the two words before a sentiment term, we flip its polarity.

<sup>5</sup>Polarity association probabilities should sum up to 1  $p^{pos}(w|L_B(\vec{\theta})) + p^{neg}(w|L_B(\vec{\theta})) = 1$ .

|              | English    |             | Spanish    |             | Russian    |             |
|--------------|------------|-------------|------------|-------------|------------|-------------|
|              | $L_I^E$    | $L_B^E$     | $L_I^S$    | $L_B^S$     | $L_I^R$    | $L_B^R$     |
| <b>Pos</b>   | 2.3        | 16.8        | 2.9        | 7.7         | 1.4        | 5.3         |
| <b>Neg</b>   | 2.8        | 4.7         | 5.2        | 14.6        | 2.3        | 5.5         |
| <b>Total</b> | <b>5.1</b> | <b>21.5</b> | <b>8.1</b> | <b>22.3</b> | <b>3.7</b> | <b>10.8</b> |

Table 2: The original and the bootstrapped (highlighted) lexicon term count ( $L_I \subset L_B$ ) with polarity across languages (thousands).

The set of parameters  $\vec{\theta}$  is optimized using a grid search on the development data using F-measure for subjectivity classification. As a result, for English  $\vec{\theta} = [0.7, 5, 50]$  meaning that on each iteration the top 50 new terms with a frequency  $\geq 5$  and probability  $\geq 0.7$  are added to the lexicon. For Spanish, the set of optimal parameters  $\vec{\theta} = [0.65, 3, 50]$  and for Russian -  $\vec{\theta} = [0.65, 3, 50]$ . In Table 2 we report size and term polarity from the original  $L_I$  and the bootstrapped  $L_B$  lexicons.

## 5 Lexicon Evaluations

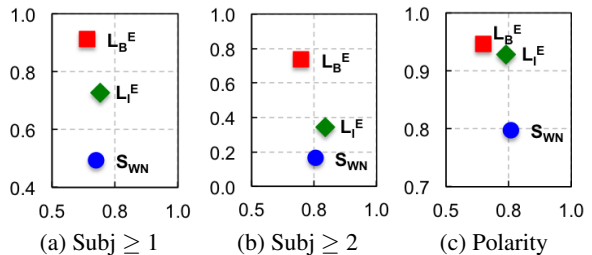
We evaluate our bootstrapped sentiment lexicons English  $L_B^E$ , Spanish  $L_B^S$  and Russian  $L_B^R$  by comparing them with existing *dictionary-expanded* lexicons that have been previously shown to be effective for subjectivity and polarity classification (Esuli and Sebastiani, 2006; Perez-Rosas et al., 2012; Chetviorkin and Loukachevitch, 2012). For that we perform subjectivity and polarity classification using rule-based classifiers<sup>6</sup> on the test data E-TEST, S-TEST and R-TEST.

We consider how the various lexicons perform for rule-based classifiers for both subjectivity and polarity. The subjectivity classifier predicts that a tweet is subjective if it contains a) at least one, or b) at least two subjective terms from the lexicon. For the polarity classifier, we predict a tweet to be positive (negative) if it contains at least one positive (negative) term taking into account negation. If the tweet contains both positive and negative terms, we take the majority label.

For English we compare our bootstrapped lexicon  $L_B^E$  against the original lexicon  $L_I^E$  and strongly subjective terms from SentiWordNet 3.0 (Esuli and Sebastiani, 2006). To make a fair comparison, we automatically expand SentiWordNet with noun plural forms and verb inflectional forms. In Figure 1 we report precision, recall

<sup>6</sup>Similar approach to a rule-based classification using terms from the MPQA lexicon (Riloff and Wiebe, 2003).

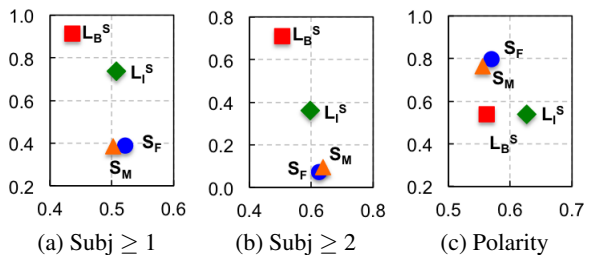
and F-measure results. They show that our bootstrapped lexicon significantly outperforms SentiWordNet for subjectivity classification. For polarity classification we get comparable F-measure but much higher recall for  $L_B^E$  compared to  $S_{WN}$ .



| Lexicon  | $F_{subj \geq 1}$ | $F_{subj \geq 2}$ | $F_{polarity}$ |
|----------|-------------------|-------------------|----------------|
| $S_{WN}$ | 0.57              | 0.27              | 0.78           |
| $L_I^E$  | 0.71              | 0.48              | <b>0.82</b>    |
| $L_B^E$  | <b>0.75</b>       | <b>0.72</b>       | 0.78           |

Figure 1: Precision (x-axis), recall (y-axis) and F-measure (in the table) for English:  $L_I^E$  = initial lexicon,  $L_B^E$  = bootstrapped lexicon,  $S_{WN}$  = strongly subjective terms from SentiWordNet.

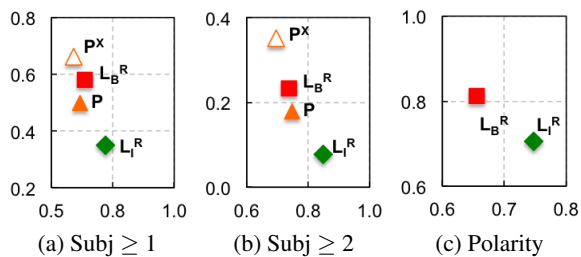
For Spanish we compare our bootstrapped lexicon  $L_B^S$  against the original  $L_I^S$  lexicon, and the full and medium strength terms from the Spanish sentiment lexicon constructed by Perez-Rosas et al. (2012). We report precision, recall and F-measure in Figure 2. We observe that our bootstrapped lexicon yields significantly better performance for subjectivity classification compared to both full and medium strength terms. However, our bootstrapped lexicon yields lower recall and similar precision for polarity classification.



| Lexicon | $F_{subj \geq 1}$ | $F_{subj \geq 2}$ | $F_{polarity}$ |
|---------|-------------------|-------------------|----------------|
| $S_M$   | 0.44              | 0.17              | 0.64           |
| $S_F$   | 0.47              | 0.13              | <b>0.66</b>    |
| $L_I^S$ | 0.59              | 0.45              | 0.58           |
| $L_B^S$ | <b>0.59</b>       | <b>0.59</b>       | 0.55           |

Figure 2: Precision (x-axis), recall (y-axis) and F-measure (in the table) for Spanish:  $L_I^S$  = initial lexicon,  $L_B^S$  = bootstrapped lexicon,  $S_F$  = full strength terms;  $S_M$  = medium strength terms.

For Russian we compare our bootstrapped lexicon  $L_B^R$  against the original  $L_I^R$  lexicon, and the Russian sentiment lexicon constructed by Chetviorkin and Loukachevitch (2012). The external lexicon in Russian  $P$  was built for the domain of product reviews and does not include polarity judgments for subjective terms. As before, we expand the external lexicon with the inflectional forms for adverbs, adjectives and verbs. We report results for Russian in Figure 3. We find that for subjectivity our bootstrapped lexicon shows better performance compared to the external lexicon (5k terms). However, the expanded external lexicon (17k terms) yields higher recall with a significant drop in precision. Note that for Russian, we report polarity classification results for  $L_B^R$  and  $L_I^R$  lexicons only because  $P$  does not have polarity labels.



| Lexicon | $F_{subj \geq 1}$ | $F_{subj \geq 2}$ | $F_{polarity}$ |
|---------|-------------------|-------------------|----------------|
| $P$     | 0.55              | 0.29              | –              |
| $P^X$   | 0.62              | 0.47              | –              |
| $L_I^R$ | 0.46              | 0.13              | 0.73           |
| $L_B^R$ | <b>0.61</b>       | <b>0.35</b>       | <b>0.73</b>    |

Figure 3: Precision (x-axis), recall (y-axis) and F-measure for Russian:  $L_I^R$  = initial lexicon,  $L_B^R$  = bootstrapped lexicon,  $P$  = external sentiment lexicon,  $P^X$  = expanded external lexicon.

We next perform error analysis for subjectivity and polarity classification for all languages and identify common errors to address them in future.

For subjectivity classification we observe that applying part-of-speech tagging during the bootstrapping could improve results for all languages. We could further improve the quality of the lexicon and reduce false negative errors (subjective tweets classified as neutral) by focusing on sentiment-bearing terms such as adjective, adverbs and verbs. However, POS taggers for Twitter are only available for a limited number of languages such as English (Gimpel et al., 2011). Other false negative errors are often caused by misspellings.<sup>7</sup>

<sup>7</sup>For morphologically-rich languages, our approach covers different linguistic forms of terms but not their misspellings. However, it can be fixed by an edit-distance check.

We also find subjective tweets with philosophical thoughts and opinions misclassified, especially in Russian, e.g., Иногда мы бываем не готовы к исполнению заветной мечты но все равно так не хочется ее спугнуть (Sometimes we are not ready to fulfill our dreams yet but, at the same time, we do not want to scare them). Such tweets are difficult to classify using lexicon-based approaches and require deeper linguistic analysis.

False positive errors for subjectivity classification happen because some terms are weakly subjective and can be used in both subjective and neutral tweets e.g., the Russian term хвастаться (brag) is often used as subjective, but in a tweet никогда не стоит хвастаться будущим (never brag about your future) it is used as neutral. Similarly, the Spanish term *buenas* (good) is often used subjectively but it is used as neutral in the following tweet “@Diveke me falta el buenas! jaja que onda que ha pasado” (I miss the good times we had, haha that wave has passed!).

For polarity classification, most errors happen because our approach relies on either positive or negative polarity scores for a term but not both.<sup>8</sup> However, in the real world terms may sometimes have both usages. Thus, some tweets are misclassified (e.g., “It is too warm outside”). We can fix this by summing over weighted probabilities rather than over term counts. Additional errors happen because tweets are very short and convey multiple messages (e.g., “What do you mean by unconventional? Sounds exciting!”) Thus, our approach can be further improved by adding word sense disambiguation and anaphora resolution.

## 6 Conclusions

We propose a scalable and language independent bootstrapping approach for learning subjectivity clues from Twitter streams. We demonstrate the effectiveness of the bootstrapping procedure by comparing the resulting subjectivity lexicons with state-of-the-art sentiment lexicons. We perform error analysis to address the most common error types in the future. The results confirm that the approach can be effectively exploited and further improved for subjectivity classification for many under-explored languages in social media.

<sup>8</sup>During the bootstrapping we calculate probability for a term to be positive and negative, e.g.,  $p(warm|+) = 0.74$  and  $p(warm|-) = 0.26$ . But during polarity classification we rely on the highest probability score and consider it to be “the polarity” for the term e.g., positive for *warm*.

## References

- Muhammad Abdul-Mageed, Mona T. Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of ACL/HLT*.
- Alina Andreevskaia and Sabine Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of EACL*.
- Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *Proceedings of LREC*.
- Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In *Proceedings of 2nd Workshop on Language in Social Media*.
- John D. Burger, John C. Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on Twitter. In *Proceedings of EMNLP*.
- Iliia Chetviorkin and Natalia V. Loukachevitch. 2012. Extraction of Russian sentiment lexicon for product meta-domain. In *Proceedings of COLING*.
- Simon Clematide and Manfred Klenner. 2010. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*.
- Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. 1988. Using latent semantic analysis to improve access to textual information. In *Proceedings of SIGCHI*.
- Andrea Esuli and Fabrizio Sebastiani. 2006. Senti-WordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*.
- Michael Gamon and Anthony Aue. 2005. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*.
- Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: annotation, features, and experiments. In *Proceedings of ACL*.
- Vasileios Hatzivassiloglou and Kathy McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of ACL*.
- Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of ACL*.
- Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of EMNLP*.
- Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of EMNLP*.
- Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2012. Multilingual subjectivity and sentiment analysis. In *Proceedings of ACL*.
- George A. Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11).
- Saif Mohammad, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of EMNLP*.
- Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning sentiment lexicons in Spanish. In *Proceedings of LREC*.
- Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of EACL*.
- Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of EMNLP*.
- Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast – but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*.
- Peter D. Turney and Michael L. Littman. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *Computing Research Repository*.
- Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*.
- Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Proceedings of NAACL*.
- Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of AAAI*.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of EMNLP*.

# Joint Modeling of News Reader's and Comment Writer's Emotions

Huanhuan Liu<sup>†</sup> Shoushan Li<sup>†‡\*</sup> Guodong Zhou<sup>†</sup> Chu-Ren Huang<sup>‡</sup> Peifeng Li<sup>†</sup>

<sup>†</sup>Natural Language Processing Lab  
Soochow University, China  
{huanhuanliu.suda, shoushan.li,  
churenhuang}@gmail.com

<sup>‡</sup>Department of CBS  
the Hong Kong Polytechnic University  
{gdzhou, pfli}@suda.edu.cn

## Abstract

Emotion classification can be generally done from both the writer's and reader's perspectives. In this study, we find that two foundational tasks in emotion classification, i.e., reader's emotion classification on the news and writer's emotion classification on the comments, are strongly related to each other in terms of coarse-grained emotion categories, i.e., *negative* and *positive*. On the basis, we propose a respective way to jointly model these two tasks. In particular, a co-training algorithm is proposed to improve semi-supervised learning of the two tasks. Experimental evaluation shows the effectiveness of our joint modeling approach.

## 1 Introduction

Emotion classification aims to predict the emotion categories (e.g., *happy*, *angry*, or *sad*) of a given text (Quan and Ren, 2009; Das and Bandyopadhyay, 2009). With the rapid growth of computer mediated communication applications, such as social websites and micro-blogs, the research on emotion classification has been attracting more and more attentions recently from the natural language processing (NLP) community (Chen et al., 2010; Purver and Battersby, 2012).

In general, a single text may possess two kinds of emotions, writer's emotion and reader's emotion, where the former concerns the emotion expressed by the writer when writing the text and the latter concerns the emotion expressed by a reader after reading the text. For example, consider two short texts drawn from a news and corresponding comments, as shown in Figure 1. On

one hand, for the news text, while its writer just objectively reports the news and thus does not express his emotion in the text, a reader could yield *sad* or *worried* emotion. On the other hand, for the comment text, its writer clearly expresses his *sad* emotion while the emotion of a reader after reading the comments is not clear (Some may feel *sorry* but others might feel *careless*).

### News:

*Today's Japan earthquake could be  
2011 quake aftershock. ....*

**News Writer's emotion:** None

**News Reader's emotion:** *sad, worried*

### Comments:

(1) *I hope everything is ok, so sad. I still can  
not forget last year.*

(2) *My father-in-law got to experience this  
quake... what a suffering.*

**Comment Writer's emotion:** *sad*

**Comment Reader's emotion:** Unknown

Figure 1: An example of writer's and reader's emotions on a news and its comments

Accordingly, emotion classification can be grouped into two categories: reader's emotion and writer's emotion classifications. Although both emotion classification tasks have been widely studied in recent years, they are always considered independently and treated separately.

However, news and their corresponding comments often appear simultaneously. For example, in many news websites, it is popular to see a news followed by many comments. In this case, because the writers of the comments are a part of the readers of the news, the writer's emotions on the comments are exactly certain reflection of the reader's emotions on the news. That is, the comment writer's emotions and the news reader's emotions are strongly related. For example,

\* Corresponding author

in Figure 1, the comment writer’s emotion ‘*sad*’ is among the news reader’s emotions.

Above observation motivates joint modeling of news reader’s and comment writer’s emotions. In this study, we systematically investigate the relationship between the news reader’s emotions and the comment writer’s emotions. Specifically, we manually analyze their agreement in a corpus collected from a news website. It is interesting to find that such agreement only applies to coarse-grained emotion categories (i.e., *positive* and *negative*) with a high probability and does not apply to fine-grained emotion categories (e.g., *happy*, *angry*, and *sad*). This motivates our joint modeling in terms of the coarse-grained emotion categories. Specifically, we consider the news text and the comment text as two different views of expressing either the news reader’s or comment writer’s emotions. Given the two views, a co-training algorithm is proposed to perform semi-supervised emotion classification so that the information in the unlabeled data can be exploited to improve the classification performance.

## 2 Related Work

### 2.1 Comment Writer’s Emotion Classification

Comment writer’s emotion classification has been a hot research topic in NLP during the last decade (Pang et al., 2002; Turney, 2002; Alm et al., 2005; Wilson et al., 2009) and previous studies can be mainly grouped into two categories: coarse-grained and fine-grained emotion classification.

Coarse-grained emotion classification, also called sentiment classification, concerns only two emotion categories, such as *like* or *dislike* and *positive* or *negative* (Pang and Lee, 2008; Liu, 2012). This kind of emotion classification has attracted much attention since the pioneer work by Pang et al. (2002) in the NLP community due to its wide applications (Cui et al., 2006; Riloff et al., 2006; Dasgupta and Ng, 2009; Li et al., 2010; Li et al., 2011).

In comparison, fine-grained emotion classification aims to classify a text into multiple emotion categories, such as *happy*, *angry*, and *sad*. One main group of related studies on this task is about emotion resource construction, such as emotion lexicon building (Xu et al., 2010; Volkova et al., 2012) and sentence-level or document-level corpus construction (Quan and Ren, 2009; Das and Bandyopadhyay, 2009). Besides, all the related studies focus on supervised learn-

ing (Alm et al., 2005; Aman and Szpakowicz, 2008; Chen et al., 2010; Purver and Battersby, 2012; Moshfeghi et al., 2011), and so far, we have not seen any studies on semi-supervised learning on fine-grained emotion classification.

### 2.2 News Reader’s Emotion Classification

While comment writer’s emotion classification has been extensively studied, there are only a few studies on news reader’s emotion classification from the NLP and related communities.

Lin et al. (2007) first describe the task of reader’s emotion classification on the news articles and then employ some standard machine learning approaches to train a classifier for determining the reader’s emotion towards a news. Their further study, Lin et al. (2008) exploit more features and achieve a higher performance.

Unlike all the studies mentioned above, our study is the first attempt on exploring the relationship between comment writer’s emotion classification and news reader’s emotion classification.

## 3 Relationship between News Reader’s and Comment Writer’s Emotions

To investigate the relationship between news reader’s and comment writer’s emotions, we collect a corpus of Chinese news articles and their corresponding comments from Yahoo! Kimo News (<http://tw.news.yahoo.com>), where each news article is voted with emotion tags from eight categories: *happy*, *sad*, *angry*, *meaningless*, *boring*, *heartwarming*, *worried*, and *useful*. These emotion tags on each news are selected by the readers of the news. Note that because the categories of “*useful*” and “*meaningless*” are not real emotion categories, we ignore them in our study. Same as previous studies of Lin et al. (2007) and Lin et al. (2008), we consider the voted emotions as reader’s emotions on the news, i.e., the news reader’s emotions. We only select the news articles with a dominant emotion (possessing more than 50% votes) in our data. Besides, as we attempt to consider the comment writer’s emotions, the news articles without any comments are filtered.

As a result, we obtain a corpus of 3495 news articles together with their comments and the numbers of the articles of *happy*, *sad*, *angry*, *boring*, *heartwarming*, and *worried* are 1405, 230, 1673, 75, 92 and 20 respectively. For coarse-grained categories, *happy* and *heartwarming* are merged into the *positive* category while



*sad*, *angry*, *boring* and *worried* are merged into the *negative* category.

Besides the tags of the reader’s emotions, each news article is followed by some comments, which can be seen as a reflection of the writer’s emotions (Averagely, each news is followed by 15 comments). In order to know the exact relationship between these two kinds of emotions, we select 20 news from each category and ask two human annotators, named **A** and **B**, to manually annotate the writer’s emotion (single-label) according to the comments of each news. Table 1 reports the agreement on annotators and emotions, measured with Cohen’s kappa ( $\kappa$ ) value (Cohen, 1960).

|            | $\kappa$ Value<br>(Fine-grained<br>emotions) | $\kappa$ Value<br>(Coarse-grained<br>emotions) |
|------------|--|--|
| Annotators | 0.566  | 0.742  |
| Emotions   | 0.504  | 0.756  |

Table 1: Agreement on annotators and emotions

**Agreement between two annotators:** The annotation agreement between the two annotators is 0.566 on the fine-grained emotion categories and 0.742 on the coarse-grained emotion categories.

**Agreement between news reader’s and comment writer’s emotions:** We compare the news reader’s emotion (automatically extracted from the web page) and the comment writer’s emotion (manually annotated by annotator **A**). The annotation agreement between the two kinds of emotions is 0.504 on the fine-grained emotion categories and 0.756 on the coarse-grained emotion categories. From the results, we can see that the agreement on the fine-grained emotions is a bit low while the agreement between the coarse-grained emotions, i.e., *positive* and *negative*, is very high. We find that although some fine-grained emotions of the comments are not consistent with the dominant emotion of the news, they belong to the same coarse-grained category.

In a word, the agreement between news reader’s and comment writer’s emotions on the coarse-grained emotions is very high, even higher than the agreement between the two annotators (0.754 vs. 0.742).

In the following, we focus on the coarse-grained emotions in emotion classification.

#### 4 Joint Modeling of News Reader’s and Comment Writer’s Emotions

Given the importance of both news reader’s and comment writer’s emotion classification as de-

scribed in Introduction and the close relationship between news reader’s and comment writer’s emotions as described in last section, we systematically explore their joint modeling on the two kinds of emotion classification.

In semi-supervised learning, the unlabeled data is exploited to improve the models with a small amount of the labeled data. In our approach, we consider the news text and the comment text as two different views to express the news or comment emotion and build the two classifiers  $C_N$  and  $C_C$ . Given the two-view classifiers, we perform co-training for semi-supervised emotion classification, as shown in Figure 2, on both news reader’s and comment writer’s emotion classification.

#### Input:

- $L_{News}$  the labeled data on the news
- $L_{Comment}$  the labeled data on the comments
- $U_{News}$  the unlabeled data on the news
- $U_{Comment}$  the labeled data on the comments

#### Output:

- $L_{News}$  New labeled data on the news
- $L_{Comment}$  New labeled data on the comments

#### Procedure:

- Loop for  $N$  iterations until  $U_{News} = \emptyset$  or  $U_{Comment} = \emptyset$
- (1). Learn classifier  $C_N$  with  $L_{News}$
  - (2). Use  $C_N$  to label the samples from  $U_{News}$
  - (3). Choose  $n_1$  *positive* and  $n_1$  *negative* news  $N_1$  most confidently predicted by  $C_N$
  - (4). Choose corresponding comments  $M_1$  (the comments of the news in  $N_1$ )
  - (5). Learn classifier  $C_C$  with  $L_{Comment}$
  - (6). Use  $C_C$  to label the samples from  $U_{Comment}$
  - (7). Choose  $n_2$  *positive* and  $n_2$  *negative* comments  $M_2$  most confidently predicted by  $C_C$
  - (8). Choose corresponding comments  $N_2$  (the news of the comments in  $M_2$ )
  - (9).  $L_{News} = L_{News} + N_1 + N_2$   
 $L_{Comment} = L_{Comment} + M_1 + M_2$
  - (10).  $U_{News} = U_{News} - N_1 - N_2$   
 $U_{Comment} = U_{Comment} - M_1 - M_2$

Figure 2: Co-training algorithm for semi-supervised emotion classification

## 5 Experimentation

### 5.1 Experimental Settings

**Data Setting:** The data set includes 3495 news articles (1572 positive and 1923 negative) and their comments as described in Section 3. Although the emotions of the comments are not given in the website, we just set their coarse-grained emotion categories the same as the emotions of their source news due to their close relationship, as described in Section 3. To make the data balanced, we randomly select 1500 positive and 1500 negative news with their comments for the empirical study. Among them, we randomly select 400 news with their comments as the test data.

**Features:** Each news or comment text is treated as a bag-of-words and transformed into a binary vector encoding the presence or absence of word unigrams.

**Classification algorithm:** the maximum entropy (ME) classifier implemented with the public tool, Mallet Toolkits\*.

### 5.2 Experimental Results

**News reader’s emotion classifier:** The classifier trained with the news text.

**Comment writer’s emotion classifier:** The classifier trained with the comment text.

Figure 3 demonstrates the performances of the news reader’s and comment writer’s emotion classifiers trained with the 10 and 50 initial labeled samples plus automatically labeled data from co-training. Here, in each iteration, we pick 2 positive and 2 negative most confident samples, i.e.,  $n_1 = n_2 = 2$ . From this figure, we can see that our co-training algorithm is very effective: using only 10 labeled samples in each category achieves a very promising performance on either news reader’s or comment writer’s emotion classification. Especially, the performance when using only 10 labeled samples is comparable to that when using more than 1200 labeled samples on supervised learning of comment writer’s emotion classification.

For comparison, we also implement a self-training algorithm for the news reader’s and comment writer’s emotion classifiers, each of which automatically labels the samples from the unlabeled data independently. For news reader’s emotion classification, the performances of self-training are 0.783 and 0.79 when 10 and 50 ini-

tial labeled samples are used. For comment writer’s emotion classification, the performances of self-training are 0.505 and 0.508. These results are much lower than the performances of our co-training approach, especially on the comment writer’s emotion classification i.e., 0.505 and 0.508 vs. 0.783 and 0.805.

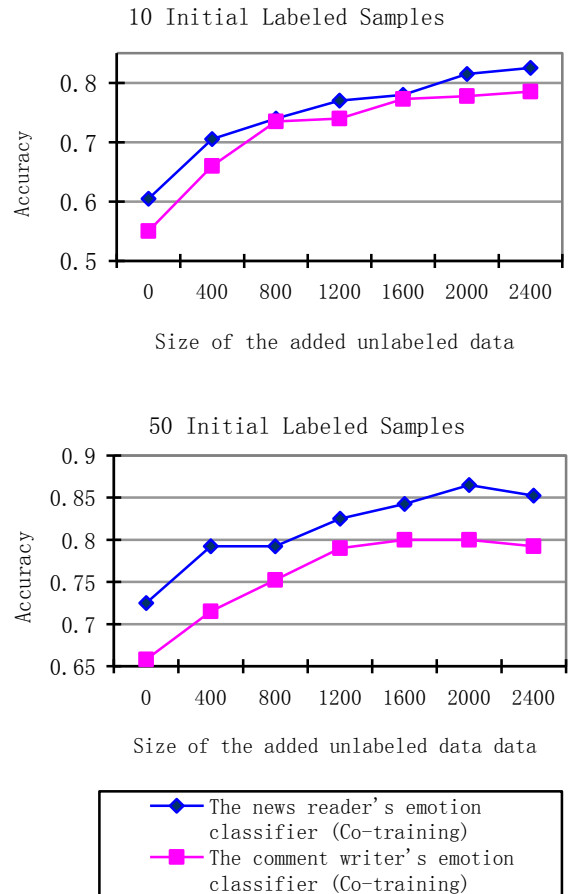


Figure 3: Performances of the news reader’s and comment writer’s emotion classifiers using the co-training algorithm

## 6 Conclusion

In this paper, we focus on two popular emotion classification tasks, i.e., reader’s emotion classification on the news and writer’s emotion classification on the comments. From the data analysis, we find that the news reader’s and comment writer’s emotions are highly consistent to each other in terms of the coarse-grained emotion categories, *positive* and *negative*. On the basis, we propose a co-training approach to perform semi-supervised learning on the two tasks. Evaluation shows that the co-training approach is so effective that using only 10 labeled samples achieves nice performances on both news reader’s and comment writer’s emotion classification.

\* <http://mallet.cs.umass.edu/>

## Acknowledgments

This research work has been partially supported by two NSFC grants, No.61003155, and No.61273320, one National High-tech Research and Development Program of China No.2012AA011102, one General Research Fund (GRF) sponsored by the Research Grants Council of Hong Kong No.543810, the NSF grant of Zhejiang Province No.Z1110551, and one project supported by Zhejiang Provincial Natural Science Foundation of China, No.Y13F020030.

## References

- Alm C., D. Roth and R. Sproat. 2005. Emotions from Text: Machine Learning for Text-based Emotion Prediction. In *Proceedings of EMNLP-05*, pp.579-586.
- Aman S. and S. Szpakowicz. 2008. Using Roget's Thesaurus for Fine-grained Emotion Recognition. In *Proceedings of IJCNLP-08*, pp.312-318.
- Chen Y., S. Lee, S. Li and C. Huang. 2010. Emotion Cause Detection with Linguistic Constructions. In *Proceeding of COLING-10*, pp.179-187.
- Cohen J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37-46.
- Cui H., V. Mittal and M. Datar. 2006. Comparative Experiments on Sentiment Classification for Online Product Comments. In *Proceedings of AAAI-06*, pp.1265-1270.
- Das D. and S. Bandyopadhyay. 2009. Word to Sentence Level Emotion Tagging for Bengali Blogs. In *Proceedings of ACL-09*, pp.149-152.
- Dasgupta S. and V. Ng. 2009. Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In *Proceedings of ACL-IJCNLP-09*, pp.701-709, 2009.
- Duin R. 2002. The Combining Classifier: To Train Or Not To Train? In *Proceedings of 16th International Conference on Pattern Recognition (ICPR-02)*.
- Fumera G. and F. Roli. 2005. A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems. *IEEE Trans. PAMI*, vol.27, pp.942-956, 2005.
- Li S., Z. Wang, G. Zhou and S. Lee. 2011. Semi-supervised Learning for Imbalanced Sentiment Classification. In *Proceeding of IJCAI-11*, pp.826-1831.
- Li S., C. Huang, G. Zhou and S. Lee. 2010. Employing Personal/Impersonal Views in Supervised and Semi-supervised Sentiment Classification. In *Proceedings of ACL-10*, pp.414-423.
- Lin K., C. Yang and H. Chen. 2007. What Emotions do News Articles Trigger in Their Readers? In *Proceeding of SIGIR-07*, poster, pp.733-734.
- Lin K., C. Yang and H. Chen. 2008. Emotion Classification of Online News Articles from the Reader's Perspective. In *Proceeding of the International Conference on Web Intelligence and Intelligent Agent Technology*, pp.220-226.
- Liu B. 2012. *Sentiment Analysis and Opinion Mining (Introduction and Survey)*. Morgan & Claypool Publishers, May 2012.
- Kittler J., M. Hatef, R. Duin, and J. Matas. 1998. On Combining Classifiers. *IEEE Trans. PAMI*, vol.20, pp.226-239, 1998
- Moshfeghi Y., B. Piwowarski and J. Jose. 2011. Handling Data Sparsity in Collaborative Filtering using Emotion and Semantic Based Features. In *Proceedings of SIGIR-11*, pp.625-634.
- Pang B. and L. Lee. 2008. Opinion Mining and Sentiment Analysis: Foundations and Trends. *Information Retrieval*, vol.2(12), 1-135.
- Pang B., L. Lee and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of EMNLP-02*, pp.79-86.
- Purver M. and S. Battersby. 2012. Experimenting with Distant Supervision for Emotion Classification. In *Proceedings of EACL-12*, pp.482-491.
- Quan C. and F. Ren. 2009. Construction of a Blog Emotion Corpus for Chinese Emotional Expression Analysis. In *Proceedings of EMNLP-09*, pp.1446-1454.
- Riloff E., S. Patwardhan and J. Wiebe. 2006. Feature Subsumption for Opinion Analysis. In *Proceedings of EMNLP-06*, pp.440-448.
- Turney P. 2002. Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of comments. In *Proceedings of ACL-02*, pp.417-424.
- Vilalta R. and Y. Drissi. 2002. A Perspective View and Survey of Meta-learning. *Artificial Intelligence Review*, 18(2): 77-95.
- Volkova S., W. Dolan and T. Wilson. 2012. CLex: A Lexicon for Exploring Color, Concept and Emotion Associations in Language. In *Proceedings of EACL-12*, pp.306-314.
- Wilson T., J. Wiebe, and P. Hoffmann. 2009. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics*, vol.35(3), pp.399-433.
- Xu G., X. Meng and H. Wang. 2010. Build Chinese Emotion Lexicons Using A Graph-based Algorithm and Multiple Resources. In *Proceeding of COLING-10*, pp.1209-1217.

# An Annotated Corpus of Quoted Opinions in News Articles

Tim O’Keefe James R. Curran Peter Ashwell Irena Koprinska

æ-lab, School of Information Technologies

University of Sydney

NSW 2006, Australia

{tokeefe, james, pash4408, irena}@it.usyd.edu.au

## Abstract

Quotes are used in news articles as evidence of a person’s opinion, and thus are a useful target for opinion mining. However, labelling each quote with a polarity score directed at a textually-anchored target can ignore the broader issue that the speaker is commenting on. We address this by instead labelling quotes as *supporting* or *opposing* a clear expression of a point of view on a topic, called a *position statement*. Using this we construct a corpus covering 7 topics with 2,228 quotes.

## 1 Introduction

News articles are a useful target for opinion mining as they discuss salient opinions by newsworthy people. Rather than asserting what a person’s opinion is, journalists typically provide evidence by using reported speech, and in particular, direct quotes. We focus on direct quotes as expressions of opinion, as they can be accurately extracted and attributed to a speaker (O’Keefe et al., 2012).

Characterising the opinions in quotes remains challenging. In sentiment analysis over product reviews, polarity labels are commonly used because the target, the product, is clearly identified. However, for quotes on topics of debate, the target and meaning of polarity labels is less clear. For example, labelling a quote about abortion as simply positive or negative is uninformative, as a speaker can use either positive or negative language to support or oppose either side of the debate.

Previous work (Wiebe et al., 2005; Balahur et al., 2010) has addressed this by giving each expression of opinion a textually-anchored target. While this makes sense for named entities, it does not apply as obviously for topics, such as abortion, that may not be directly mentioned. Our solution is to instead define *position statements*, which are

**Abortion:** Women should have the right to choose an abortion.

**Carbon tax:** Australia should introduce a tax on carbon or an emissions trading scheme to combat global warming.

**Immigration:** Immigration into Australia should be maintained or increased because its benefits outweigh any negatives.

**Reconciliation:** The Australian government should formally apologise to the Aboriginal people for past injustices.

**Republic:** Australia should cease to be a monarchy with the Queen as head of state and become a republic with an Australian head of state.

**Same-sex marriage:** Same-sex couples should have the right to attain the legal state of marriage as it is for heterosexual couples.

**Work choices:** Australia should introduce WorkChoices to give employers more control over wages and conditions.

Table 1: Topics and their position statements.

clear statements of a viewpoint or position on a particular topic. Quotes related to this topic can then be labelled as *supporting*, *neutral*, or *opposing* the position statement. This disambiguates the meaning of the polarity labels, and allows us to determine the side of the debate that the speaker is on. Table 1 shows the topics and position statements used in this work, and some example quotes from the republic topic are given below. Note that the first example includes no explicit mention of the monarchy or the republic.

**Positive:** “I now believe that the time has come... for us to have a truly Australian constitutional head of state.”

**Neutral:** “The establishment of an Australian republic is essentially a symbolic change, with the main arguments, for and against, turning on national identity...”

**Negative:** “I personally think that the monarchy is a tradition which we want to keep.”

With this formulation we define an annotation scheme and build a corpus covering 7 topics, with 100 documents per topic. This corpus includes 3,428 quotes, of which 1,183 were marked invalid, leaving 2,228 that were marked as *supporting*, *neutral*, or *opposing* the relevant topic statement. All quotes in our corpus were annotated by three annotators, with Fleiss’  $\kappa$  values of between 0.43 and 0.45, which is moderate.

## 2 Background

Early work in sentiment analysis (Turney, 2002; Pang et al., 2002; Dave et al., 2003; Blitzer et al., 2007) focused on product and movie reviews, where the text under analysis discusses a single product or movie. In these cases, labels like *positive* and *negative* are appropriate as they align well with the overall communicative goal of the text.

Later work established *aspect-oriented opinion mining* (Hu and Liu, 2004), where the aim is to find features or *aspects* of products that are discussed in a review. The reviewer’s position on each aspect can then be classified as *positive* or *negative*, which results in a more fine-grained classification that can be combined to form an *opinion summary*. These approaches assume that each document has a single source (the document’s author), whose communicative goal is to evaluate a well-defined target, such as a product or a movie. However this does not hold in news articles, where the goal of the journalist is to present the viewpoints of potentially many people.

Several studies (Wiebe et al., 2005; Wilson et al., 2005; Kim and Hovy, 2006; Godbole et al., 2007) have looked at sentiment in news text, with some (Balahur and Steinberger, 2009; Balahur et al., 2009, 2010) focusing on quotes. In all of these studies the authors have textually-anchored the target of the sentiment. While this makes sense for targets that can be resolved back to named entities, it does not apply as obviously when the quote is arguing for a particular viewpoint in a debate, as the topic may not be mentioned explicitly and polarity labels may not align to sides of the debate.

Work on debate summarisation and subgroup detection (Somasundaran and Wiebe, 2010; Abu-Jbara et al., 2012; Hassan et al., 2012) has often used data from online debate forums, particularly those forums where users are asked to select whether they support or oppose a given proposition before they can participate. This is similar to our aim with news text, where instead of a textually-anchored target, we have a proposition, against which we can evaluate quotes.

## 3 Position Statements

Our goal in this study is to determine which side of a debate a given quote supports. Assigning polarity labels to a textually-anchored target does not work here for several reasons. Quotes may not mention the debate topic, there may be many rel-

| Topic        | Quotes | No cont. |          | Context |          |
|--------------|--------|----------|----------|---------|----------|
|              |        | AA       | $\kappa$ | AA      | $\kappa$ |
| Abortion     | 343    | .77      | .57      | .73     | .53      |
| Carbon tax   | 278    | .71      | .42      | .57     | .34      |
| Immigration  | 249    | .58      | .18      | .58     | .25      |
| Reconcil.    | 513    | .66      | .37      | .68     | .44      |
| Republic     | 347    | .68      | .51      | .71     | .58      |
| Same-sex m.  | 246    | .72      | .51      | .71     | .55      |
| Work choices | 269    | .72      | .45      | .65     | .44      |
| <b>Total</b> | 2,245  | .69      | .43      | .66     | .45      |

Table 2: Average Agreement (AA) and Fleiss’  $\kappa$  over the valid quotes

evant textually-anchored targets for a single topic, and polarity labels do not necessarily align with sides of a debate.

We instead define *position statements*, which clearly state the position that one side of the debate is arguing for. We can then characterise opinions as *supporting*, *neutral* towards, or *opposing* this particular position. Position statements should not argue for a particular position, rather they should simply state what the position is. Table 1 shows the position statements that we use in this work.

## 4 Annotation

For our task we expect a set of news articles on a given topic as input, where the direct quotes in the articles have been extracted and attributed to speakers. A position statement will have been defined, that states a point of view on the topic, and a small subset of quotes will have been labelled as *supporting*, *neutral*, or *opposing* the given statement. A system performing this task would then label the remaining quotes as *supporting*, *neutral*, or *opposing*, and return them to the user.

A major contribution of this work is that we construct a fully labelled corpus, which can be used to evaluate systems that perform the task described above. To build this corpus we employed three annotators, one of whom is an author, while the other two were hired using the outsourcing website Freelancer<sup>1</sup>. Our data is drawn from the Sydney Morning Herald<sup>2</sup> archive, which ranges from 1986 until 2009, and it covers seven topics that were subject to debate within Australian news media during that time. For each topic we used

<sup>1</sup><http://www.freelancer.com>

<sup>2</sup><http://www.smh.com.au>

| Topic        | Quotes | No cont. |          | Context |          |
|--------------|--------|----------|----------|---------|----------|
|              |        | AA       | $\kappa$ | AA      | $\kappa$ |
| Abortion     | 343    | .78      | .52      | .74     | .46      |
| Carbon tax   | 278    | .72      | .39      | .59     | .19      |
| Immigration  | 249    | .58      | .08      | .58     | .14      |
| Reconcil.    | 513    | .66      | .31      | .69     | .36      |
| Republic     | 347    | .69      | .39      | .72     | .41      |
| Same-sex m.  | 246    | .73      | .43      | .73     | .40      |
| Work choices | 269    | .73      | .40      | .67     | .32      |
| <b>Total</b> | 2,245  | .70      | .36      | .68     | .32      |

Table 3: Average Agreement (AA) and Fleiss’  $\kappa$  when the labels are neutral versus non-neutral

Apache Solr<sup>3</sup> to find the top 100 documents that matched a manually-constructed search query. All documents were tokenised and POS-tagged and the named entities were found using the system from Hachey et al. (2013). Finally, the quotes were extracted and attributed to speakers using the system from O’Keefe et al. (2012).

For the first part of the task, annotators were asked to label each quote without considering any context. In other words they were asked to only use the text of the quote itself as evidence for an opinion, not the speaker’s prior opinions or the text of the document. They were then asked to label the quote a second time, while considering the text surrounding the quote, although they were still asked to ignore the prior opinions of the speaker. For each of these choices annotators were given a five-point scale ranging from *strong or clear opposition* to *strong or clear support*, where *support* or *opposition* is relative to the position statement.

Annotators were also asked to mark instances where either the speaker or quote span was incorrectly identified, although they were asked to continue annotating the quote as though it were correct. They were also asked to mark quotes that were invalid due to either the quote being off-topic, or the item not being a quote (e.g. book titles, scare quotes, etc.).

## 5 Corpus results

In order to achieve the least amount of noise in our corpus, we opted to discard quotes that any annotator had marked as invalid. From the original set of 3,428 quotes, 1,183 (35%) were removed, which leaves 2,245 (65%). From the original corpus, 23% were marked off-topic, which shows that

<sup>3</sup><http://lucene.apache.org/solr/>

in order to label opinions in news, a system would first have to identify the topic-relevant parts of the text. The annotators further indicated that 16% were not quotes, and there were a small number of cases (<1%) where the quote span was incorrect. Annotators were able to select multiple reasons for a quote being invalid.

Table 2 shows both Fleiss’  $\kappa$  and the raw agreement averaged between annotators for each topic. We collapsed the two supporting labels together, as well as the two opposing labels, such that we end up with a classification of *opposes* vs. *neutral* vs. *supports*. The no context and context cases scored 0.69 and 0.66 in raw agreement, while the  $\kappa$  values were 0.43 and 0.45, which is moderate.

Intuitively we expect that the confusion is largely between neutral and the two polar labels. To examine this we merged all the non-neutral labels into one group and calculated the agreement between the non-neutral group and the neutral label, as shown in Table 3. For the non-neutral vs. neutral agreement we find that despite stability in raw agreement, Fleiss’  $\kappa$  drops substantially, to 0.36 (no context) and 0.32 (context).

For comparison we remove all neutral annotations and focus on disagreement between the polar labels. For this we cannot use Fleiss’  $\kappa$ , as it requires a fixed number of annotations per quote, however we can average the pairwise  $\kappa$  values between annotators, which results in values of 0.93 (no context) and 0.92 (context). Though they are not directly comparable, the magnitude of the difference between the numbers (0.36 and 0.32 vs. 0.93 and 0.92) indicates that deciding when an opinion provides sufficient evidence of support or opposition is the main challenge facing annotators.

To adjudicate the decisions annotators made, we opted to take a majority vote for cases of two or three-way agreement, while discarding cases where annotators did not agree (1% of quotes). The final distribution of labels in the corpus is shown in Table 4. For both the no context and context cases the largest class was neutral with 61% and 46% of the corpus respectively. The drop in neutrality between the no context and context cases shows that the interpretation of a quote can change based on the context it is placed in.

## 6 Discussion

In refining our annotation scheme we noted several factors that make annotation difficult.

| Topic             | No context |      |       |       | Context |      |       |       |
|-------------------|------------|------|-------|-------|---------|------|-------|-------|
|                   | Quotes     | Opp. | Neut. | Supp. | Quotes  | Opp. | Neut. | Supp. |
| Abortion          | 343        | .13  | .63   | .25   | 340     | .16  | .52   | .32   |
| Carbon tax        | 273        | .09  | .70   | .21   | 273     | .14  | .44   | .42   |
| Immigration       | 247        | .09  | .72   | .19   | 245     | .12  | .64   | .23   |
| Reconciliation    | 509        | .05  | .57   | .38   | 503     | .07  | .42   | .50   |
| Republic          | 345        | .24  | .48   | .28   | 342     | .32  | .37   | .32   |
| Same-sex marriage | 246        | .16  | .55   | .28   | 243     | .25  | .38   | .37   |
| Work choices      | 265        | .14  | .72   | .14   | 266     | .26  | .50   | .24   |
| <b>Total</b>      | 2,228      | .12  | .61   | .26   | 2,212   | .18  | .46   | .36   |

Table 4: Label distribution for the final corpus.

**Opinion relevance** When discussing a topic, journalists will often delve into the related aspects and opinions that people hold. This introduces a challenge as annotators need to decide whether a particular quote is on-topic enough to be labelled. For instance, these quotes by the same speaker were in an article on the carbon tax:

- 1) “Whether it’s a stealth tax, the emissions trading scheme, whether it’s an upfront. . . tax like a carbon tax, there will not be any new taxes as part of the Coalition’s policy”
- 2) “I don’t think it’s something that we should rush into. But certainly I’m happy to see a debate about the nuclear option.”

In the first quote the speaker is voicing opposition to a tax on carbon, which is easy to annotate with our scheme. However in the second quote, the speaker is discussing nuclear power in relation to a carbon tax, which is much more difficult, as it is unclear whether it is *off-topic* or *neutral*.

**Obfuscation and self-contradiction** While journalists usually quote someone to provide evidence of the person’s opinion, there are some cases where they include quotes to show that the person is inconsistent. The following quotes by the same speaker were included in an article to illustrate that the speaker’s position was inconsistent:

- 1) “My point is that. . . the most potent argument in favour of the republic, is that why should we have a Briton as the Queen – who, of course, in reality is also the Queen of Australia – but a Briton as the head of State of Australia”
- 2) “The Coalition supports the Constitution not because we support the. . . notion of the monarchy, but because we support the way our present Constitution works”

The above example also indicates a level of obfuscation that is reasonably common for politicians. Neither of the quotes actually expresses a clear statement of how the speaker feels about a potential republic. The first quote is an opinion

about the strongest argument in favour of a republic, without necessarily making that argument, while the second quote states a party line, with a caveat that might indicate personal disagreement.

**Annotator bias** This task is prone to be influenced by an annotator’s biases, including their political or cultural background, their opinion about the topic or speaker, or their level of knowledge about the topic.

## 7 Conclusion

In this work we examined the problem of annotating opinions in news articles. We proposed to exploit quotes, as they are used by journalists to provide evidence of an opinion, and are easy to extract and attribute to speakers. Our key contribution is that rather than requiring a textually-anchored target for each quote, we instead label quotes as *supporting*, *neutral*, or *opposing* a position statement, which states a particular viewpoint on a topic. This allowed us to resolve ambiguities that arise when considering a polarity label towards a topic. We next defined an annotation scheme and built a corpus, which covers 7 topics, with 100 documents per topic, and a total of 2,228 annotated quotes. Future work will include building a system able to perform the task we have defined, as well as extending this work to include indirect quotes.

## Acknowledgements

O’Keefe was supported by a University of Sydney Merit scholarship and a Capital Markets CRC top-up scholarship. This work was supported by ARC Discovery grant DP1097291 and the Capital Markets CRC Computable News project.

## References

- Amjad Abu-Jbara, Mona Diab, Pradeep Dasigi, and Dragomir Radev. 2012. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 399–409.
- Alexandra Balahur and Ralf Steinberger. 2009. Rethinking sentiment analysis in the news: From theory to practice and back. *Proceedings of the First Workshop on Opinion Mining and Sentiment Analysis*.
- Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2010. Sentiment analysis in the news. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 2216–2220.
- Alexandra Balahur, Ralf Steinberger, Erik Van Der Goot, Bruno Pouliquen, and Mijail Kabadjov. 2009. Opinion mining on newspaper quotations. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 523–526.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447.
- Kushal Dave, Steve Lawrence, and David 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*, pages 519–528.
- Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media*.
- Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial Intelligence*.
- Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 59–70.
- 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.
- Soo-Min Kim and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8.
- Tim O’Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 790–799.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.
- Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124.
- Peter 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*, pages 417–424.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2/3):165–210.
- Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. OpinionFinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP Interactive Demonstrations*.



# Dual Training and Dual Prediction for Polarity Classification

**Rui Xia, Tao Wang, Xuelei Hu**

Department of Computer Science  
Nanjing University of  
Science and Technology  
rxia@njust.edu.cn,  
linclonwang@163.com,  
xlhu@njust.edu.cn

**Shoushan Li**

NLP Lab  
Department of  
Computer Science  
Soochow University  
shoushan.li  
@gmail.com

**Chengqing Zong**

National Lab of  
Pattern Recognition  
Institute of Automation  
CAS  
cqzong  
@nlpr.ia.ac.cn

## Abstract

Bag-of-words (BOW) is now the most popular way to model text in machine learning based sentiment classification. However, the performance of such approach sometimes remains rather limited due to some fundamental deficiencies of the BOW model. In this paper, we focus on the polarity shift problem, and propose a novel approach, called dual training and dual prediction (DTDP), to address it. The basic idea of DTDP is to first generate artificial samples that are polarity-opposite to the original samples by polarity reversion, and then leverage both the original and opposite samples for (dual) training and (dual) prediction. Experimental results on four datasets demonstrate the effectiveness of the proposed approach for polarity classification.

## 1 Introduction

The most popular text representation model in machine learning based sentiment classification is known as the bag-of-words (BOW) model, where a piece of text is represented by an unordered collection of words, based on which standard machine learning algorithms are employed as classifiers. Although the BOW model is simple and has achieved great successes in topic-based text classification, it disrupts word order, breaks the syntactic structures and discards some kinds of semantic information that are possibly very important for sentiment classification. Such disadvantages sometimes limit the performance of sentiment classification systems.

A lot of subsequent work focused on feature engineering that aims to find a set of effective features based on the BOW representation. However, there still remain some problems that are not well addressed. Out of them, the polarity shift problem is the biggest one.

We refer to “polarity shift” as a linguistic phenomenon that the sentiment orientation of a text is reversed (from positive to negative or vice versa) because of some particular expressions called polarity shifters. Negation words (e.g., “no”, “not” and “don’t”) are the most important type of polarity shifter. For example, by adding a negation word “don’t” to a positive text “I like this book” in front of “like”, the orientation of the text is reversed from positive to negative.

Naturally, handling polarity shift is very important for sentiment classification. However, the BOW representations of two polarity-opposite texts, e.g., “*I like this book*” and “*I don’t like this book*”, are considered to be very similar by most of machine learning algorithms. Although some methods have been proposed in the literature to address the polarity shift problem (Das and Chen, 2001; Pang et al., 2002; Na et al., 2004; Kenndey and Inkpen, 2006; Ikeda et al., 2008; Li and Huang, 2009; Li et al., 2010), the state-of-the-art results are still far from satisfactory. For example, the improvements are less than 2% after considering polarity shift in Li et al. (2010).

In this work, we propose a novel approach, called dual training and dual prediction (DTDP), to address the polarity shift problem. By taking advantage of the unique nature of polarity classification, DTDP is motivated by first generating artificial samples that are polarity-opposite to the original ones. For example, given the original sample “*I don’t like this book. It is boring,*” its polarity-opposite version, “*I like this book. It is interesting*”, is artificially generated. Second, the original and opposite training samples are used together for training a sentiment classifier (called dual training), and the original and opposite test samples are used together for prediction (called dual prediction). Experimental results prove that the procedure of DTDP is very effective at correcting the training and prediction errors caused

by polarity shift, and it beats other alternative methods of considering polarity shift.

## 2 Related Work

The lexicon-based sentiment classification systems can be easily modified to include polarity shift. One common way is to directly reverse the sentiment orientation of polarity-shifted words, and then sum up the orientations word by word (Hu and Liu, 2004; Kim and Hovy, 2004; Polanyi and Zaenen, 2004; Kennedy and Inkpen, 2006). Wilson et al. (2005) discussed other complex negation effects by using conjunctive and dependency relations among polarity words. Although handling polarity shift is easy and effective in term-counting systems, they rarely outperform the baselines of machine learning methods (Kennedy, 2006).

The machine learning methods are generally more effective for sentiment classification. However, it is difficult to handle polarity shift based on the BOW model. Das and Chen (2001) proposed a method by simply attaching “NOT” to words in the scope of negation, so that in the text “*I don’t like book*”, the word “*like*” is changed to a new word “*like-NOT*”. There were also some attempts to model polarity shift by using more complex linguistic features (Na et al., 2004; Kennedy and Inkpen, 2006). But the improvements upon the baselines of machine learning systems are very slight (less than 1%).

Ikeda et al. (2008) proposed a machine learning method, to model polarity-shifters for both word-wise and sentence-wise sentiment classification, based on a dictionary extracted from General Inquirer. Li and Huang (2009) proposed a method first to classify each sentence in a text into a polarity-unshifted part and a polarity-shifted part according to certain rules, then to represent them as two bag-of-words for sentiment classification. Li et al. (2010) further proposed a method to separate the shifted and unshifted text based on training a binary detector. Classification models are then trained based on each of the two parts. An ensemble of two component parts is used at last to get the final polarity of the whole text.

## 3 The Proposed Approach

We first present the method for generating artificial polarity-opposite samples, and then introduce the algorithm of dual training and dual prediction (DTDP).

### 3.1 Generating Artificial Polarity-Opposite Samples

Given an original sample and an antonym dictionary (e.g., WordNet<sup>1</sup>), a polarity-opposite sample is generated artificially according to the following rules:

- 1) **Sentiment word reversion:** All sentiment words out of the scope of negation are reversed to their antonyms;
- 2) **Handling negation:** If there is a negation expression, we first detect the scope of negation, and then remove the negation words (e.g., “no”, “not”, and “don’t”). The sentiment words in the scope of negation are not reversed;
- 3) **Label reversion:** The class label of the labeled sample is also reversed to its opposite (i.e., Positive to Negative, or vice versa) as the class label of newly generated samples (called polarity-opposite samples).

Let us use a simple example to explain the generation process. Given the original sample:

The original sample

Text: *I don’t like this book. It is boring.*

Label: Negative

According to Rule 1, “*boring*” is reversed to its antonym “*interesting*”; According to Rule 2, the negation word “*don’t*” is removed, and “*like*” is not reversed; According to Rule 3, the class label Negative is reversed to Positive. Finally, an artificial polarity-opposite sample is generated:

The generated opposite sample

Text: *I like this book. It is interesting.*

Label: Positive

All samples in the training and test set are reversed to their polarity-opposite versions. We refer to them as “opposite training set” and “opposite test set”, respectively.

### 3.2 Dual Training and Dual Prediction

In this part, we introduce how to make use of the original and opposite training/test data together for dual training and dual prediction (DTDP).

**Dual Training:** Let  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$  and  $\tilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^N$  be the original and opposite training set respectively, where  $x$  denotes the feature vector,  $y$  denotes the class label, and  $N$  denotes the size of training set. In dual training,  $\mathcal{D} \cup \tilde{\mathcal{D}}$  are used together as training data to learn

---

<sup>1</sup> <http://wordnet.princeton.edu/>

a classification model. The size of training data is doubled in dual training.

Suppose the example in Section 3.1 is used as one training sample. As far as only the original sample (“*I don’t like this book. It is boring.*”) is considered, the feature “*like*” will be improperly recognized as a negative indicator (since the class label is Negative), ignoring the expression of negation. Nevertheless, if the generated opposite sample (“*I like this book. It is interesting.*”) is also used for training, “*like*” will be learned correctly, due to the removal of negation in sample reversion. Therefore, the procedure of dual training can correct some learning errors caused by polarity shift.

**Dual Prediction:** Given an already-trained classification model, in dual prediction, the original and opposite test samples are used together for prediction. In dual prediction, when we predict the positive degree of a test sample, we measure not only how positive the original test sample is, but also how negative the opposite sample is.

Let  $x$  and  $\tilde{x}$  denote the feature vector of the original and opposite test samples respectively; let  $p_d(c|x)$  and  $p_d(c|\tilde{x})$  denote the predictions of the original and opposite test sample, based on the dual training model. The dual predicting function is defined as:

$$p_d(+|x, \tilde{x}) = (1 - a)p_d(+|x) + ap_d(-|\tilde{x}),$$

$$p_d(-|x, \tilde{x}) = (1 - a)p_d(-|x) + ap_d(+|\tilde{x}),$$

where  $a$  ( $0 \leq a \leq 1$ ) is the weight of the opposite prediction.

Now suppose the example in Section 3.1 is a test sample. As far as only the original test sample (“*I don’t like this book. It is boring.*”) is used for prediction, it is very likely that it is falsely predicted as Positive, since “*like*” is a strong positive feature, despite that it is in the scope of negation. While in dual prediction, we still measure the “sentiment-opposite” degree of the opposite test sample (“*I like this book. It is interesting.*”). Since negation is removed, it is very likely that the opposite test sample is assigned with a high positive score, which could compensate the prediction errors of the original test sample.

**Final Output:** It should be noted that although the artificially generated training and testing data are helpful in most cases, they still produce some noises (e.g., some poorly generated samples may violate the quality of the original data set). Therefore, instead of using all dual predictions as the final output, we use the origi-

nal prediction  $p_o(c|x)$  as an alternate, in case that the dual prediction  $p_d(c|x, \tilde{x})$  is not enough confident, according to a confidence threshold  $t$ . The final output is defined as:

$$p_f(c|x) = \begin{cases} p_d(c|x, \tilde{x}), & \text{if } \Delta p \geq t \\ p_o(c|x), & \text{if } \Delta p < t \end{cases}$$

where  $\Delta p = p_d(c|x, \tilde{x}) - p_o(c|x)$ .

## 4 Experimental Study

### 4.1 Datasets

The Multi-Domain Sentiment Datasets<sup>2</sup> are used for evaluations. They consist of product reviews collected from four different domains: Book, DVD, Electronics and Kitchen. Each of them contains 1,000 positive and 1,000 negative reviews. Each of the datasets is randomly split into 5 folds, with four folds serving as training data, and the remaining one fold serving as test data. All of the following results are reported in terms of an average of 5-fold cross validation.

### 4.2 Evaluated Systems

We evaluate four machine learning systems that are proposed to address polarity shift in document-level polarity classification:

- 1) **Baseline:** standard machine learning methods based on the BOW model, without handling polarity shift;
- 2) **Das-2001:** the method proposed by Das and Chen (2001), where “NOT” is attached to the words in the scope of negation as a preprocessing step;
- 3) **Li-2010:** the approach proposed by Li et al. (2010). The details of the algorithm is introduced in related work;
- 4) **DTDP:** our approach proposed in Section 3. The WordNet dictionary is used for sample reversion. The empirical value of the parameter  $a$  and  $t$  are used in the evaluation.

### 4.3 Comparison of the Evaluated Systems

In table 1, we report the classification accuracy of four evaluated systems using unigram features. We consider two widely-used classification algorithms: SVM and Naïve Bayes. For SVM, the LibSVM toolkit<sup>3</sup> is used with a linear kernel and the default penalty parameter. For Naïve Bayes, the OpenPR-NB toolkit<sup>4</sup> is used.

<sup>2</sup> <http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

<sup>3</sup> <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>4</sup> <http://www.openpr.org.cn>

| Dataset     | SVM      |          |         |              | Naïve Bayes |          |         |              |
|-------------|----------|----------|---------|--------------|-------------|----------|---------|--------------|
|             | Baseline | Das-2001 | Li-2010 | DTDP         | Baseline    | Das-2001 | Li-2010 | DTDP         |
| Book        | 0.745    | 0.763    | 0.760   | <b>0.800</b> | 0.779       | 0.783    | 0.792   | <b>0.814</b> |
| DVD         | 0.764    | 0.771    | 0.795   | <b>0.823</b> | 0.795       | 0.793    | 0.810   | <b>0.820</b> |
| Electronics | 0.796    | 0.813    | 0.812   | <b>0.828</b> | 0.815       | 0.827    | 0.824   | <b>0.841</b> |
| Kitchen     | 0.822    | 0.820    | 0.844   | <b>0.849</b> | 0.830       | 0.847    | 0.840   | <b>0.859</b> |
| Avg.        | 0.782    | 0.792    | 0.803   | <b>0.825</b> | 0.804       | 0.813    | 0.817   | <b>0.834</b> |

Table 1: Classification accuracy of different systems using unigram features

| Dataset     | SVM      |          |         |              | Naïve Bayes |          |         |              |
|-------------|----------|----------|---------|--------------|-------------|----------|---------|--------------|
|             | Baseline | Das-2001 | Li-2010 | DTDP         | Baseline    | Das-2001 | Li-2010 | DTDP         |
| Book        | 0.775    | 0.777    | 0.788   | <b>0.818</b> | 0.811       | 0.815    | 0.822   | <b>0.840</b> |
| DVD         | 0.790    | 0.793    | 0.809   | <b>0.828</b> | 0.824       | 0.826    | 0.837   | <b>0.868</b> |
| Electronics | 0.818    | 0.834    | 0.841   | <b>0.848</b> | 0.841       | 0.857    | 0.852   | <b>0.866</b> |
| Kitchen     | 0.847    | 0.844    | 0.870   | <b>0.878</b> | 0.878       | 0.879    | 0.883   | <b>0.896</b> |
| Avg.        | 0.808    | 0.812    | 0.827   | <b>0.843</b> | 0.839       | 0.844    | 0.849   | <b>0.868</b> |

Table 2: Classification accuracy of different systems using both unigram and bigram features

Compared to the Baseline system, the Das-2001 approach achieves very slight improvements (less than 1%). The performance of Li-2010 is relatively effective: it improves the average score by 0.21% and 0.13% on SVM and Naïve Bayes, respectively. Yet, the improvements are still not satisfactory.

As for our approach (DTDP), the improvements are remarkable. Compared to the Baseline system, the average improvements are 4.3% and 3.0% on SVM and Naïve Bayes, respectively. In comparison with the state-of-the-art (Li-2010), the average improvement is 2.2% and 1.7% on SVM and Naïve Bayes, respectively.

We also report the classification accuracy of four systems using both unigrams and bigrams features for classification in Table 2. From this table, we can see that the performance of each system is improved compared to that using unigrams. It is now relatively difficult to show improvements by incorporating polarity shift, because using bigrams already captured a part of negations (e.g., “don’t like”).

The Das-2001 approach still shows very limited improvements (less than 0.5%), which agrees with the reports in Pang et al. (2002). The improvements of Li-2010 are also reduced: 1.9% and 1% on SVM and Naïve Bayes, respectively.

Although the improvements of the previous two systems are both limited, the performance of our approach (DTDP) is still sound. It improves the Baseline system by 3.7% and 2.9% on SVM and Naïve Bayes, respectively, and outperforms the state-of-the-art (Li-2010) by 1.6% and 1.9% on SVM and Naïve Bayes, respectively.

## 5 Conclusions

In this work, we propose a method, called dual training and dual prediction (DTDP), to address the polarity shift problem in sentiment classification. The basic idea of DTDP is to generate artificial samples that are polarity-opposite to the original samples, and to make use of both the original and opposite samples for dual training and dual prediction. Experimental studies show that our DTDP algorithm is very effective for sentiment classification and it beats other alternative methods of considering polarity shift.

One limitation of current work is that the tuning of parameters in DTDP (such as  $a$  and  $t$ ) is not well discussed. We will leave this issue to an extended version.

## Acknowledgments

The research work is supported by the Jiangsu Provincial Natural Science Foundation of China (BK2012396), the Research Fund for the Doctoral Program of Higher Education of China (20123219120025), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). This work is also partly supported by the Hi-Tech Research and Development Program of China (2012AA011102 and 2012AA011101), the Program of Introducing Talents of Discipline to Universities (B13022), and the Open Project Program of the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (30920130122006).

## References

- S. Das and M. Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference*.
- M. Hu and B. Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.
- D. Ikeda, H. Takamura L. Ratinov M. Okumura. 2008. Learning to Shift the Polarity of Words for Sentiment Classification. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.
- S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceeding of the International Conference on Computational Linguistics (COLING)*.
- A. Kennedy and D. Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22:110–125.
- S. Li and C. Huang. 2009. Sentiment classification considering negation and contrast transition. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC)*.
- S. Li, S. Lee, Y. Chen, C. Huang and G. Zhou. 2010. Sentiment Classification and Polarity Shifting. In *Proceeding of the International Conference on Computational Linguistics (COLING)*.
- J. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. 2004. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Proceeding of the Conference of the International Society for Knowledge Organization*.
- B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- L. Polanyi and A. Zaenen. 2004. Contextual lexical valence shifters. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text, AAAI technical report*.
- P. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceeding of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

# Co-Regression for Cross-Language Review Rating Prediction

Xiaojun Wan

Institute of Computer Science and Technology, The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China

wanxiaojun@pku.edu.cn

## Abstract

The task of review rating prediction can be well addressed by using regression algorithms if there is a reliable training set of reviews with human ratings. In this paper, we aim to investigate a more challenging task of cross-language review rating prediction, which makes use of only rated reviews in a source language (e.g. English) to predict the rating scores of unrated reviews in a target language (e.g. German). We propose a new co-regression algorithm to address this task by leveraging unlabeled reviews. Evaluation results on several datasets show that our proposed co-regression algorithm can consistently improve the prediction results.

## 1 Introduction

With the development of e-commerce, more and more people like to buy products on the web and express their opinions about the products by writing reviews. These reviews usually contain valuable information for other people's reference when they buy the same or similar products. In some applications, it is useful to categorize a review into either positive or negative, but in many real-world scenarios, it is important to provide numerical ratings rather than binary decisions.

The task of review rating prediction aims to automatically predict the rating scores of unrated product reviews. It is considered as a finer-grained task than the binary sentiment classification task. Review rating prediction has been modeled as a multi-class classification or regression task, and the regression based methods have shown better performance than the multi-class classification based methods in recent studies (Li et al. 2011). Therefore, we focus on investigating regression-based methods in this study.

Traditionally, the review rating prediction task has been investigated in a monolingual setting, which means that the training reviews with human ratings and the test reviews are in the same language. However, a more challenging task is to

predict the rating scores of the reviews in a target language (e.g. German) by making use of the rated reviews in a different source language (e.g. English), which is called Cross-Language Review Rating Prediction. Considering that the resources (i.e. the rated reviews) for review rating prediction in different languages are imbalanced, it would be very useful to make use of the resources in resource-rich languages to help address the review rating prediction task in resource-poor languages.

The task of cross-language review rating prediction can be typically addressed by using machine translation services for review translation, and then applying regression methods based on the monolingual training and test sets. However, due to the poor quality of machine translation, the reviews translated from one language A to another language B are usually very different from the original reviews in language B, because the words or syntax of the translated reviews may be erroneous or non-native. This phenomenon brings great challenges for existing regression algorithms.

In this study, we propose a new co-regression algorithm to address the above problem by leveraging unlabeled reviews in the target language. Our algorithm can leverage both views of the reviews in the source language and the target language to collaboratively determine the confidently predicted ones out of the unlabeled reviews, and then use the selected examples to enlarge the training set. Evaluation results on several datasets show that our proposed co-regression algorithm can consistently improve the prediction results.

## 2 Related Work

Most previous works on review rating prediction model this problem as a multi-class classification task or a regression task. Various features have been exploited from the review text, including words, patterns, syntactic structure, and semantic topic (Qu et al. 2010; Pang and Lee, 2005; Leung et al. 2006; Ganu et al. 2009). Traditional learn-

ing models, such as SVM, are adopted for rating prediction. Most recently, Li et al. (2011) propose a novel tensor-based learning framework to incorporate reviewer and product information into the text based learner for rating prediction. Saggion et al. (2012) study the use of automatic text summaries instead of the full reviews for movie review rating prediction. In addition to predicting the overall rating of a full review, multi-aspect rating prediction has also been investigated (Lu et al. 2011b; Snyder and Barzilay, 2007; Zhu et al. 2009; Wang et al. 2010; Lu et al. 2009; Titov and McDonald, 2008). All the above previous works are working under a monolingual setting, and to the best of our knowledge, there exists no previous work on cross-language review rating prediction.

It is noteworthy that a few studies have been conducted for the task of cross-lingual sentiment classification or text classification, which aims to make use of labeled data in a language for the binary classification task in a different language (Mihalcea et al., 2007; Banea et al., 2008; Wan 2009; Lu et al. 2011a; Meng et al. 2012; Shi et al., 2010; Prettenhofer and Stein 2010). However, the binary classification task is very different from the regression task studied in this paper, and the proposed methods in the above previous works cannot be directly applied.

### 3 Problem Definition and Baseline Approaches

Let  $L = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\}$  denote the labeled training set of reviews in a source language (e.g. English), where  $x_i$  is the  $i$ -th review and  $y_i$  is its real-valued label, and  $n$  is the number of labeled examples; Let  $T$  denote the test review set in a different target language (e.g. German); Then the task of cross-language review rating prediction aims at automatically predicting the rating scores of the reviews in  $T$  by leveraging the labeled reviews in  $L$ . No labeled reviews in the target language are allowed to be used.

The task is a regression problem and it is challenging due to the language gap between the labeled training dataset and the test dataset. Fortunately, due to the development of machine translation techniques, a few online machine translation services can be used for review translation. We adopt *Google Translate*<sup>1</sup> for review translation. After review translation, the training reviews and the test reviews are now in the same

language, and any regression algorithm (e.g. logistic regression, least squares regression, KNN regressor) can be applied for learning and prediction. In this study, without loss of generality, we adopt the widely used regression SVM (Vapnik 1995; Joachims 1999) implemented in the SVMLight toolkit<sup>2</sup> as the basic regressor. For comparative analysis, we simply use the default parameter values in SVMLight with linear kernel. The features include all unigrams and bigrams in the review texts, and the value of each feature is simply set to its frequency (TF) in a review.

Using features in different languages, we have the following baseline approaches for addressing the cross-language regression problem.

**REG\_S:** It conducts regression learning and prediction in the source language.

**REG\_T:** It conducts regression learning and prediction in the target language.

**REG\_ST:** It conducts regression learning and prediction with all the features in both languages.

**REG\_STC:** It combines **REG\_S** and **REG\_T** by averaging their prediction values.

However, the above regression methods do not perform very well due to the unsatisfactory machine translation quality and the various language expressions. Therefore, we need to find new approaches to improve the above methods.

## 4 Our Proposed Approach

### 4.1 Overview

Our basic idea is to make use of some amounts of unlabeled reviews in the target language to improve the regression performance. Considering that the reviews have two views in two languages and inspired by the co-training style algorithms (Blum and Mitchell, 1998; Zhou and Li, 2005), we propose a new co-training style algorithm called co-regression to leverage the unlabeled data in a collaborative way. The proposed co-regression algorithm can make full use of both the features in the source language and the features in the target language in a unified framework similar to (Wan 2009). Each review has two versions in the two languages. The source-language features and the target-language features for each review are considered two redundant views of the review. In the training phase, the co-regression algorithm is applied to learn two regressors in the two languages. In the prediction phase, the two regressors are applied to predict two rating scores of the review. The

<sup>1</sup> <http://translate.google.com>

<sup>2</sup> <http://svmlight.joachims.org>

final rating score of the review is the average of the two rating scores.

## 4.2 Our Proposed Co-Regression Algorithm

In co-training for classification, some confidently classified examples by one classifier are provided for the other classifier, and vice versa. Each of the two classifiers can improve by learning from the newly labeled examples provided by the other classifier. The intuition is the same for co-regression. However, in the classification scenario, the confidence value of each prediction can be easily obtained through consulting the classifier. For example, the SVM classifier provides a confidence value or probability for each prediction. However, in the regression scenario, the confidence value of each prediction is not provided by the regressor. So the key question is how to get the confidence value of each labeled example. In (Zhou and Li, 2005), the assumption is that the most confidently labeled example of a regressor should be with such a property, i.e. the error of the regressor on the labeled example set (i.e. the training set) should decrease the most if the most confidently labeled example is utilized. In other words, the confidence value of each labeled example is measured by the decrease of the error (e.g. mean square error) on the labeled set of the regressor utilizing the information provided by the example. Thus, each example in the unlabeled set is required to be checked by training a new regression model utilizing the example. However, the model training process is usually very time-consuming for many regression algorithms, which significantly limits the use of the work in (Zhou and Li, 2005). Actually, in (Zhou and Li, 2005), only the lazy learning based KNN regressor is adopted. Moreover, the confidence of the labeled examples is assessed based only on the labeled example set (i.e. the training set), which makes the generalization ability of the regressor not good.

In order to address the above problem, we propose a new confidence evaluation strategy based on the consensus of the two regressors. Our intuition is that if the two regressors agree on the prediction scores of an example very well, then the example is very confidently labeled. On the contrary, if the prediction scores of an example by the two regressors are very different, we can hardly make a decision whether the example is confidently labeled or not. Therefore, we use the absolute difference value between the prediction scores of the two regressors as the confidence value of a labeled example, and if the ex-

ample is chosen, its final prediction score is the average of the two prediction scores. Based on this strategy, the confidently labeled examples can be easily and efficiently chosen from the unlabeled set as in the co-training algorithm, and these examples are then added into the labeled set for re-training the two regressors.

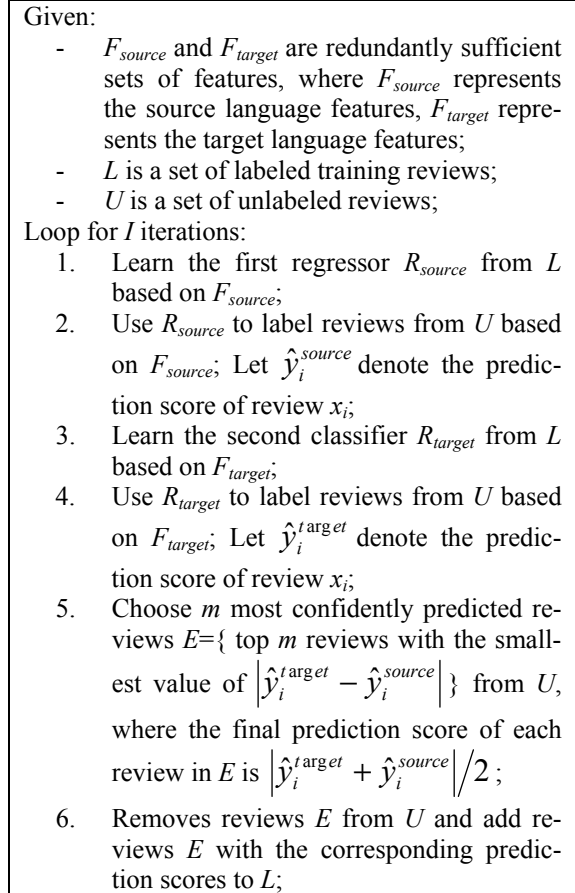


Figure 1. Our proposed co-regression algorithm

Our proposed co-regression algorithm is illustrated in Figure 1. In the proposed co-regression algorithm, any regression algorithm can be used as the basic regressor to construct  $R_{source}$  and  $R_{target}$ , and in this study, we adopt the same regression SVM implemented in the SVMlight toolkit with default parameter values. Similarly, the features include both unigrams and bigrams and the feature weight is simply set to term frequency. There are two parameters in the algorithm:  $I$  is the iteration number and  $m$  is the growth size in each iteration.  $I$  and  $m$  can be empirically set according to the total size of the unlabeled set  $U$ , and we have  $I \times m \leq |U|$ .

Our proposed co-regression algorithm is much more efficient than the COREG algorithm (Zhou and Li, 2005). If we consider the time-consuming regression learning process as one



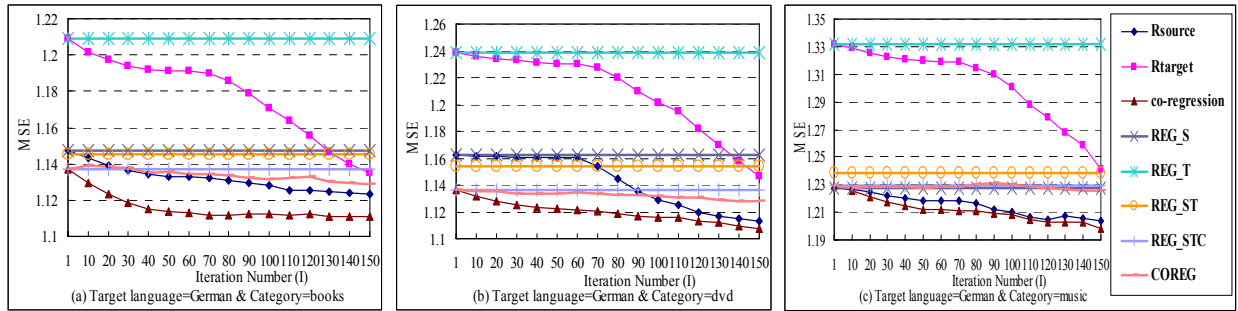


Figure 2. Comparison results vs. Iteration Number ( $I$ ) ( $R_{\text{source}}$  and  $R_{\text{target}}$  are the two component regressors)

basic operation and make use of all unlabeled examples in  $U$ , the computational complexity of COREG is  $O(|U|+I)$ . By contrast, the computational complexity of our proposed co-regression algorithm is just  $O(I)$ . Since  $|U|$  is much larger than  $I$ , our proposed co-regression algorithm is much more efficient than COREG, and thus our proposed co-regression algorithm is more suitable to be used in applications with a variety of regression algorithms.

Moreover, in our proposed co-regression algorithm, the confidence of each prediction is determined collaboratively by two regressors. The selection is not restricted by the training set, and it is very likely that a portion of good examples can be chosen for generalize the regressor towards the test set.

## 5 Empirical Evaluation

We used the WEBIS-CLS-10 corpus<sup>3</sup> provided by (Prettenhofer and Stein, 2010) for evaluation. It consists of Amazon product reviews for three product categories (i.e. books, dvds and music) written in different languages including English, German, etc. For each language-category pair there exist three sets of training documents, test documents, and unlabeled documents. The training and test sets comprise 2000 documents each, whereas the number of unlabeled documents varies from 9000 – 170000. The dataset is provided with the rating score between 1 to 5 assigned by users, which can be used for the review rating prediction task. We extracted texts from both the summary field and the text field to represent a review text. We then extracted the rating score as a review’s corresponding real-valued label. In the cross-language scenario, we regarded English as the source language, and regarded German as the target language. The experiments were conducted on each product category separately. Without loss of generality, we sampled and used

only 8000 unlabeled documents for each product category. We use Mean Square Error (MSE) as the evaluation metric, which penalizes more severe errors more heavily.

In the experiments, our proposed co-regression algorithm (i.e. “co-regression”) is compared with the COREG algorithm (Zhou and Li, 2005) and a few other baselines. For our proposed co-regression algorithm, the growth size  $m$  is simply set to 50. We implemented the COREG algorithm by replacing the KNN regressor with the regression SVM and the pool size is also set to 50. The iteration number  $I$  varies from 1 to 150. The comparison results are shown in Figure 2.

We can see that on all product categories, the MSE values of our co-regression algorithm and the two component regressors tend to decline over a wide range of  $I$ , which means that the selected confidently labeled examples at each iteration are indeed helpful to improve the regressors. Our proposed co-regression algorithm outperforms all the baselines (including COREG) over different iteration members, which verifies the effectiveness of our proposed algorithm. We can also see that the COREG algorithm does not perform well for this cross-language regression task. Overall, our proposed co-regression algorithm can consistently improve the prediction results.

## 6 Conclusion and Future Work

In this paper, we study a new task of cross-language review rating prediction and propose a new co-regression algorithm to address this task. In future work, we will apply the proposed co-regression algorithm to other cross-language or cross-domain regression problems in order to verify its robustness.

### Acknowledgments

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03) and National High-Tech R&D Program (2012AA011101).

<sup>3</sup> <http://www.uni-weimar.de/medien/webis/research/corpora/corpus-webis-cls-10.html>

## References

- Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 127-135.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Annual Meeting-Association For Computational Linguistics.
- 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, pp. 92-100.
- Hang Cui, Vibhu Mittal, and Mayur Datar. 2006. Comparative experiments on sentiment classification for online product reviews. In Proceedings of the National Conference on Artificial Intelligence.
- Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In WebDB.
- Thorsten Joachims, 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, MIT-Press.
- CaneWing Leung, Stephen Chi Chan, and Fu Chung. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In ECAI Workshop, pages 300-307.
- 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 the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI2011).
- Yue Lu, ChengXiang Zhai, Neel Sundaresan. 2009. Rated Aspect Summarization of Short Comments. Proceedings of the World Wide Conference 2009 (WWW'09), pages 131-140.
- Bin Lu, Chenhao Tan, Claire Cardie, Ka Yin Benjamin TSOU. 2011a. Joint bilingual sentiment classification with unlabeled parallel corpora. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 320-330.
- Bin Lu, Myle Ott, Claire Cardie and Benjamin K. Tsou. 2011b. Multi-aspect sentiment analysis with topic models. In Proceedings of Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, pp. 81-88, IEEE.
- Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Ge Xu, and Houfeng Wang. 2012. Cross-Lingual Mixture Model for Sentiment Classification. In Proceedings of ACL-2012.
- Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In Proceedings of ACL-2007.
- 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, pp. 79-86, 2002.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the ACL, pages 115-124.
- Peter Prettenhofer and Benno Stein. 2010. Cross-Language Text Classification using Structural Correspondence Learning. In 48th Annual Meeting of the Association of Computational Linguistics (ACL 10), 1118-1127.
- Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In COLING, pages 913-921, Stroudsburg, PA, USA, 2010. ACL.
- Horacio Saggion, Elena Lloret, and Manuel Palomar. 2012. Can text summaries help predict ratings? a case study of movie reviews. Natural Language Processing and Information Systems (2012): 271-276.
- Lei Shi, Rada Mihalcea, and Mingjun Tian. 2010. Cross language text classification by model translation and semi-supervised learning. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1057-1067, 2010.
- Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL).
- Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In Proceedings of ACL-08:HLT, pages 308-316.
- 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, pp. 417-424.
- Vladimir N. Vapnik, 1995. The Nature of Statistical Learning Theory. Springer.
- Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on

Natural Language Processing of the AFNLP, pp. 235-243.

Hongning Wang, Yue Lu, ChengXiang Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10), pages 115-124.

Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. 2009. Multi-aspect opinion polling from textual reviews. In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 1799-1802. ACM.

Zhi-Hua Zhou and Ming Li. 2005. Semi-supervised regression with co-training. In Proceedings of the 19th international joint conference on Artificial intelligence, pp. 908-913. Morgan Kaufmann Publishers Inc.

# Extracting Definitions and Hypernym Relations relying on Syntactic Dependencies and Support Vector Machines

**Guido Boella**

University of Turin  
Department of Computer Science  
boella@di.unito.it

**Luigi Di Caro**

University of Turin  
Department of Computer Science  
dicaro@di.unito.it

## Abstract

In this paper we present a technique to reveal definitions and hypernym relations from text. Instead of using pattern matching methods that rely on lexico-syntactic patterns, we propose a technique which only uses syntactic dependencies between terms extracted with a syntactic parser. The assumption is that syntactic information are more robust than patterns when coping with length and complexity of the sentences. Afterwards, we transform such syntactic contexts in abstract representations, that are then fed into a Support Vector Machine classifier. The results on an annotated dataset of definitional sentences demonstrate the validity of our approach overtaking current state-of-the-art techniques.

## 1 Introduction

Nowadays, there is a huge amount of textual data coming from different sources of information. Wikipedia<sup>1</sup>, for example, is a free encyclopedia that currently contains 4,208,409 English articles<sup>2</sup>. Even Social Networks play a role in the construction of data that can be useful for Information Extraction tasks like Sentiment Analysis, Question Answering, and so forth.

From another point of view, there is the need of having more structured data in the forms of ontologies, in order to allow semantics-based retrieval and reasoning. Ontology Learning is a task that permits to automatically (or semi-automatically) extract structured knowledge from plain text. Manual construction of ontologies usually requires strong efforts from domain experts, and it thus needs an automatization in such sense.

<sup>1</sup><http://www.wikipedia.org/>

<sup>2</sup>April 12, 2013.

In this paper, we focus on the extraction of hypernym relations. The first step of such task relies on the identification of what (Navigli and Velardi, 2010) called *definitional sentences*, i.e., sentences that contain at least one hypernym relation. This subtask is important by itself for many tasks like Question Answering (Cui et al., 2007), construction of glossaries (Klavans and Muresan, 2001), extraction of taxonomic and non-taxonomic relations (Navigli, 2009; Snow et al., 2004), enrichment of concepts (Gangemi et al., 2003; Cataldi et al., 2009), and so forth.

Hypernym relation extraction involves two aspects: linguistic knowledge, and model learning. Patterns collapse both of them, preventing to face them separately with the most suitable techniques. First, patterns have limited expressivity; then, linguistic knowledge inside patterns is learned from small corpora, so it is likely to have low coverage. Classification strictly depends on the learned patterns, so performance decreases, and the available classification techniques are restricted to those compatible with the pattern approach. Instead, we use a syntactic parser for the first aspect (with all its native and domain-independent knowledge on language expressivity), and a state-of-the-art approach to learn models with the use of Support Vector Machine classifiers.

Our assumption is that syntax is less dependent than learned patterns from the length and the complexity of textual expressions. In some way, patterns grasp syntactic relationships, but they actually do not use them as input knowledge.

## 2 Related Work

In this section we present the current state of the art concerning the automatic extraction of definitions and hypernym relations from plain text. We will use the term *definitional sentence* referring to the more general meaning given by (Navigli and Velardi, 2010): *A sentence that provides a for-*

mal explanation for the term of interest, and more specifically as a sentence containing at least one hypernym relation.

So far, most of the proposed techniques rely on lexico-syntactic patterns, either manually or semi-automatically produced (Hovy et al., 2003; Zhang and Jiang, 2009; Westerhout, 2009). Such patterns are sequences of words like “*is a*” or “*refers to*”, rather than more complex sequences including part-of-speech tags.

In the work of (Westerhout, 2009), after a manual identification of types of definitions and related patterns contained in a corpus, he successively applied Machine Learning techniques on syntactic and location features to improve the results.

A fully-automatic approach has been proposed by (Borg et al., 2009), where the authors applied genetic algorithms to the extraction of English definitions containing the keyword “*is*”. In detail, they assign weights to a set of features for the classification of definitional sentences, reaching a precision of 62% and a recall of 52%.

Then, (Cui et al., 2007) proposed an approach based on *soft patterns*, i.e., probabilistic lexico-semantic patterns that are able to generalize over rigid patterns enabling partial matching by calculating a generative degree-of-match probability between a test instance and the set of training instances.

Similarly to our approach, (Fahmi and Bouma, 2006) used three different Machine Learning algorithms to distinguish actual definitions from other sentences also relying on syntactic features, reaching high accuracy levels.

The work of (Klavans and Muresan, 2001) relies on a rule-based system that makes use of “cue phrases” and structural indicators that frequently introduce definitions, reaching 87% of precision and 75% of recall on a small and domain-specific corpus.

As for the task of definition extraction, most of the existing approaches use symbolic methods that are based on lexico-syntactic patterns, which are manually crafted or deduced automatically. The seminal work of (Hearst, 1992) represents the main approach based on fixed patterns like “ $NP_x$  is a/an  $NP_y$ ” and “ $NP_x$  such as  $NP_y$ ”, that usually imply  $\langle x$  IS-A  $y \rangle$ .

The main drawback of such technique is that it does not face the high variability of how a relation can be expressed in natural language. Still, it gen-

erally extracts single-word terms rather than well-formed and compound concepts. (Berland and Charniak, 1999) proposed similar lexico-syntactic patterns to extract *part-whole* relationships.

(Del Gaudio and Branco, 2007) proposed a rule-based approach to the extraction of hypernyms that, however, leads to very low accuracy values in terms of Precision.

(Ponzetto and Strube, 2007) proposed a technique to extract hypernym relations from Wikipedia by means of methods based on the connectivity of the network and classical lexico-syntactic patterns. (Yamada et al., 2009) extended their work by combining extracted Wikipedia entries with new terms contained in additional web documents, using a distributional similarity-based approach.

Finally, pure statistical approaches present techniques for the extraction of hierarchies of terms based on words frequency as well as co-occurrence values, relying on clustering procedures (Candan et al., 2008; Fortuna et al., 2006; Yang and Callan, 2008). The central hypothesis is that similar words tend to occur together in similar contexts (Harris, 1954). Despite this, they are defined by (Biemann, 2005) as *prototype-based ontologies* rather than formal terminological ontologies, and they usually suffer from the problem of data sparsity in case of small corpora.

### 3 Approach

In this section we present our approach to identify hypernym relations within plain text. Our methodology consists in relaxing the problem into two easier subtasks. Given a relation  $rel(x, y)$  contained in a sentence, the task becomes to find 1) a possible  $x$ , and 2) a possible  $y$ . In case of more than one possible  $x$  or  $y$ , a further step is needed to associate the correct  $x$  to the right  $y$ .

By seeing the problem as two different classification problems, there is no need to create abstract patterns between the target terms. In addition to this, the general problem of identifying definitional sentences can be seen as to find at least one  $x$  and one  $y$  in a sentence.

#### 3.1 Local Syntactic Information

Dependency parsing is a procedure that extracts syntactic dependencies among the terms contained in a sentence. The idea is that, given a hypernym relation, hyponyms and hypernyms may be

characterized by specific sets of syntactic contexts. According to this assumption, the task can be seen as a classification problem where each term in a sentence has to be classified as hyponym, hypernym, or neither of the two.

For each noun, we construct a textual representation containing its syntactic dependencies (i.e., its syntactic context). In particular, for each syntactic dependency  $dep(a, b)$  (or  $dep(b, a)$ ) of a target noun  $a$ , we create an abstract token<sup>3</sup>  $dep\text{-}target\text{-}\hat{b}$  (or  $dep\text{-}\hat{b}\text{-}target$ ), where  $\hat{b}$  becomes the generic string “*noun*” in case it is another noun; otherwise it is equal to  $b$ . This way, the nouns are transformed into abstract strings; on the contrary, no abstraction is done for verbs.

For instance, let us consider the sentence “*The Albedo of an object is the extent to which it diffusely reflects light from the sun*”. After the Part-Of-Speech annotation, the parser will extract a series of syntactic dependencies like “*det(Albedo, The)*”, “*nsubj(extent, Albedo)*”, “*prepof(Albedo, object)*”, where *det* identifies a determiner, *nsubj* represents a noun phrase which is the syntactic subject of a clause, and so forth<sup>4</sup>. Then, such dependencies will be transformed in abstract terms like “*det-target-the*”, “*nsubj-noun-target*”, and “*prepof-target-noun*”. These triples represent the feature space on which the Support Vector Machine classifiers will construct the models.

### 3.2 Learning phase

Our model assumes a transformation of the local syntactic information into labelled numeric vectors. More in detail, given a sentence  $S$  annotated with the terms linked by the hypernym relation, the system produces as many input instances as the number of nouns contained in  $S$ . For each noun  $n$  in  $S$ , the method produces two instances  $S_x^n$  and  $S_y^n$ , associated to the label *positive* or *negative* depending on their presence in the target relation (i.e., as  $x$  or  $y$  respectively). If a noun is not involved in a hypernym relation, both the two instances will have the label *negative*. At the end of this process, two training sets are built, i.e., one for each relation argument, namely the  $x$ -set and the  $y$ -set. All the instances of both the datasets are then transformed into numeric vectors according

<sup>3</sup>We make use of the term “abstract” to indicate that some words are replaced with more general entity identifiers.

<sup>4</sup>A complete overview of the Stanford dependencies is available at [http://nlp.stanford.edu/software/dependencies\\_manual.pdf](http://nlp.stanford.edu/software/dependencies_manual.pdf).

to the Vector Space Model (Salton et al., 1975), and are finally fed into a Support Vector Machine classifier<sup>5</sup> (Cortes and Vapnik, 1995). We refer to the two resulting models as the  $x$ -model and the  $y$ -model. These models are binary classifiers that, given the local syntactic information of a noun, estimate if it can be respectively an  $x$  or a  $y$  in a hypernym relation.

Once the  $x$ -model and the  $y$ -model are built, we can both classify definitional sentences and extract hypernym relations. In the next section we deepen our proposed strategy in that sense.

The whole set of instances of all the sentences are fed into two Support Vector Machine classifiers, one for each target label (i.e.,  $x$  and  $y$ ).

At this point, it is possible to classify each term as possible  $x$  or  $y$  by querying the respective classifiers with its local syntactic information.

## 4 Setting of the Tasks

In this section we present how our proposed technique is able to classify definitional sentences unraveling hypernym relations.

### 4.1 Classification of definitional sentences

As already mentioned in previous sections, we label as *definitional* all the sentences that contain at least one noun  $n$  classified as  $x$ , and one noun  $m$  classified as  $y$  (where  $n \neq m$ ). In this phase, it is not further treated the case of having more than one  $x$  or  $y$  in one single sentence. Thus, given an input sentence:

1. we extract all the nouns (POS-tagging),
2. we extract all the syntactic dependencies of the nouns (dependency parsing),
3. we fed each noun (i.e., its instance) to the  $x$ -model and to the  $y$  model,
4. we check if there exist at least one noun classified as  $x$  and one noun classified as  $y$ : in this case, we classify the sentences as *definitional*.

### 4.2 Extraction of hypernym relations

Our method for extracting hypernym relations makes use of both the  $x$ -model and the  $y$ -model as for the the task of classifying definitional sentences. If exactly one  $x$  and one  $y$  are identified

<sup>5</sup>We used the Sequential Minimal Optimization implementation of the Weka framework (Hall et al., 2009).

in the same sentence, they are directly connected and the relation is extracted. The only constraint is that  $x$  and  $y$  must be connected within the same parse tree.

Now, considering our target relation  $hyp(x, y)$ , in case the sentence contains more than one noun that is classified as  $x$  (or  $y$ ), there are two possible scenarios:

1. there are actually more than one  $x$  (or  $y$ ), or
2. the classifiers returned some false positive.

Up to now, we decided to keep all the possible combinations, without further filtering operations<sup>6</sup>. Finally, in case of multiple classifications of both  $x$  and  $y$ , i.e., if there are multiple  $x$  and multiple  $y$  at the same time, the problem becomes to select which  $x$  is linked to which  $y$ <sup>7</sup>. To do this, we simply calculate the distance between these terms in the parse tree (the closer the terms, the better the connection between the two). Nevertheless, in the used corpus, only around 1.4% of the sentences are classified with multiple  $x$  and  $y$ .

Finally, since our method is able to extract single nouns that can be involved in a hypernym relation, we included modifiers preceded by preposition “of”, while the other modifiers are removed. For example, considering the sentence “An Archipelago is a chain of islands”, the whole chunk “chain of islands” is extracted from the single triggered noun chain.

## 5 Evaluation

In this section we present the evaluation of our approach, that we carried out on an annotated dataset of definitional sentences (Navigli et al., 2010). The corpus contains 4,619 sentences extracted from Wikipedia, and only 1,908 are annotated as *definitional*. On a first instance, we test the classifiers on the extraction of hyponyms ( $x$ ) and hypernyms ( $y$ ) from the definitional sentences, independently. Then, we evaluate the classification of definitional sentences. Finally, we evaluate the ability of our technique when extracting whole hypernym relations. With the used dataset, the constructed training sets for the two classifiers ( $x$ -set and  $y$ -set) resulted to have approximately 1,500 features.

<sup>6</sup>We only used the constraint that  $x$  has to be different from  $y$ .

<sup>7</sup>Notice that this is different from the case in which a single noun is labeled as both  $x$  and  $y$ .

| Alg.     | $P$   | $R$          | $F$          | $Acc$        |
|----------|-------|--------------|--------------|--------------|
| WCL-3    | 98.8% | 60.7%        | 75.2 %       | 83.4 %       |
| Star P.  | 86.7% | 66.1%        | 75.0 %       | 81.8 %       |
| Bigrams  | 66.7% | <b>82.7%</b> | 73.8 %       | 75.8 %       |
| Our sys. | 88.0% | 76.0%        | <b>81.6%</b> | <b>89.6%</b> |

Table 1: Evaluation results for the classification of definitional sentences, in terms of Precision ( $P$ ), Recall ( $R$ ), F-Measure ( $F$ ), and Accuracy ( $Acc$ ), using 10-folds cross validation. For the WCL-3 approach and the Star Patterns see (Navigli and Velardi, 2010), and (Cui et al., 2007) for Bigrams.

| Algorithm  | $P$           | $R$           | $F$           |
|------------|---------------|---------------|---------------|
| WCL-3      | 78.58%        | 60.74% *      | 68.56%        |
| Our system | <b>83.05%</b> | <b>68.64%</b> | <b>75.16%</b> |

Table 2: Evaluation results for the hypernym relation extraction, in terms of Precision ( $P$ ), Recall ( $R$ ), and F-Measure ( $F$ ). For the WCL-3 approach, see (Navigli and Velardi, 2010). These results are obtained using 10-folds cross validation (\* Recall has been inherited from the definition classification task, since no indication has been reported in their contribution).

## 5.1 Results

In this section we present the evaluation of our technique on both the tasks of classifying definitional sentences and extracting hypernym relations. Notice that our approach is susceptible from the errors given by the POS-tagger<sup>8</sup> and the syntactic parser<sup>9</sup>. In spite of this, our approach demonstrates how syntax can be more robust for identifying semantic relations. Our approach does not make use of the full parse tree, and we are not dependent on a complete and correct result of the parser.

The goal of our evaluation is twofold: first, we evaluate the ability of classifying definitional sentences; finally, we measure the accuracy of the hypernym relation extraction.

A definitional sentences is extracted only if at least one  $x$  and one  $y$  are found in the same sentence. Table 1 shows the accuracy of the approach for this task. As can be seen, our proposed approach has a high Precision, with a high Recall. Although Precision is lower than the pat-

<sup>8</sup><http://nlp.stanford.edu/software/tagger.shtml>

<sup>9</sup><http://www-nlp.stanford.edu/software/lex-parser.shtml>

tern matching approach proposed by (Navigli and Velardi, 2010), our Recall is higher, leading to an higher overall F-Measure.

Table 2 shows the results of the extraction of the whole hypernym relations. Note that our approach has high levels of accuracy. In particular, even in this task, our system outperforms the pattern matching algorithm proposed by (Navigli and Velardi, 2010) in terms of Precision and Recall.

## 6 Conclusion and Future Work

We presented an approach to reveal definitions and extract underlying hypernym relations from plain text, making use of local syntactic information fed into a Support Vector Machine classifier. The aim of this work was to revisit these tasks as classical supervised learning problems that usually carry to high accuracy levels with high performance when faced with standard Machine Learning techniques. Our first results on this method highlight the validity of the approach by significantly improving current state-of-the-art techniques in the classification of definitional sentences as well as in the extraction of hypernym relations from text. In future works, we aim at using larger syntactic contexts. In fact, currently, the detection does not surpass the sentence level, while taxonomical information can be even contained in different sentences or paragraphs. We also aim at evaluating our approach on the construction of entire taxonomies starting from domain-specific text corpora, as in (Navigli et al., 2011; Velardi et al., 2012). Finally, the desired result of the task of extracting hypernym relations from text (as for any semantic relationships in general) depends on the domain and the specific later application. Thus, we think that a precise evaluation and comparison of any systems strictly depends on these factors. For instance, given a sentence like “In mathematics, computing, linguistics and related disciplines, an algorithm is a sequence of instructions” one could want to extract only “instructions” as hypernym (as done in the annotation), rather than the entire chunk “sequence of instructions” (as extracted by our technique). Both results can be valid, and a further discrimination can only be done if a specific application or use of this knowledge is taken into consideration.

## References

- M. Berland and E. Charniak. 1999. Finding parts in very large corpora. In *Annual Meeting Association for Computational Linguistics*, volume 37, pages 57–64. Association for Computational Linguistics.
- C. Biemann. 2005. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93.
- C. Borg, M. Rosner, and G. Pace. 2009. Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 26–32. Association for Computational Linguistics.
- K.S. Candan, L. Di Caro, and M.L. Sapino. 2008. Creating tag hierarchies for effective navigation in social media. In *Proceedings of the 2008 ACM workshop on Search in social media*, pages 75–82. ACM.
- Mario Cataldi, Claudio Schifanella, K Selçuk Candan, Maria Luisa Sapino, and Luigi Di Caro. 2009. Cosena: a context-based search and navigation system. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, page 33. ACM.
- C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.
- Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2007. Soft pattern matching models for definitional question answering. *ACM Trans. Inf. Syst.*, 25(2), April.
- R. Del Gaudio and A. Branco. 2007. Automatic extraction of definitions in portuguese: A rule-based approach. *Progress in Artificial Intelligence*, pages 659–670.
- I. Fahmi and G. Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*, pages 64–71.
- B. Fortuna, D. Mladenič, and M. Grobelnik. 2006. Semi-automatic construction of topic ontologies. *Semantics, Web and Mining*, pages 121–131.
- Aldo Gangemi, Roberto Navigli, and Paola Velardi. 2003. The ontowordnet project: Extension and axiomatization of conceptual relations in wordnet. In Robert Meersman, Zahir Tari, and Douglas C. Schmidt, editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 820–838. Springer Berlin Heidelberg.
- 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.
- Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.



- M.A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.
- E. Hovy, A. Philpot, J. Klavans, U. Germann, P. Davis, and S. Popper. 2003. Extending metadata definitions by automatically extracting and organizing glossary definitions. In *Proceedings of the 2003 annual national conference on Digital government research*, pages 1–6. Digital Government Society of North America.
- J.L. Klavans and S. Muresan. 2001. Evaluation of the finder system for fully automatic glossary construction. In *Proceedings of the AMIA Symposium*, page 324. American Medical Informatics Association.
- Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden, July. Association for Computational Linguistics.
- Roberto Navigli, Paola Velardi, and Juana Mara Ruiz-Martinez. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
- R. Navigli, P. Velardi, and S. Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1872–1877. AAAI Press.
- R. Navigli. 2009. Using cycles and quasi-cycles to disambiguate dictionary glosses. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 594–602. Association for Computational Linguistics.
- S.P. Ponzetto and M. Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the national conference on artificial intelligence*, volume 22, page 1440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November.
- R. Snow, D. Jurafsky, and A.Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.
- Paola Velardi, Stefano Faralli, and Roberto Navigli. 2012. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, pages 1–72.
- Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction, WDE '09*, pages 61–67, Stroudsburg, PA, USA. Association for Computational Linguistics.
- I. Yamada, K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. De Saeger, F. Bond, and A. Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 929–937. Association for Computational Linguistics.
- H. Yang and J. Callan. 2008. Ontology generation for large email collections. In *Proceedings of the 2008 international conference on Digital government research*, pages 254–261. Digital Government Society of North America.
- Chunxia Zhang and Peng Jiang. 2009. Automatic extraction of definitions. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 364–368, aug.

# *Neighbors Help: Bilingual Unsupervised WSD Using Context*

Sudha Bhingardive Samiulla Shaikh Pushpak Bhattacharyya

Department of Computer Science and Engineering,

IIT Bombay, Powai,

Mumbai, 400076.

{sudha , samiulla , pb}@cse.iitb.ac.in

## **Abstract**

Word Sense Disambiguation (WSD) is one of the toughest problems in NLP, and in WSD, verb disambiguation has proved to be extremely difficult, because of high degree of polysemy, too fine grained senses, absence of deep verb hierarchy and low inter annotator agreement in verb sense annotation. Unsupervised WSD has received widespread attention, but has performed poorly, specially on verbs. Recently an unsupervised bilingual EM based algorithm has been proposed, which makes use only of the raw counts of the translations in comparable corpora (Marathi and Hindi). But the performance of this approach is poor on verbs with accuracy level at 25-38%. We suggest a modification to this mentioned formulation, using context and semantic relatedness of neighboring words. An improvement of 17% - 35% in the accuracy of verb WSD is obtained compared to the existing EM based approach. On a general note, the work can be looked upon as contributing to the framework of unsupervised WSD through context aware expectation maximization.

## **1 Introduction**

The importance of unsupervised approaches in WSD is well known, because they do not need sense tagged corpus. In multilingual unsupervised scenario, either comparable or parallel corpora have been used by past researchers for disambiguation (Dagan et al., 1991; Diab and Resnik, 2002; Kaji and Morimoto, 2002; Specia et al., 2005; Lefever and Hoste, 2010; Khapra et al., 2011). Recent work by Khapra et al., (2011) has shown that, in comparable corpora, sense distribution of a word in one language can be estimated

using the raw counts of translations of the target words in the other language; such sense distributions contribute to the ranking of senses. Since translations can themselves be ambiguous, Expectation Maximization based formulation is used to determine the sense frequencies. Using this approach every instance of a word is tagged with the most probable sense according to the algorithm.

In the above formulation, no importance is given to the context. That would do, had the accuracy of disambiguation on verbs not been poor 25-35%. This motivated us to propose and investigate use of context in the formulation by Khapra et al. (2011).

For example consider the sentence in chemistry domain, “*Keep the beaker on the flat table.*” In this sentence, the target word ‘table’ will be tagged as ‘the tabular array’ sense since it is dominant in the chemistry domain by their algorithm. But its actual sense is ‘a piece of furniture’ which can be captured only if context is taken into consideration. In our approach we tackle this problem by taking into account the words from the context of the target word. We use semantic relatedness between translations of the target word and those of its context words to determine its sense.

Verb disambiguation has proved to be extremely difficult (Jean, 2004), because of high degree of polysemy (Khapra et al., 2010), too fine grained senses, absence of deep verb hierarchy and low inter annotator agreement in verb sense annotation. On the other hand, verb disambiguation is very important for NLP applications like MT and IR. Our approach has shown significant improvement in verb accuracy as compared to Khapra’s (2011) approach.

The roadmap of the paper is as follows. Section 2 presents related work. Section 3 covers the background work. Section 4 explains the modified EM formulation using context and semantic relatedness. Section 5 presents the experimental setup.

Results are presented in section 6. Section 7 covers phenomena study and error analysis. Conclusions and future work are given in the last section, section 8.

## 2 Related work

Word Sense Disambiguation is one of the hardest problems in NLP. Successful supervised WSD approaches (Lee et al., 2004; Ng and Lee, 1996) are restricted to resource rich languages and domains. They are directly dependent on availability of good amount of sense tagged data. Creating such a costly resource for all language-domain pairs is impracticable looking at the amount of time and money required. Hence, unsupervised WSD approaches (Diab and Resnik, 2002; Kaji and Morimoto, 2002; Mihalcea et al., 2004; Jean, 2004; Khapra et al., 2011) attract most of the researchers.

## 3 Background

Khapra et al. (2011) dealt with bilingual unsupervised WSD. It uses EM algorithm for estimating sense distributions in comparable corpora. Every polysemous word is disambiguated using the raw counts of its translations in different senses. Synset aligned multilingual dictionary (Mohanty et al., 2008) is used for finding its translations. In this dictionary, synsets are linked, and after that the words inside the synsets are also linked. For example, for the concept of ‘boy’, the Hindi synset {*ladakaa*, *balak*, *bachhaa*} is linked with the Marathi synset {*mulagaa*, *poragaa*, *por*}. The Marathi word ‘*mulagaa*’ is linked to the Hindi word ‘*ladakaa*’ which is its exact lexical substitution.

Suppose words  $u$  in language  $L_1$  and  $v$  in language  $L_2$  are translations of each other and their senses are required. The EM based formulation is as follows:

### E-Step:

$$P(S^{L_1}|u) = \frac{\sum_v P(\pi_{L_2}(S^{L_1})|v) \cdot \#(v)}{\sum_{S_i^{L_1}} \sum_x P(\pi_{L_2}(S_i^{L_1})|x) \cdot \#(x)}$$

where,  $S_i^{L_1} \in \text{synsets}_{L_1}(u)$

$v \in \text{crosslinks}_{L_2}(u, S^{L_1})$

$x \in \text{crosslinks}_{L_2}(u, S_i^{L_1})$

### M-Step:

$$P(S^{L_2}|v) = \frac{\sum_u P(\pi_{L_1}(S^{L_2})|u) \cdot \#(u)}{\sum_{S_i^{L_2}} \sum_y P(\pi_{L_1}(S_i^{L_2})|y) \cdot \#(y)}$$

where,  $S_i^{L_2} \in \text{synsets}_{L_2}(v)$

$u \in \text{crosslinks}_{L_1}(v, S^{L_2})$

$y \in \text{crosslinks}_{L_1}(v, S_i^{L_2})$

Here,

- ‘#’ indicates the raw count.
- $\text{crosslinks}_{L_1}(a, S^{L_2})$  is the set of possible translations of the word ‘ $a$ ’ from language  $L_1$  to  $L_2$  in the sense  $S^{L_2}$ .
- $\pi_{L_2}(S^{L_1})$  means the linked synset of the sense  $S^{L_1}$  in  $L_2$ .

E and M steps are symmetric except for the change in language. In both the steps, we estimate sense distribution in one language using raw counts of translations in another language. But this approach has following limitations:

**Poor performance on verbs:** This approach gives poor performance on verbs (25%-38%). See section 6.

**Same sense throughout the corpus:** Every occurrence of a word is tagged with the single sense found by the algorithm, throughout the corpus.

**Closed loop of translations:** This formulation does not work for some common words which have the same translations in all senses. For example, the verb ‘*karna*’ in Hindi has two different senses in the corpus *viz.*, ‘*to do*’ ( $S_1$ ) and ‘*to make*’ ( $S_2$ ). In both these senses, it gets translated as ‘*karne*’ in Marathi. The word ‘*karne*’ also back translates to ‘*karna*’ in Hindi through both its senses. In this case, the formulation works out as follows:

The probabilities are initialized uniformly. Hence,  $P(S_1|karna) = P(S_2|karna) = 0.5$ . Now, in first iteration the sense of ‘*karne*’ will be estimated as follows (E-step):

$$P(S_1|karne) = \frac{P(S_1|karna) * \#(karna)}{\#(karna)} = 0.5,$$

$$P(S_2|karna) = \frac{P(S_2|karna) * \#(karna)}{\#(karna)} = 0.5$$

Similarly, in M-step, we will get  $P(S_1|karna) = P(S_2|karna) = 0.5$ . Eventually, it will end up with initial probabilities and no strong decision can be made.

To address these problems we have introduced contextual clues in their formulation by using semantic relatedness.

#### 4 Modified Bilingual EM approach

We introduce context in the EM formulation stated above and treat the context as a bag of words. We assume that each word in the context influences the sense of the target word independently. Hence,

$$p(S|w, C) = \prod_{c_i \in C} p(S|w, c_i)$$

where,  $w$  is the target word,  $S$  is one of the candidate synsets of  $w$ ,  $C$  is the set of words in context (sentence in our case) and  $c_i$  is one of the context words.

Suppose we would have sense tagged data,  $p(S|w, c)$  could have been computed as:

$$p(S|w, c) = \frac{\#(S, w, c)}{\#(w, c)}$$

But since the sense tagged corpus is not available, we cannot find  $\#(S, w, c)$  from the corpus directly. However, we can estimate it using the comparable corpus in other language. Here, we assume that given a word and its context word in language  $L_1$ , the sense distribution in  $L_1$  will be same as that in  $L_2$  given the translation of a word and the translation of its context word in  $L_2$ . But these translations can be ambiguous, hence we can use Expectation Maximization approach similar to (Khpra et al., 2011) as follows:

**E-Step:**

$$P(S^{L_1}|u, a) = \frac{\sum_{v,b} P(\pi_{L_2}(S^{L_1})|v, b) \cdot \sigma(v, b)}{\sum_{S_i^{L_1}} \sum_{x,b} P(\pi_{L_2}(S_i^{L_1})|x, b) \cdot \sigma(x, b)}$$

where,  $S_i^{L_1} \in \text{synsets}_{L_1}(u)$

$a \in \text{context}(u)$

$v \in \text{crosslinks}_{L_2}(u, S^{L_1})$

$b \in \text{crosslinks}_{L_2}(a)$

$x \in \text{crosslinks}_{L_2}(u, S_i^{L_1})$

$\text{crosslinks}_{L_1}(a)$  is the set of all possible translations of the word 'a' from  $L_1$  to  $L_2$  in all its senses.

$\sigma(v, b)$  is the semantic relatedness between the senses of  $v$  and senses of  $b$ . Since,  $v$  and  $b$  go over all possible translations of  $u$  and  $a$  respectively.  $\sigma(v, b)$  has the effect of indirectly capturing the semantic similarity between the senses of  $u$  and  $a$ . A symmetric formulation in the M-step below takes the computation back from language  $L_2$  to language  $L_1$ . The semantic relatedness comes as an additional weighing factor, capturing context, in the probabilistic score.

**M-Step:**

$$P(S^{L_2}|v, b) = \frac{\sum_{u,a} P(\pi_{L_1}(S^{L_2})|u, a) \cdot \sigma(u, a)}{\sum_{S_i^{L_2}} \sum_{y,b} P(\pi_{L_1}(S_i^{L_2})|y, a) \cdot \sigma(y, a)}$$

where,  $S_i^{L_2} \in \text{synsets}_{L_2}(v)$

$b \in \text{context}(v)$

$u \in \text{crosslinks}_{L_1}(v, S^{L_2})$

$a \in \text{crosslinks}_{L_1}(b)$

$y \in \text{crosslinks}_{L_1}(v, S_i^{L_2})$

$\sigma(u, a)$  is the semantic relatedness between the senses of  $u$  and senses of  $a$  and contributes to the score like  $\sigma(v, b)$ .

Note how the computation moves back and forth between  $L_1$  and  $L_2$  considering translations of both target words and their context words.

In the above formulation, we could have considered the term  $\#(\text{word}, \text{context\_word})$  (i.e., the co-occurrence count of the translations of the word and the context word) instead of  $\sigma(\text{word}, \text{context\_word})$ . But it is very unlikely that every translation of a word will co-occur with

| Algorithm | HIN-HEALTH |       |       |              |         | MAR-HEALTH |       |       |              |         |
|-----------|------------|-------|-------|--------------|---------|------------|-------|-------|--------------|---------|
|           | NOUN       | ADV   | ADJ   | VERB         | Overall | NOUN       | ADV   | ADJ   | VERB         | Overall |
| EM-C      | 59.82      | 67.80 | 56.66 | <b>60.38</b> | 59.63   | 62.90      | 62.54 | 53.63 | <b>52.49</b> | 59.77   |
| EM        | 60.68      | 67.48 | 55.54 | <b>25.29</b> | 58.16   | 63.88      | 58.88 | 55.71 | <b>35.60</b> | 58.03   |
| WFS       | 53.49      | 73.24 | 55.16 | <b>38.64</b> | 54.46   | 59.35      | 67.32 | 38.12 | <b>34.91</b> | 52.57   |
| RB        | 32.52      | 45.08 | 35.42 | <b>17.93</b> | 33.31   | 33.83      | 38.76 | 37.68 | <b>18.49</b> | 32.45   |

Table 1: Comparison(F-Score) of EM-C and EM for Health domain

| Algorithm | HIN-TOURISM |       |       |              |         | MAR-TOURISM |       |       |              |         |
|-----------|-------------|-------|-------|--------------|---------|-------------|-------|-------|--------------|---------|
|           | NOUN        | ADV   | ADJ   | VERB         | Overall | NOUN        | ADV   | ADJ   | VERB         | Overall |
| EM-C      | 62.78       | 65.10 | 54.67 | <b>55.24</b> | 60.70   | 59.08       | 63.66 | 58.02 | <b>55.23</b> | 58.67   |
| EM        | 61.16       | 62.31 | 56.02 | <b>31.85</b> | 57.92   | 59.66       | 62.15 | 58.42 | <b>38.33</b> | 56.90   |
| WFS       | 63.98       | 75.94 | 52.72 | <b>36.29</b> | 60.22   | 61.95       | 62.39 | 48.29 | <b>46.56</b> | 57.47   |
| RB        | 32.46       | 42.56 | 36.35 | <b>18.29</b> | 32.68   | 33.93       | 39.30 | 37.49 | <b>15.99</b> | 32.65   |

Table 2: Comparison(F-Score) of EM-C and EM for Tourism domain

every translation of its context word considerable number of times. This term may make sense only if we have arbitrarily large comparable corpus in the other language.

#### 4.1 Computation of semantic relatedness

The semantic relatedness is computed by taking the inverse of the length of the shortest path among two senses in the wordnet graph (Pedersen et al., 2005). All the semantic relations (including cross-part-of-speech links) *viz.*, hypernymy, hyponymy, meronymy, entailment, attribute *etc.*, are used for computing the semantic relatedness.

Sense scores thus obtained are used to disambiguate all words in the corpus. We consider all the content words from the context for disambiguation of a word. The winner sense is the one with the highest probability.

## 5 Experimental setup

We have used freely available in-domain comparable corpora<sup>1</sup> in Hindi and Marathi languages. These corpora are available for health and tourism domains. The dataset is same as that used in (Khapra et al., 2011) in order to compare the performance.

## 6 Results

Table 1 and Table 2 compare the performance of the following two approaches:

1. **EM-C** (EM with Context): Our modified approach explained in section 4.
2. **EM**: Basic EM based approach by Khapra et al., (2011).

3. **WFS**: Wordnet First Sense baseline.

4. **RB**: Random baseline.

Results clearly show that EM-C outperforms EM especially in case of verbs in all language-domain pairs. In health domain, verb accuracy is increased by 35% for Hindi and 17% for Marathi, while in tourism domain, it is increased by 23% for Hindi and 17% for Marathi. The overall accuracy is increased by (1.8-2.8%) for health domain and (1.5-1.7%) for tourism domain. Since there are less number of verbs, the improved accuracy is not directly reflected in the overall performance.

## 7 Error analysis and phenomena study

Our approach tags all the instances of a word depending on its context as apposed to basic EM approach. For example, consider the following sentence from the tourism domain:

वह पत्ते खेल रहे थे ।  
(vaha patte khel rahe the)  
(They were playing cards/leaves)

Here, the word पत्ते (plural form of पत्ता) has two senses *viz.*, 'leaf' and 'playing\_card'. In tourism domain, the 'leaf' sense is more dominant. Hence, basic EM will tag पत्ते with 'leaf' sense. But it's true sense is 'playing\_card'. The true sense is captured only if context is considered. Here, the word खेलना (to play) (root form of खेल) endorses the 'playing\_card' sense of the word पत्ता. This phenomenon is captured by our approach through semantic relatedness.

But there are certain cases where our algorithm fails. For example, consider the following sentence:

<sup>1</sup>[http://www.cfilt.iitb.ac.in/wsd/annotated\\_corpus/](http://www.cfilt.iitb.ac.in/wsd/annotated_corpus/)

वह पेड के निचे पत्ते खेल रहे थे।

(vaha ped ke niche patte khel rahe the)

(They were playing cards/leaves below the tree)

Here, two strong context words पेड (tree) and खेल (play) are influencing the sense of the word पत्ते. Semantic relatedness between पेड (tree) and पत्ता (leaf) is more than that of खेल (play) and पत्ता (playing\_card). Hence, the 'leaf sense' is assigned to पत्ता.

This problem occurred because we considered the context as a bag of words. This problem can be solved by considering the semantic structure of the sentence. In this example, the word पत्ता (leaf/playing\_card) is the subject of the verb खेलना (to play) while पेड (tree) is not even in the same clause with पत्ता (leaf/playing\_cards). Thus we could consider खेलना (to play) as the stronger clue for its disambiguation.

## 8 Conclusion and Future Work

We have presented a context aware EM formulation building on the framework of Khapra et al (2011). Our formulation solves the problems of “inhibited progress due to lack of translation diversity” and “uniform sense assignment, irrespective of context” that the previous EM based formulation of Khapra et al. suffers from. More importantly our accuracy on verbs is much higher and more than the state of the art, to the best of our knowledge. Improving the performance on other parts of speech is the primary future work. Future directions also point to usage of semantic role clues, investigation of familiarly apart pair of languages and effect of variation of measures of semantic relatedness.

## References

- Ido Dagan, Alon Itai, and Ulrike Schwall. 1991. Two languages are more informative than one. In Douglas E. Appelt, editor, *ACL*, pages 130–137. ACL.
- 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*, ACL '02, pages 255–262, Morristown, NJ, USA. Association for Computational Linguistics.
- Véronis Jean. 2004. Hyperlex: Lexical cartography for information retrieval. In *Computer Speech and Language*, pages 18(3):223–252.
- Hiroyuki Kaji and Yasutsugu Morimoto. 2002. Unsupervised word sense disambiguation using bilingual comparable corpora. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mitesh M. Khapra, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. All words domain adapted wsd: Finding a middle ground between supervision and unsupervision. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL*, pages 1532–1541. The Association for Computer Linguistics.
- Mitesh M Khapra, Salil Joshi, and Pushpak Bhattacharyya. 2011. It takes two to tango: A bilingual unsupervised approach for estimating sense distributions using expectation maximization. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 695–704, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.
- K. Yoong Lee, Hwee T. Ng, and Tee K. Chia. 2004. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 137–140.
- Els Lefever and Veronique Hoste. 2010. Semeval-2010 task 3: cross-lingual word sense disambiguation. In Katrin Erk and Carlo Strapparava, editors, *SemEval 2010 : 5th International workshop on Semantic Evaluation : proceedings of the workshop*, pages 15–20. ACL.
- Rada Mihalcea, Paul Tarau, and Elizabeth Figa. 2004. Pagerank on semantic networks, with application to word sense disambiguation. In *COLING*.
- Rajat Mohanty, Pushpak Bhattacharyya, Prabhakar Pande, Shraddha Kalele, Mitesh Khapra, and Aditya Sharma. 2008. Synset based multilingual dictionary: Insights, applications and challenges. In *Global Wordnet Conference*.
- Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47, Morristown, NJ, USA. ACL.
- T. Pedersen, S. Banerjee, and S. Patwardhan. 2005. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute, March.
- Lucia Specia, Maria Das Graças, Volpe Nunes, and Mark Stevenson. 2005. Exploiting parallel texts to produce a multilingual sense tagged corpus for word sense disambiguation. In *In Proceedings of RANLP-05, Borovets*, pages 525–531.

# Reducing Annotation Effort for Quality Estimation via Active Learning

Daniel Beck and Lucia Specia and Trevor Cohn

Department of Computer Science

University of Sheffield

Sheffield, United Kingdom

{debeck1, l.specia, t.cohn}@sheffield.ac.uk

## Abstract

Quality estimation models provide feedback on the quality of machine translated texts. They are usually trained on human-annotated datasets, which are very costly due to its task-specific nature. We investigate active learning techniques to reduce the size of these datasets and thus annotation effort. Experiments on a number of datasets show that with as little as 25% of the training instances it is possible to obtain similar or superior performance compared to that of the complete datasets. In other words, our active learning query strategies can not only reduce annotation effort but can also result in better quality predictors.

## 1 Introduction

The purpose of machine translation (MT) quality estimation (QE) is to provide a quality prediction for new, unseen machine translated texts, without relying on reference translations (Blatz et al., 2004; Specia et al., 2009; Callison-Burch et al., 2012). This task is usually addressed with machine learning models trained on datasets composed of source sentences, their machine translations, and a quality label assigned by humans. A common use of quality predictions is the decision between post-editing a given machine translated sentence and translating its source from scratch, based on whether its post-editing effort is estimated to be lower than the effort of translating the source sentence.

Since quality scores for the training of QE models are given by human experts, the annotation process is costly and subject to inconsistencies due to the subjectivity of the task. To avoid inconsistencies because of disagreements among annotators, it is often recommended that a QE model is trained

for each translator, based on labels given by such a translator (Specia, 2011). This further increases the annotation costs because different datasets are needed for different tasks. Therefore, strategies to reduce the demand for annotated data are needed. Such strategies can also bring the possibility of selecting data that is less prone to inconsistent annotations, resulting in more robust and accurate predictions.

In this paper we investigate Active Learning (AL) techniques to reduce the size of the dataset while keeping the performance of the resulting QE models. AL provides methods to select informative data points from a large pool which, if labelled, can potentially improve the performance of a machine learning algorithm (Settles, 2010). The rationale behind these methods is to help the learning algorithm achieve satisfactory results from only on a subset of the available data, thus incurring less annotation effort.

## 2 Related Work

Most research work on QE for machine translation is focused on feature engineering and feature selection, with some recent work on devising more reliable and less subjective quality labels. Blatz et al. (2004) present the first comprehensive study on QE for MT: 91 features were proposed and used to train predictors based on an automatic metric (e.g. NIST (Doddington, 2002)) as the quality label. Quirk (2004) showed that small datasets manually annotated by humans for quality can result in models that outperform those trained on much larger, automatically labelled sets.

Since quality labels are subjective to the annotators' judgements, Specia and Farzindar (2010) evaluated the performance of QE models using HTER (Snover et al., 2006) as the quality score, i.e., the edit distance between the MT output and its post-edited version. Specia (2011) compared the performance of models based on labels for

post-editing effort, post-editing time, and HTER.

In terms of learning algorithms, by and large most approaches use Support Vector Machines, particularly regression-based approaches. For an overview on various feature sets and machine learning algorithms, we refer the reader to a recent shared task on the topic (Callison-Burch et al., 2012).

Previous work use supervised learning methods (“passive learning” following the AL terminology) to train QE models. On the other hand, AL has been successfully used in a number of natural language applications such as text classification (Lewis and Gale, 1994), named entity recognition (Vlachos, 2006) and parsing (Baldrige and Osborne, 2004). See Olsson (2009) for an overview on AL for natural language processing as well as a comprehensive list of previous work.

### 3 Experimental Settings

#### 3.1 Datasets

We perform experiments using four MT datasets manually annotated for quality:

**English-Spanish** (*en-es*): 2,254 sentences translated by Moses (Koehn et al., 2007), as provided by the WMT12 Quality Estimation shared task (Callison-Burch et al., 2012). Effort scores range from 1 (too bad to be post-edited) to 5 (no post-editing needed). Three expert post-editors evaluated each sentence and the final score was obtained by a weighted average between the three scores. We use the default split given in the shared task: 1,832 sentences for training and 432 for test.

**French-English** (*fr-en*): 2,525 sentences translated by Moses as provided in Specia (2011), annotated by a single translator. Human labels indicate post-editing effort ranging from 1 (too bad to be post-edited) to 4 (little or no post-editing needed). We use a random split of 90% sentences for training and 10% for test.

**Arabic-English** (*ar-en*): 2,585 sentences translated by two state-of-the-art SMT systems (denoted *ar-en-1* and *ar-en-2*), as provided in (Specia et al., 2011). A random split of 90% sentences for training and 10% for test is used. Human labels indicate the adequacy of the translation ranging from 1 (completely inadequate) to 4 (adequate). These datasets were annotated by two expert translators.

#### 3.2 Query Methods

The core of an AL setting is how the learner will gather new instances to add to its training data. In our setting, we use a pool-based strategy, where the learner queries an instance pool and selects the best instance according to an informativeness measure. The learner then asks an “oracle” (in this case, the human expert) for the true label of the instance and adds it to the training data.

Query methods use different criteria to predict how informative an instance is. We experiment with two of them: Uncertainty Sampling (US) (Lewis and Gale, 1994) and Information Density (ID) (Settles and Craven, 2008). In the following, we denote  $M(x)$  the query score with respect to method  $M$ .

According to the US method, the learner selects the instance that has the highest labelling variance according to its model:

$$US(x) = Var(y|x)$$

The ID method considers that more dense regions of the query space bring more useful information, leveraging the instance uncertainty and its similarity to all the other instances in the pool:

$$ID(x) = Var(y|x) \times \left( \frac{1}{U} \sum_{u=1}^U sim(x, x^{(u)}) \right)^\beta$$

The  $\beta$  parameter controls the relative importance of the density term. In our experiments, we set it to 1, giving equal weights to variance and density. The  $U$  term is the number of instances in the query pool. As similarity measure  $sim(x, x^{(u)})$ , we use the cosine distance between the feature vectors. With each method, we choose the instance that maximises its respective equation.

#### 3.3 Experiments

To build our QE models, we extracted the 17 features used by the baseline approach in the WMT12 QE shared task.<sup>1</sup> These features were used with a Support Vector Regressor (SVR) with radial basis function and fixed hyperparameters ( $C=5$ ,  $\gamma=0.01$ ,  $\epsilon=0.5$ ), using the Scikit-learn toolkit (Pedregosa et al., 2011). For each dataset and each query method, we performed 20 active learning simulation experiments and averaged the results. We

<sup>1</sup>We refer the reader to (Callison-Burch et al., 2012) for a detailed description of the feature set, but this was a very strong baseline, with only five out of 19 participating systems outperforming it.



started with 50 randomly selected sentences from the training set and used all the remaining training sentences as our query pool, adding one new sentence to the training set at each iteration.

Results were evaluated by measuring Mean Absolute Error (MAE) scores on the test set. We also performed an “oracle” experiment: at each iteration, it selects the instance that minimises the MAE on the test set. The oracle results give an upper bound in performance for each test set.

Since an SVR does not supply variance values for its predictions, we employ a technique known as *query-by-bagging* (Abe and Mamitsuka, 1998). The idea is to build an ensemble of  $N$  SVRs trained on sub-samples of the training data. When selecting a new query, the ensemble is able to return  $N$  predictions for each instance, from where a variance value can be inferred. We used 20 SVRs as our ensemble and 20 as the size of each training sub-sample.<sup>2</sup> The variance values are then used as-is in the case of US strategy and combined with query densities in case of the ID strategy.

## 4 Results and Discussion

Figure 1 shows the learning curves for all query methods and all datasets. The “random” curves are our baseline since they are equivalent to passive learning (with various numbers of instances). We first evaluated our methods in terms of how many instances they needed to achieve 99% of the MAE score on the full dataset. For three datasets, the AL methods significantly outperformed the random selection baseline, while no improvement was observed on the *ar-en-1* dataset. Results are summarised in Table 1.

The learning curves in Figure 1 show an interesting behaviour for most AL methods: some of them were able to yield lower MAE scores than models trained on the full dataset. This is particularly interesting in the *fr-en* case, where both methods were able to obtain better scores using only  $\sim 25\%$  of the available instances, with the US method resulting in 0.03 improvement. The random selection strategy performs surprisingly well (for some datasets it is better than the AL strategies with certain number of instances), providing extra evidence that much smaller annotated

<sup>2</sup>We also tried sub-samples with the same size of the current training data but this had a large impact in the query methods running time while not yielding significantly better results.

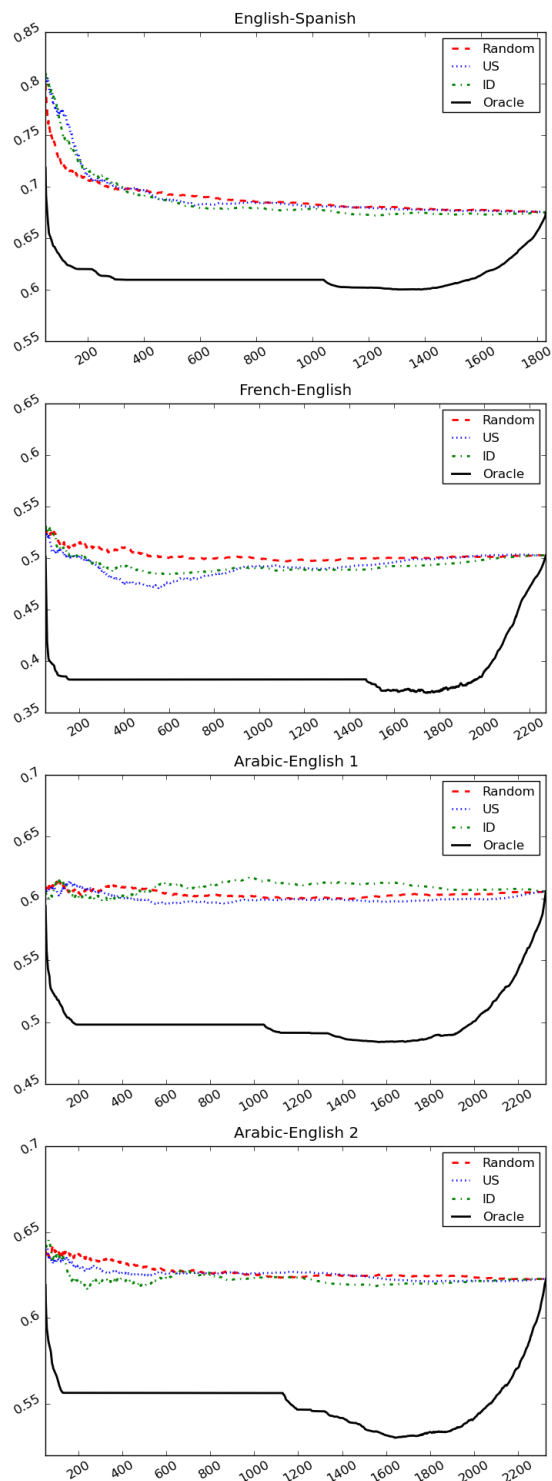


Figure 1: Learning curves for different query selection strategies in the four datasets. The horizontal axis shows the number of instances in the training set and the vertical axis shows MAE scores.

|         | US             |        | ID               |        | Random     |        | Full dataset |
|---------|----------------|--------|------------------|--------|------------|--------|--------------|
|         | #instances     | MAE    | #instances       | MAE    | #instances | MAE    |              |
| en-es   | 959 (52%)      | 0.6818 | <b>549 (30%)</b> | 0.6816 | 1079 (59%) | 0.6818 | 0.6750       |
| fr-en   | <b>79 (3%)</b> | 0.5072 | 134 (6%)         | 0.5077 | 325 (14%)  | 0.5070 | 0.5027       |
| ar-en-1 | 51 (2%)        | 0.6067 | 51 (2%)          | 0.6052 | 51 (2%)    | 0.6061 | 0.6058       |
| ar-en-2 | 209 (9%)       | 0.6288 | <b>148 (6%)</b>  | 0.6289 | 532 (23%)  | 0.6288 | 0.6290       |

Table 1: Number (proportion) of instances needed to achieve 99% of the performance of the full dataset. Bold-faced values indicate the best performing datasets.

|         | Best MAE US |               |            | Best MAE ID |               |            | Full dataset |
|---------|-------------|---------------|------------|-------------|---------------|------------|--------------|
|         | #instances  | MAE US        | MAE Random | #instances  | MAE ID        | MAE Random |              |
| en-es   | 1832 (100%) | 0.6750        | 0.6750     | 1122 (61%)  | <b>0.6722</b> | 0.6807     | 0.6750       |
| fr-en   | 559 (25%)   | <b>0.4708</b> | 0.5010     | 582 (26%)   | 0.4843        | 0.5008     | 0.5027       |
| ar-en-1 | 610 (26%)   | <b>0.5956</b> | 0.6042     | 351 (15%)   | 0.5987        | 0.6102     | 0.6058       |
| ar-en-2 | 1782 (77%)  | 0.6212        | 0.6242     | 190 (8%)    | <b>0.6170</b> | 0.6357     | 0.6227       |

Table 2: Best MAE scores obtained in the AL experiments. For each method, the first column shows the number (proportion) of instances used to obtain the best MAE, the second column shows the MAE score obtained and the third column shows the MAE score for random instance selection at the same number of instances. The last column shows the MAE obtained using the full dataset. Best scores are shown in bold and are significantly better (paired t-test,  $p < 0.05$ ) than both their randomly selected counterparts and the full dataset MAE.

datasets than those used currently can be sufficient for machine translation QE.

The best MAE scores achieved for each dataset are shown in Table 2. The figures were tested for significance using pairwise t-test with 95% confidence,<sup>3</sup> with bold-faced values in the table indicating significantly better results.

The lower bounds in MAE given by the oracle curves show that AL methods can indeed improve the performance of QE models: an ideal query method would achieve a very large improvement in MAE using fewer than 200 instances in all datasets. The fact that different datasets present similar oracle curves suggests that this is not related for a specific dataset but actually a common behaviour in QE. Although some of this gain in MAE may be due to overfitting to the test set, the results obtained with the *fr-en* and *ar-en-2* datasets are very promising, and therefore we believe that it is possible to use AL to improve QE results in other cases, as long as more effective query techniques are designed.

## 5 Further analysis on the oracle behaviour

By analysing the oracle curves we can observe another interesting phenomenon which is the rapid increase in error when reaching the last  $\sim 200$  instances of the training data. A possible explana-

<sup>3</sup>We took the average of the MAE scores obtained from the 20 runs with each query method for that.

tion for this behaviour is the existence of erroneous, inconsistent or contradictory labels in the datasets. Quality annotation is a subjective task by nature, and it is thus subject to noise, e.g., due to misinterpretations or disagreements. Our hypothesis is that these last sentences are the most difficult to annotate and therefore more prone to disagreements.

To investigate this phenomenon, we performed an additional experiment with the *en-es* dataset, the only dataset for which multiple annotations are available (from three judges). We measure the Kappa agreement index (Cohen, 1960) between all pairs of judges in the subset containing the first 300 instances (the 50 initial random instances plus 250 instances chosen by the oracle). We then measured Kappa in windows of 300 instances until the last instance of the training set is selected by the oracle method. We also measure variances in sentence length using windows of 300 instances. The idea of this experiment is to test whether sentences that are more difficult to annotate (because of their length or subjectivity, generating more disagreement between the judges) add noise to the dataset.

The resulting Kappa curves are shown in Figure 2: the agreement between judges is high for the initial set of sentences selected, tends to decrease until it reaches  $\sim 1000$  instances, and then starts to increase again. Figure 3 shows the results for source sentence length, which follow the same trend (in a reversed manner). Contrary to our hy-

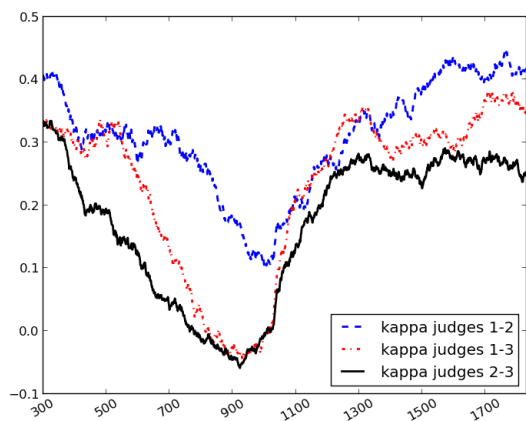


Figure 2: Kappa curves for the en-es dataset. The horizontal axis shows the number of instances and the vertical axis shows the kappa values. Each point in the curves shows the kappa index for a window containing the last 300 sentences chosen by the oracle.

pothesis, these results suggest that the most difficult sentences chosen by the oracle are those in the middle range instead of the last ones. If we compare this trend against the oracle curve in Figure 1, we can see that those middle instances are the ones that do not change the performance of the oracle.

The resulting trends are interesting because they give evidence that sentences that are difficult to annotate do not contribute much to QE performance (although not hurting it either). However, they do not confirm our hypothesis about the oracle behaviour. Another possible source of disagreement is the feature set: the features may not be discriminative enough to distinguish among different instances, i.e., instances with very similar features but different labels might be genuinely different, but the current features are not sufficient to indicate that. In future work we plan to further investigate this by hypothesis by using other feature sets and analysing their behaviour.

## 6 Conclusions and Future Work

We have presented the first known experiments using active learning for the task of estimating machine translation quality. The results are promising: we were able to reduce the number of instances needed to train the models in three of the four datasets. In addition, in some of the datasets active learning yielded significantly better models using only a small subset of the training instances.

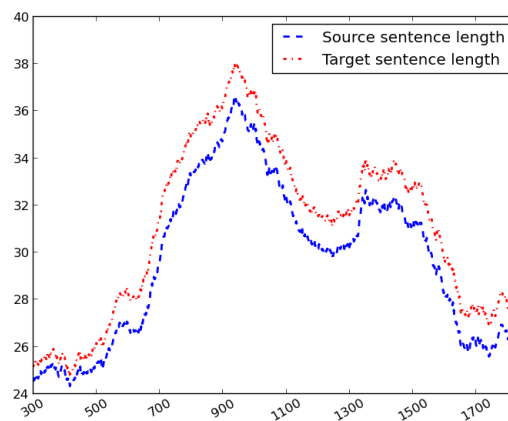


Figure 3: Average source and target sentence lengths for the en-es dataset. The horizontal axis shows the number of instances and the vertical axis shows the length values. Each point in the curves shows the average length for a window containing the last 300 sentences chosen by the oracle.

The oracle results give evidence that it is possible to go beyond these encouraging results by employing better selection strategies in active learning. In future work we will investigate more advanced query techniques that consider features other than variance and density of the data points. We also plan to further investigate the behaviour of the oracle curves using not only different feature sets but also different quality scores such as HTER and post-editing time. We believe that a better understanding of this behaviour can guide further developments not only for instance selection techniques but also for the design of better quality features and quality annotation schemes.

## Acknowledgments

This work was supported by funding from CNPq/Brazil (No. 237999/2012-9, Daniel Beck) and from the EU FP7-ICT QTLaunchPad project (No. 296347, Lucia Specia).

## References

- Naoki Abe and Hiroshi Mamitsuka. 1998. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 1–9.
- Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of EMNLP*, pages 9–16.

- John Blatz, Erin Fitzgerald, and George Foster. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th Conference on Computational Linguistics*, pages 315–321.
- 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 7th Workshop on Statistical Machine Translation*.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 128–132.
- 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 ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.
- David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–10.
- Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing. Technical report.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Duborg, 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.
- Chris Quirk. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of LREC*, pages 825–828.
- Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079.
- Burr Settles. 2010. Active learning literature survey. Technical report.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.
- Lucia Specia and Atefeh Farzindar. 2010. Estimating machine translation post-editing effort with HTER. In *Proceedings of AMTA Workshop Bringing MT to the User: MT Research and the Translation Industry*.
- Lucia Specia, M Turchi, Zhuoran Wang, and J Shawe-Taylor. 2009. Improving the confidence of machine translation quality estimates. In *Proceedings of MT Summit XII*.
- Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *Proceedings of MT Summit XIII*.
- Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of EAMT*.
- Andreas Vlachos. 2006. Active annotation. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining at EACL*.

# Reranking with Linguistic and Semantic Features for Arabic Optical Character Recognition

Nadi Tomeh, Nizar Habash, Ryan Roth, Noura Farra  
Center for Computational Learning Systems, Columbia University  
{nadi, habash, ryanr, noura}@ccls.columbia.edu

Pradeep Dasigi  
Safaba Translation Solutions  
pradeep@safaba.com

Mona Diab  
The George Washington University  
mtdiab@gwu.edu

## Abstract

Optical Character Recognition (OCR) systems for Arabic rely on information contained in the scanned images to recognize sequences of characters and on language models to emphasize fluency. In this paper we incorporate linguistically and semantically motivated features to an existing OCR system. To do so we follow an  $n$ -best list reranking approach that exploits recent advances in learning to rank techniques. We achieve 10.1% and 11.4% reduction in recognition word error rate (WER) relative to a standard baseline system on typewritten and handwritten Arabic respectively.

## 1 Introduction

Optical Character Recognition (OCR) is the task of converting scanned images of handwritten, typewritten or printed text into machine-encoded text. Arabic OCR is a challenging problem due to Arabic's connected letter forms, consonantal diacritics and rich morphology (Habash, 2010). Therefore only a few OCR systems have been developed (Märgner and Abed, 2009). The BBN Byblos OCR system (Natajan et al., 2002; Prasad et al., 2008; Saleem et al., 2009), which we use in this paper, relies on a hidden Markov model (HMM) to recover the sequence of characters from the image, and uses an  $n$ -gram language model (LM) to emphasize the fluency of the output. For an input image, the OCR decoder generates an  $n$ -best list of hypotheses each of which is associated with HMM and LM scores.

In addition to fluency as evaluated by LMs, other information potentially helps in discriminating good from bad hypotheses. For example, Habash and Roth (2011) use a variety of linguistic (morphological and syntactic) and non-linguistic features to automatically identify errors in OCR

hypotheses. Another example presented by Devlin et al. (2012) shows that using a statistical machine translation system to assess the difficulty of translating an Arabic OCR hypothesis into English gives valuable feedback on OCR quality. Therefore, combining additional information with the LMs could reduce recognition errors. However, direct integration of such information in the decoder is difficult.

A straightforward alternative which we advocate in this paper is to use the available information to rerank the hypotheses in the  $n$ -best lists. The new top ranked hypothesis is considered as the new output of the system. We propose combining LMs with linguistically and semantically motivated features using learning to rank methods. Discriminative reranking allows each hypothesis to be represented as an arbitrary set of features without the need to explicitly model their interactions. Therefore, the system benefits from global and potentially complex features which are not available to the baseline OCR decoder. This approach has successfully been applied in numerous Natural Language Processing (NLP) tasks including syntactic parsing (Collins and Koo, 2005), semantic parsing (Ge and Mooney, 2006), machine translation (Shen et al., 2004), spoken language understanding (Dinarelli et al., 2012), etc. Furthermore, we propose to combine several ranking methods into an ensemble which learns from their predictions to further reduce recognition errors.

We describe our features and reranking approach in §2, and we present our experiments and results in §3.

## 2 Discriminative Reranking for OCR

Each hypothesis in an  $n$ -best list  $\{h_i\}_{i=1}^n$  is represented by a  $d$ -dimensional feature vector  $\mathbf{x}_i \in \mathbb{R}^d$ . Each  $\mathbf{x}_i$  is associated with a loss  $l_i$  to generate a labeled  $n$ -best list  $H = \{(\mathbf{x}_i, l_i)\}_{i=1}^n$ . The loss is computed as the Word Error Rate (WER) of the

hypotheses compared to a reference transcription. For supervised training we use a set of  $n$ -best lists  $\mathcal{H} = \{H^{(k)}\}_{k=1}^M$ .

## 2.1 Learning to rank approaches

Major approaches to learning to rank can be divided into pointwise score regression, pairwise preference satisfaction, and listwise structured learning. See Liu (2009) for a survey. In this paper, we explore all of the following learning to rank approaches.

**Pointwise** In the pointwise approach, the ranking problem is formulated as a regression, or ordinal classification, for which any existing method can be applied. Each hypothesis constitutes a learning instance. In this category we use a regression method called Multiple Additive Regression Trees (MART) (Friedman, 2000) as implemented in RankLib.<sup>1</sup> The major problem with pointwise approaches is that the structure of the list of hypotheses is ignored.

**Pairwise** The pairwise approach takes pairs of hypotheses as instances in learning, and formalizes the ranking problem as a pairwise classification or pairwise regression. We use several methods from this category.

*RankSVM* (Joachims, 2002) is a method based on Support Vector Machines (SVMs) for which we use only linear kernels to keep complexity low. Exact optimization of the RankSVM objective can be computationally expensive as the number of hypothesis pairs can be very large. Approximate stochastic training strategies reduces complexity and produce comparable performance. Therefore, in addition to RankSVM, we use stochastic sub-gradient descent (*SGDSVM*), Pegasos (*PegasosSVM*) and Passive-Aggressive Perceptron (*PAPSVM*) as implemented in Sculley (2009).<sup>2</sup>

*RankBoost* (Freund et al., 2003) is a pairwise boosting approach implemented in RankLib. It uses a linear combination of *weak* rankers, each of which is a binary function associated with a single feature. This function is 1 when the feature value exceeds some threshold and 0 otherwise.

*RankMIRA* is a ranking method presented in (Le Roux et al., 2012).<sup>3</sup> It uses a weighted linear combination of features which assigns the highest

score to the hypotheses with the lowest loss. During training, the weights are updated according to the Margin-Infused Relaxed Algorithm (MIRA), whenever the highest scoring hypothesis differs from the hypothesis with the lowest error rate.

In pairwise approaches, the group structure of the  $n$ -best list is still ignored. Additionally, the number of training pairs generated from an  $n$ -best list depends on its size, which could result in training a model biased toward larger hypothesis lists (Cao et al., 2006).

**Listwise** The listwise approach takes  $n$ -best lists as instances in both learning and prediction. The group structure is considered explicitly and ranking evaluation measures can be directly optimized. The listwise methods we use are implemented in RankLib.

*AdaRank* (Xu and Li, 2007) is a boosting approach, similar to RankBoost, except that it optimizes an arbitrary ranking metric, for which we use Mean Average Precision (MAP).

*Coordinate Ascent (CA)* uses a listwise linear model whose weights are learned by a coordinate ascent method to optimize a ranking metric (Metzler and Bruce Croft, 2007). As with *AdaRank* we use MAP.

*ListNet* (Cao et al., 2007) uses a neural network model whose parameters are learned by gradient descent method to optimize a listwise loss based on a probabilistic model of permutations.

## 2.2 Ensemble reranking

In addition to the above mentioned approaches, we couple simple feature selection and reranking models combination via a straightforward ensemble learning method similar to *stacked generalization* (Wolpert, 1992) and *Combiner* (Chan and Stolfo, 1993). Our goal is to generate an overall *meta-ranker* that outperforms all *base-rankers* by learning from their predictions how they correlate with each other.

To obtain the base-rankers, we train each of the ranking models of §2.1 using all the features of §2.3 and also using each feature family added to the baseline features separately. Then, we use the best model for each ranking approach to make predictions on a held-out data set of  $n$ -best lists. We can think of each base-ranker as computing one feature for each hypothesis. Hence, the scores generated by all the rankers for a given hypothesis constitute its feature vector.

The held-out  $n$ -best lists and the predictions of

<sup>1</sup><http://people.cs.umass.edu/~vdang/ranklib.html>

<sup>2</sup><http://code.google.com/p/sofia-ml>

<sup>3</sup><https://github.com/jihelhere/adMIRABLE>

the base-rankers represent the training data for the meta-ranker. We choose RankSVM<sup>4</sup> as the meta-ranker since it performed well as a base-ranker.

### 2.3 Features

Our features fall into five families.

**Base** features include the HMM and LM scores produced by the OCR system. These features are used by the baseline system<sup>5</sup> as well as by the various reranking methods.

**Simple** features (“simple”) include the baseline rank of the hypothesis and a 0-to-1 range normalized version of it. We also use a hypothesis confidence feature which corresponds to the average of the confidence of individual words in the hypothesis; “confidence” for a given word is computed as the fraction of hypotheses in the  $n$ -best list that contain the word (Habash and Roth, 2011). The more consensus words a hypothesis contains, the higher its assigned confidence. We also use the average word length and the number of content words (normalized by the hypothesis length). We define “content words” as non-punctuation and non-digit words. Additionally, we use a set of binary features indicating if the hypothesis contains a sequence of duplicated characters, a date-like sequence and an occurrence of a specific character class (punctuation, alphabetic and digit).

**Word LM** features (“LM-word”) include the log probabilities of the hypothesis obtained using  $n$ -gram LMs with  $n \in \{1, \dots, 5\}$ . Separate LMs are trained on the Arabic Gigaword 3 corpus (Graff, 2007), and on the reference transcriptions of the training data (see §3.1). The LM models are built using the SRI Language Modeling Toolkit (Stolcke, 2002).

**Linguistic LM** features (“LM-MADA”) are similar to the word LM features except that they are computed using the part-of-speech and the lemma of the words instead of the actual words.<sup>6</sup>

**Semantic coherence** feature (“SemCoh”) is motivated by the fact that semantic information can be very useful in modeling the fluency of phrases, and can augment the information provided by  $n$ -gram LMs. In modeling contextual

lexical semantic information, simple bag-of-words models usually have a lot of noise; while more sophisticated models considering positional information have sparsity issues. To strike a balance between these two extremes, we introduce a novel model of semantic coherence that is based on a measure of semantic relatedness between pairs of words. We model semantic relatedness between two words using the Information Content (IC) of the pair in a method similar to the one used by Lin (1997) and Lin (1998).

$$IC(w_1, d, w_2) = \log \frac{f(w_1, d, w_2)f(*, d, *)}{f(w_1, d, *)f(*, d, w_2)}$$

Here,  $d$  can generally represent some form of relation between  $w_1$  and  $w_2$ . Whereas Lin (1997) and Lin (1998) used dependency relation between words, we use distance. Given a sentence, the distance between  $w_1$  and  $w_2$  is one plus the number of words that are seen after  $w_1$  and before  $w_2$  in that sentence. Hence,  $f(w_1, d, w_2)$  is the number of times  $w_1$  occurs before  $w_2$  at a distance  $d$  in all the sentences in a corpus.  $*$  is a placeholder for any word, i.e.,  $f(*, d, *)$  is the frequency of all word pairs occurring at distance  $d$ . The distances are directional and not absolute values. A similar measure of relatedness was also used by Kolb (2009).

We estimate the frequencies from the Arabic Gigaword. We set the window size to 3 and calculate IC values of all pairs of words occurring at distance within the window size. Since the distances are directional, it has to be noted that given a word, its relations with three words before it and three words after it are modeled. During testing, for each phrase in our test set, we measure semantic relatedness of pairs of words using the IC values estimated from the Arabic Gigaword, and normalize their sum by the number of pairs in the phrase to obtain a measure of Semantic Coherence (SC) of the phrase. That is,

$$SC(p) = \frac{1}{m} \times \sum_{\substack{1 \leq d \leq W \\ 1 \leq i+d < n}} IC(w_i, d, w_{i+d})$$

where  $p$  is the phrase being evaluated,  $n$  is the number of words in it,  $d$  is the distance between words,  $W$  is the window size (set to 3), and  $m$  is the number of all possible  $w_i, w_{i+d}$  pairs in the phrase given these conditions.

<sup>4</sup>RankSVM has also been shown to be a good choice for the meta-learner in general stacking ensemble learning (Tang et al., 2010).

<sup>5</sup>The baseline ranking is simply based on the sum of the logs of the HMM and LM scores.

<sup>6</sup>The part-of-speech and the lemmas are obtained using MADA 3.0, a tool for Arabic morphological analysis and disambiguation (Habash and Rambow, 2005; Habash et al., 2009).

|                 | print             |           |                  | hand              |           |                  |
|-----------------|-------------------|-----------|------------------|-------------------|-----------|------------------|
|                 | $ \mathcal{H}_* $ | $\bar{n}$ | $\overline{ h }$ | $ \mathcal{H}_* $ | $\bar{n}$ | $\overline{ h }$ |
| $\mathcal{H}_b$ | 1,560             | 62        | 9                | 2,295             | 225       | 8                |
| $\mathcal{H}_m$ | 1,000             | 76        | 9                | 1,000             | 225       | 9                |
| $\mathcal{H}_t$ | 1,000             | 64        | 9                | 1,000             | 227       | 9                |

Table 1: Data sets statistics.  $|\mathcal{H}_*|$  refers to the number of  $n$ -best lists,  $\bar{n}$  is the average size of the lists, and  $\overline{|h|}$  is the average length of a hypothesis.

|             | print | hand  |
|-------------|-------|-------|
| Baseline    | 13.8% | 35%   |
| Oracle      | 9.8%  | 20.9% |
| Best result | 12.4% | 30.9% |

Table 2: WER for baseline, oracle and best reranked hypotheses.

### 3 Experiments

#### 3.1 Data and baselines

We used two data sets derived from high-resolution image scans of *typewritten* and *handwritten* Arabic text along with ground truth transcriptions.<sup>7</sup> The BBN Byblos system was then used to process these scanned images into sequences of segments (sentence fragments) and generate a ranked  $n$ -best list of hypotheses for each segment (Natajan et al., 2002; Prasad et al., 2008; Saleem et al., 2009). We divided each of the typewritten data set (“print”) and handwritten data set (“hand”) into three disjoint parts: a training set for the base-rankers  $\mathcal{H}_b$ , a training set for the meta-ranker  $\mathcal{H}_m$  and a test set  $\mathcal{H}_t$ . Table 1 presents some statistics about these data sets. Our baseline is based on the sum of the logs of the HMM and LM scores. Table 2 presents the WER for our baseline hypothesis, the best hypothesis in the list (our oracle) and our best reranking results which we describe in details in §3.2.

For LM training we used 220M words from Arabic Gigaword 3, and 2.4M words from each “print” and “hand” ground truth annotations.

**Effect of  $n$ -best training size on WER** The size of the training  $n$ -best lists is crucial to the learning of the ranking model. In particular, it determines the number of training instances per list. To determine the optimal  $n$  to use for the rest of this paper, we conducted the following experiment aims to understand the effect of the size of  $n$ -best lists

<sup>7</sup>The Anfal data set discussed here was collected by the Linguistic Data Consortium.

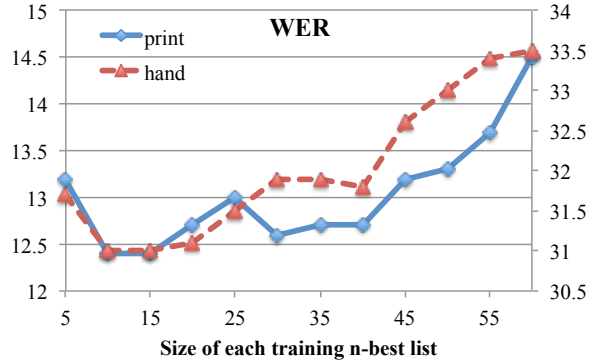


Figure 1: Effect of the size of training  $n$ -best lists on WER. The horizontal axis represents the maximum size of the  $n$ -best lists and the vertical axis represents WER, left is “print” and right is “hand”.

on the reranking performance for one of our best reranking models, namely RankSVM. We trained each model with different sizes of  $n$ -best, varying from  $n = 5$  to  $n = 60$  for “print” data, and between  $n = 5$  and  $n = 150$  for “hand” data. The top  $n$  hypotheses according to the baseline are selected for each  $n$ . Figure 1 plots WER as a function of the size of the training list  $n$  for both “print” and “hand” data.

The lowest WER scores are achieved for  $n = 10$  and  $n = 15$  for both “print” and “hand” data. We note that a small number of hypotheses per list is sufficient for RankSVM to obtain a good performance, but also increasing  $n$  further seems to increase the error rate. For the rest of this paper we use the top 10-best hypotheses per segment.

#### 3.2 Reranking results

The reranking results for “print” and “hand” are presented in Table 3. The results are presented as the difference in WER from the baseline WER. See the caption in Table 3 for more information.

For “print”, the pairwise approaches clearly outperform the listwise approaches and achieve the lowest WER of 12.4% (10.1% WER reduction relative to the baseline) with 7 different combinations of rankers and feature families. While both approaches do not minimize WER directly, the pairwise methods have the advantage of using objectives that are simpler to optimize, and they are trained on much larger number of examples which may explain their superiority. RankBoost, however, is less competitive with a performance closer to that of listwise approaches. All the methods improved over the baseline with any feature family, except for the pointwise approach which did



| Features     | Pointwise |         |         |      | Pairwise  |         |             |             |             |             |             |
|--------------|-----------|---------|---------|------|-----------|---------|-------------|-------------|-------------|-------------|-------------|
|              | MART      | AdaRank | ListNet | CA   | RankBoost | RankSVM | SGDSVM      | RankMIRA    | PegaSVM     | PAP SVM     |             |
| <b>Print</b> | Base      | 1.1     | -0.4    | -1.0 | -1.0      | -1.0    | -1.1        | -1.2        | -1.2        | -1.3        | -1.3        |
|              | +simple   | -0.1    | 0.0     | -0.1 | -0.2      | 0.0     | -0.1        | 0.1         | 0.0         | 0.1         | 0.0         |
|              | +LM-word  | -1.0    | -0.2    | 0.1  | -0.1      | -0.1    | <b>-0.3</b> | <b>-0.2</b> | -0.1        | 0.0         | <b>-0.1</b> |
|              | +LM-MADA  | 0.0     | -0.3    | 0.1  | -0.2      | -0.1    | 0.0         | -0.1        | <b>-0.2</b> | <b>-0.1</b> | <b>-0.1</b> |
|              | +SemCoh   | 0.0     | -0.4    | 0.0  | -0.2      | -0.1    | -0.1        | 0.0         | -0.1        | 0.0         | 0.1         |
|              | +All      | 0.6     | 0.1     | 0.0  | 0.1       | 0.0     | 0.1         | 0.2         | 0.2         | 0.2         | <b>0.0</b>  |
| <b>Hand</b>  | Base      | 4.2     | -3.1    | -3.2 | -3.4      | -2.9    | -3.2        | -3.5        | -3.8        | -3.6        | -3.8        |
|              | +simple   | 0.3     | -0.1    | 0.1  | 0.2       | 0.1     | -0.1        | 0.2         | <b>-0.2</b> | 0.1         | 0.2         |
|              | +LM-word  | 0.4     | -0.1    | 0.1  | 0.8       | -0.2    | -0.7        | -0.2        | -0.1        | 0.0         | 0.1         |
|              | +LM-MADA  | 0.0     | -0.5    | 0.1  | 0.0       | 0.1     | -0.4        | -0.1        | 0.3         | -0.2        | 0.1         |
|              | +SemCoh   | 0.0     | -0.1    | 0.0  | -0.4      | 0.0     | -0.2        | -0.3        | <b>-0.2</b> | -0.2        | 0.0         |
|              | +All      | 0.2     | 0.4     | 0.0  | 0.4       | 0.2     | 0.4         | 0.2         | 0.1         | 0.2         | 0.0         |

Table 3: Reranking results for the “print” and “hand” data sets; the “print” baseline WER is 13.9% and the “hand” baseline WER is 35.0%. The “Base” numbers represent the difference in WER between the corresponding ranker using “Base” features only and the baseline, which uses the same “Base” features. The “+features” numbers represent additional gain (relative to “Base”) obtained by adding the corresponding feature family. The “+All” numbers represent the gain of using all features, relative to the best single-family system. The actual WER of a ranker can be obtained by summing the baseline WER and the corresponding “Base” and “+features” scores. Bolded values are the best performers overall.

worse than the baseline. When combined with the “Base” features, “LM-words” lead to improvements with 8 out of 10 rankers, and proved to be the most helpful among feature families. “LM-MADA” follows with improvements with 7 out of 10 rankers. The lowest WER is achieved using one of these two LM-based families. Combining all feature families did not help and in many cases resulted in a higher WER than the best family.

Similar improvements are observed for “hand”. The lowest achieved WER is 31% (11.4% WER reduction relative to the baseline). Here also, the pointwise method increased the WER by 12% relative to the baseline (as opposed to 7% for “print”). Again, the listwise approaches are overall less effective than their pairwise counterparts, except for RankBoost which resulted in the smallest WER reduction among all rankers. The two best rankers correspond to RankMIRA with the “simple” and the “SemCoh” features. The “SemCoh” feature resulted in improvements for 6 out of the 10 rankers, and thus was the best single feature on average for the “hand” data set. As observed with “print” data, combining all the features does not lead to the best performance.

In an additional experiment, we selected the best model for each ranking method and combined them to build an ensemble as described in §2.2. For “hand”, the ensemble slightly outperformed all the individual rankers and achieved the lowest WER of 30.9%. However, for the “print” data, the

ensemble failed to improve over the base-rankers and resulted in a WER of 12.4%.

The best overall results are presented in Table 2. Our best results reduce the distance to the oracle top line by 35% for “print” and 29% for “hand”.

## 4 Conclusion

We presented a set of experiments on incorporating features into an existing OCR system via  $n$ -best list reranking. We compared several learning to rank techniques and combined them using an ensemble technique. We obtained 10.1% and 11.4% reduction in WER relative to the baseline for “print” and “hand” data respectively. Our best systems used pairwise reranking which outperformed the other methods, and used the MADA based features for “print” and our novel semantic coherence feature for “hand”.

## Acknowledgment

We would like to thank Rohit Prasad and Matin Kamali for providing the data and helpful discussions. This work was funded under DARPA project number HR0011-08-C-0004. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA. The last two authors, Dasigi and Diab, worked on this project while at Columbia University.

## References

- Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 186–193.
- Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136.
- Philip K. Chan and Salvatore J. Stolfo. 1993. Experiments on multistrategy learning by meta-learning. In *Proceedings of the second international conference on Information and knowledge management*, CIKM '93, pages 314–323.
- Michael Collins and Terry Koo. 2005. Discriminative Reranking for Natural Language Parsing. *Comput. Linguist.*, 31(1):25–70, March.
- Jacob Devlin, Matin Kamali, Krishna Subramanian, Rohit Prasad, and Prem Natarajan. 2012. Statistical Machine Translation as a Language Model for Handwriting Recognition. In *ICFHR*, pages 291–296.
- Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2012. Discriminative Reranking for Spoken Language Understanding. *IEEE Transactions on Audio, Speech & Language Processing*, 20(2):526–539.
- Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December.
- Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29:1189–1232.
- Ruifang Ge and Raymond J. Mooney. 2006. Discriminative Reranking for Semantic Parsing. In *ACL*.
- David Graff. 2007. Arabic Gigaword 3, LDC Catalog No.: LDC2003T40. Linguistic Data Consortium, University of Pennsylvania.
- Nizar Habash and Owen Rambow. 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, June.
- Nizar Habash and Ryan M. Roth. 2011. Using deep morphology to improve automatic error detection in Arabic handwriting recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 875–884.
- 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, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. The MEDAR Consortium, April.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142.
- Peter Kolb. 2009. Experiments on the difference between semantic similarity and relatedness. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, NEALT Proceedings Series Vol. 4.
- Joseph Le Roux, Benoit Favre, Alexis Nasr, and Seyed Abolghasem Mirroshandel. 2012. Generative Constituent Parsing and Discriminative Dependency Reranking: Experiments on English and French. In *SP-SEM-MRL 2012*.
- Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, EACL '97, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98.
- Tie-Yan Liu. 2009. *Learning to Rank for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA.
- Volker Märgner and Haikal El Abed. 2009. Arabic Word and Text Recognition - Current Developments. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.
- Donald Metzler and W. Bruce Croft. 2007. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, June.
- Premkumar Natarajan, Zhidong Lu, Richard Schwartz, Issam Bazzi, and John Makhoul. 2002. Hidden Markov models. chapter Multilingual machine printed OCR, pages 43–63. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

- Rohit Prasad, Shirin Saleem, Matin Kamali, Ralf Meier, and Premkumar Natarajan. 2008. Improvements in hidden Markov model based Arabic OCR. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1–4.
- Shirin Saleem, Huaigu Cao, Krishna Subramanian, Matin Kamali, Rohit Prasad, and Prem Natarajan. 2009. Improvements in BBN’s HMM-Based Offline Arabic Handwriting Recognition System. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, IC-DAR '09*, pages 773–777.
- D. Sculley. 2009. Large scale learning to rank. In *NIPS 2009 Workshop on Advances in Ranking*.
- Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative Reranking for Machine Translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 177–184, Boston, Massachusetts, USA, May.
- Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO.
- Buzhou Tang, Qingcai Chen, Xuan Wang, and Xiaolong Wang. 2010. Reranking for stacking ensemble learning. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I, ICONIP'10*, pages 575–584.
- David H. Wolpert. 1992. Original Contribution: Stacked generalization. *Neural Netw.*, 5(2):241–259, February.
- Jun Xu and Hang Li. 2007. AdaRank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07*, pages 391–398.

# Evolutionary Hierarchical Dirichlet Process for Timeline Summarization

**Jiwei Li**

School of Computer Science  
Cornell University  
Ithaca, NY, 14853  
jl3226@cornell.edu

**Sujian Li**

Laboratory of Computational Linguistics  
Peking University  
Beijing, P.R.China, 150001  
lisujian@pku.edu.cn

## Abstract

Timeline summarization aims at generating concise summaries and giving readers a faster and better access to understand the evolution of news. It is a new challenge which combines salience ranking problem with novelty detection. Previous researches in this field seldom explore the evolutionary pattern of topics such as birth, splitting, merging, developing and death. In this paper, we develop a novel model called Evolutionary Hierarchical Dirichlet Process (EHDP) to capture the topic evolution pattern in timeline summarization. In EHDP, time varying information is formulated as a series of HDPs by considering time-dependent information. Experiments on 6 different datasets which contain 3156 documents demonstrates the good performance of our system with regard to ROUGE scores.

## 1 Introduction

Faced with thousands of news articles, people usually try to ask the general aspects such as the beginning, the evolutionary pattern and the end. General search engines simply return the top ranking articles according to query relevance and fail to trace how a specific event goes. Timeline summarization, which aims at generating a series of concise summaries for news collection published at different epochs can give readers a faster and better access to understand the evolution of news.

The key of timeline summarization is how to select sentences which can tell readers the evolutionary pattern of topics in the event. It is very common that the themes of a corpus evolve over time, and topics of adjacent epochs usually exhibit strong correlations. Thus, it is important to model topics across different documents and over different time periods to detect how the events evolve.

The task of timeline summarization is firstly proposed by Allan et al. (2001) by extracting clusters of noun phrases and name entities. Chieu et al. (2004) built a similar system in unit of sentences with interest and burstiness. However, these methods seldom explored the evolutionary characteristics of news. Recently, Yan et al. (2011) extended the graph based sentence ranking algorithm used in traditional multi-document summarization (MDS) to timeline generation by projecting sentences from different time into one plane. They further explored the timeline task from the optimization of a function considering the combination of different respects such as relevance, coverage, coherence and diversity (Yan et al., 2011b). However, their approaches just treat timeline generation as a sentence ranking or optimization problem and seldom explore the topic information lied in the corpus.

Recently, topic models have been widely used for capturing the dynamics of topics via time. Many dynamic approaches based on LDA model (Blei et al., 2003) or Hierarchical Dirichlet Processes (HDP) (Teh et al., 2006) have been proposed to discover the evolving patterns in the corpus as well as the snapshot clusters at each time epoch (Blei and Lafferty, 2006; Chakrabarti et al., 2006; Wang and McCallum, 2007; Caron et al., 2007; Ren et al., 2008; Ahmed and Xing, 2008; Zhang et al., 2010).

In this paper, we propose EHDP: a evolutionary hierarchical Dirichlet process (HDP) model for timeline summarization. In EHDP, each HDP is built for multiple corpora at each time epoch, and the time dependencies are incorporated into epochs under the Markovian assumptions. Topic popularity and topic-word distribution can be inferred from a Chinese Restaurant Process (CRP). Sentences are selected into timelines by considering different aspects such as topic relevance, coverage and coherence. We built the evaluation sys-

tems which contain 6 real datasets and performance of different models is evaluated according to the ROUGE metrics. Experimental results demonstrate the effectiveness of our model .

## 2 EHDP for Timeline Summarization

### 2.1 Problem Formulation

Given a general query  $Q = \{w_{qi}\}_{i=1}^{Q_n}$ , we firstly obtain a set of query related documents. We notate different corpus as  $C = \{C^t\}_{t=1}^T$  according to their published time where  $C^t = \{D_{ti}\}_{i=1}^{N_t}$  denotes the document collection published at epoch  $t$ . Document  $D_i^t$  is formulated as a collection of sentences  $\{s_{ij}^t\}_{j=1}^{N_{ti}}$ . Each sentence is presented with a series of words  $s_{ij}^t = \{w_{ijl}^t\}_{l=1}^{N_{ij}^t}$  and associated with a topic  $\theta_{ij}^t$ .  $V$  denotes the vocabulary size. The output of the algorithm is a series of timelines summarization  $I = \{I^t\}_{t=1}^T$  where  $I^t \subset C^t$

### 2.2 EHDP

Our EHDP model is illustrated in Figure 2. Specifically, each corpus  $C^t$  is modeled as a HDP. These HDP shares an identical base measure  $G_0$ , which serves as an overall bookkeeping of overall measures. We use  $G_0^t$  to denote the base measure at each epoch and draw the local measure  $G_i^t$  for each document at time  $t$  from  $G_0^t$ . In EHDP, each sentence is assigned to an aspect  $\theta_{ij}^t$  with the consideration of words within current sentence.

To consider time dependency information in EHDP, we link all time specific base measures  $G_0^t$  with a temporal Dirichlet mixture model as follows:

$$G_0^t \sim DP(\gamma^t, \frac{1}{K}G_0 + \frac{1}{K} \sum_{\delta=0}^{\Delta} F(v, \delta) \cdot G_0^{t-\delta}) \quad (1)$$

where  $F(v, \delta) = \exp(-\delta/v)$  denotes the exponential kernel function that controls the influence of neighboring corpus.  $K$  denotes the normalization factor where  $K = 1 + \sum_{\delta=0}^{\Delta} F(v, \delta)$ .  $\Delta$  is the time width and  $\lambda$  is the decay factor. In Chinese Restaurant Process (CRP), each document is referred to a restaurant and sentences are compared to customers. Customers in the restaurant sit around different tables and each table  $b_{in}^t$  is associated with a dish (topic)  $\Psi_{in}^t$  according to the dish menu. Let  $m_{tk}$  denote the number of tables enjoying dish  $k$  in all restaurants at epoch  $t$ ,  $m_{tk} = \sum_{i=1}^{N_t} \sum_{n=1}^{N_{ib}^t} 1(\Psi_{in}^t = k)$ . We redefine

---



---

for each epoch  $t \in [1, T]$

1. draw global measure  $G_0^t \sim DP(\alpha, \frac{1}{K}G_0 + \frac{1}{K} \sum_{\delta=0}^{\Delta} F(v, \delta)G_0^{t-\delta})$
2. for each document  $D_i^t$  at epoch  $t$ ,
  - 2.1 draw local measure  $G_i^t \sim DP(\gamma, G_0^t)$
  - 2.2 for each sentence  $s_{ij}^t$  in  $D_i^t$ 
    - draw aspect  $\theta_{ij}^t \sim G_i^t$
    - for  $w \in s_{ij}^t$  draw  $w \sim f(w)|\theta_{ij}^t$

---



---

Figure 1: Generation process for EHDP

another parameter  $M_{tk}$  to incorporate time dependency into EHDP.

$$M_{tk} = \sum_{\delta=0}^{\Delta} F(v, \delta) \cdot m_{t-\delta, k} \quad (2)$$

Let  $n_{ib}^t$  denote the number of sentences sitting around table  $b$ , in document  $i$  at epoch  $t$ . In CRP for EHDP, when a new customer  $s_{ij}^t$  comes in, he can sit on the existing table with probability  $n_{ib}^t/(n_i^t-1+\gamma)$ , sharing the dish (topic)  $\Psi_{ib}^t$  served at that table or picking a new table with probability  $\gamma/(n_i^t-1+\gamma)$ . The customer has to select a dish from the global dish menu if he chooses a new table. A dish that has already been shared in the global menu would be chosen with probability  $M_k^t/(\sum_k M_k^t + \alpha)$  and a new dish with probability  $\alpha/(\sum_k M_k^t + \alpha)$ .

$$\begin{aligned} &\theta_{ij}^t | \theta_{i1}^t, \dots, \theta_{ij-1}^t, \alpha \sim \\ &\sum_{\phi_{tb}=\theta_{ij}^t} \frac{n_{ib}^t}{n_i^t-1+\gamma} \delta_{\phi_{jb}} + \frac{\gamma}{n_i^t-1+\gamma} \delta_{\phi_{jb}^{new}} \\ &\phi_{ti}^{new} | \phi, \alpha \sim \\ &\sum_k \frac{M_{tk}}{\sum_i M_{ti} + \alpha} \delta_{\phi_k} + \frac{\alpha}{\sum_i M_{ti} + \alpha} G_0 \end{aligned} \quad (3)$$

We can see that EHDP degenerates into a series of independent HDPs when  $\Delta = 0$  and one global HDP when  $\Delta = T$  and  $v = \infty$ , as discussed in Amred and Xings work (2008).

### 2.3 Sentence Selection Strategy

The task of timeline summarization aims to produce a summary for each time and the generated summary should meet criteria such as relevance , coverage and coherence (Li et al., 2009). To care for these three criteria, we propose a topic scoring algorithm based on Kullback-Leibler(KL) divergence. We introduce the decreasing logistic function  $\zeta(x) = 1/(1 + e^x)$  to map the distance into interval (0,1).

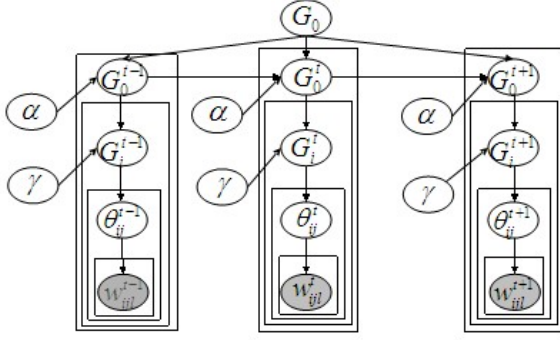


Figure 2: Graphical model of EHDP.

**Relevance:** the summary should be related with the proposed query  $Q$ .

$$F_R(I^t) = \zeta(KL(I^t||Q))$$

**Coverage:** the summary should highly generalize important topics mentioned in document collection at epoch  $t$ .

$$F_{Cv}(I^t) = \zeta(KL(I^t||C^t))$$

**Coherence:** News evolves over time and a good component summary is coherent with neighboring corpus so that a timeline tracks the gradual evolution trajectory for multiple correlative news.

$$F_{Ch}(I^t) = \frac{\sum_{\delta=-\Delta/2}^{\delta=\Delta/2} F(v, \delta) \cdot \zeta(KL(I^t||C^{t-\delta}))}{\sum_{\delta=-\Delta/2}^{\delta=\Delta/2} F(v, \delta)}$$

Let  $Score(I^t)$  denote the score of the summary and it is calculated in Equ.(4).

$$Score(I^t) = \lambda_1 F_R(I^t) + \lambda_2 F_{Cv}(I^t) + \lambda_3 F_{Ch}(I^t) \quad (4)$$

$\sum_i \lambda_i = 1$ . Sentences with higher score are selected into timeline. To avoid aspect redundancy, MMR strategy (Goldstein et al., 1999) is adopted in the process of sentence selection.

### 3 Experiments

#### 3.1 Experiments set-up

We downloaded 3156 news articles from selected sources such as BBC, New York Times and CNN with various time spans and built the evaluation systems which contains 6 real datasets. The news belongs to different categories of Rule of Interpretation (ROI) (Kumaran and Allan, 2004). Detailed statistics are shown in Table 1. Dataset 2(Deepwater Horizon oil spill), 3(Haiti Earthquake) and 5(Hurricane Sandy) are used as training data and

| New Source | Nation | News Source     | Nation |
|------------|--------|-----------------|--------|
| BBC        | UK     | New York Times  | US     |
| Guardian   | UK     | Washington Post | US     |
| CNN        | US     | Fox News        | US     |
| ABC        | US     | MSNBC           | US     |

Table 1: New sources of datasets

| News Subjects (Query)            | #docs | #epoch |
|----------------------------------|-------|--------|
| 1.Michael Jackson Death          | 744   | 162    |
| 2.Deepwater Horizon oil spill    | 642   | 127    |
| 3.Haiti Earthquake               | 247   | 83     |
| 4.American Presidential Election | 1246  | 286    |
| 5.Hurricane Sandy                | 317   | 58     |
| 6.Jerry Sandusky Sexual Abuse    | 320   | 74     |

Table 2: Detailed information for datasets

the rest are used as test data. Summary at each epoch is truncated to the same length of 50 words.

Summaries produced by baseline systems and ours are automatically evaluated through ROUGE evaluation metrics (Lin and Hovy, 2003). For the space limit, we only report three ROUGE ROUGE-2-F and ROUGE-W-F score. Reference timeline in ROUGE evaluation is manually generated by using Amazon Mechanical Turk<sup>1</sup>. Workers were asked to generate reference timeline for news at each epoch in less than 50 words and we collect 790 timelines in total.

#### 3.2 Parameter Tuning

To tune the parameters  $\lambda(i = 1, 2, 3)$  and  $v$  in our system, we adopt a gradient search strategy. We firstly fix  $\lambda_i$  to  $1/3$ . Then we perform experiments on with setting different values of  $v/\#epoch$  in the range from 0.02 to 0.2 at the interval of 0.02. We find that the Rouge score reaches its peak at round 0.1 and drops afterwards in the experiments. Next, we set the value of  $v$  is set to  $0.1 \cdot \#epoch$  and gradually change the value of  $\lambda_1$  from 0 to 1 with interval of 0.05, with simultaneously fixing  $\lambda_2$  and  $\lambda_3$  to the same value of  $(1 - \lambda_1)/2$ . The performance gets better as  $\lambda_1$  increases from 0 to 0.25 and then declines. Then we set the value of  $\lambda_1$  to 0.25 and change the value of  $\lambda_2$  from 0 to 0.75 with interval of 0.05. And the value of  $\lambda_2$  is set to 0.4, and  $\lambda_3$  is set to 0.35 correspondingly.

#### 3.3 Comparison with other topic models

In this subsection, we compare our model with 4 topic model baselines on the test data. *Stand-HDP(1)*: A topic approach that models different time epochs as a series of independent HDPs without considering time dependency. *Stand-HDP(2)*:

<sup>1</sup><http://mturk.com>

| System              | M.J. Death |       | US Election |       | S. Sexual Abuse |       |
|---------------------|------------|-------|-------------|-------|-----------------|-------|
|                     | R2         | RW    | R2          | RW    | R2              | RW    |
| <b>EHDP</b>         | 0.089      | 0.130 | 0.081       | 0.154 | 0.086           | 0.152 |
| <b>Stand-HDP(1)</b> | 0.080      | 0.127 | 0.075       | 0.134 | 0.072           | 0.138 |
| <b>Stand-HDP(2)</b> | 0.077      | 0.124 | 0.072       | 0.127 | 0.071           | 0.131 |
| <b>Dyn-LDA</b>      | 0.080      | 0.129 | 0.073       | 0.130 | 0.077           | 0.134 |
| <b>Stan-LDA</b>     | 0.072      | 0.117 | 0.065       | 0.122 | 0.071           | 0.121 |

Table 3: Comparison with topic models

| System          | M.J. Death |       | US Election |       | S. Sexual Abuse |       |
|-----------------|------------|-------|-------------|-------|-----------------|-------|
|                 | R2         | RW    | R2          | RW    | R2              | RW    |
| <b>EHDP</b>     | 0.089      | 0.130 | 0.081       | 0.154 | 0.086           | 0.152 |
| <b>Centroid</b> | 0.057      | 0.101 | 0.054       | 0.098 | 0.060           | 0.132 |
| <b>Manifold</b> | 0.053      | 0.108 | 0.060       | 0.111 | 0.069           | 0.128 |
| <b>ETS</b>      | 0.078      | 0.120 | 0.073       | 0.130 | 0.075           | 0.135 |
| <b>Chieu</b>    | 0.064      | 0.107 | 0.064       | 0.122 | 0.071           | 0.131 |

Table 4: Comparison with other baselines

A global HDP which models the whole time span as a restaurant. The third baseline, *Dynamic-LDA* is based on Blei and Laffery(2007)'s work and *Stan-LDA* is based on standard LDA model. In LDA based models, aspect number is predefined as 80<sup>2</sup>. Experimental results of different models are shown in Table 2. As we can see, EHDP achieves better results than the two standard HDP baselines where time information is not adequately considered. We also find an interesting result that Stan-HDP performs better than Stan-LDA. This is partly because new aspects can be automatically detected in HDP. As we know, how to determine topic number in the LDA-based models is still an open problem.

### 3.4 Comparison with other baselines

We implement several baselines used in traditional summarization or timeline summarization for comparison. (1) *Centroid* applies the MEAD algorithm (Radev et al., 2004) according to the features including centroid value, position and first-sentence overlap. (2) *Manifold* is a graph based unsupervised method for summarization, and the score of each sentence is got from the propagation through the graph (Wan et al., 2007). (3) *ETS* is the timeline summarization approach developed by Yan et al., (2011a), which is a graph based approach with optimized global and local biased summarization. (4) *Chieu* is the timeline system provided by (Chieu and Lee, 2004) utilizing interest and bursty ranking but neglecting trans-temporal news evolution. As we can see from Table 3, *Centroid* and *Manifold* get the worst results. This is probably because methods in multi-document summarization only care

<sup>2</sup>In our experiments, the aspect number is set as 50, 80, 100 and 120 respectively and we select the best performed result with the aspect number as 80

about sentence selection and neglect the novelty detection task. We can also see that EHDP under our proposed framework outputs existing timeline summarization approaches ETS and chieu. Our approach outputs Yan et al.,(2011a)s model by 6.9% and 9.3% respectively with regard to the average score of ROUGE-2-F and ROUGE-W-F.

## 4 Conclusion

In this paper we present an evolutionary HDP model for timeline summarization. Our EHDP extends original HDP by incorporating time dependencies and background information. We also develop an effective sentence selection strategy for candidate in the summaries. Experimental results on real multi-time news demonstrate the effectiveness of our topic model.

|   |
|---|
| Oct. 3, 2012  |
| S1: The first debate between President Obama and Mitt Romney, so long anticipated, quickly sunk into an unenlightening recitation of tired talking points and mendacity. S2: Mr. Romney wants to restore the Bush-era tax cut that expires at the end of this year and largely benefits the wealthy   |
| Oct. 11, 2012   |
| S1: The vice presidential debate took place on Thursday, October 11 at Kentucky's Centre College, and was moderated by Martha Raddatz. S2: The first and only debate between Vice President Joe Biden and Congressman Paul Ryan focused on domestic and foreign policy. The domestic policy segments included questions on health care, abortion          |
| Oct. 16, 2012   |
| S1. President Obama fights back in his second debate with Mitt Romney, banishing some of the doubts he raised in their first showdown. S2: The second debate dealt primarily with domestic affairs and include some segues into foreign policy. including taxes, unemployment, job creation, the national debt, energy and women's rights, both legal and |

Table 5: Selected timeline summarization generated by EHDP for American Presidential Election

## 5 Acknowledgement

This research has been supported by NSFC grants (No.61273278), National Key Technology RD Program (No:2011BAH1B0403), National 863 Program (No.2012AA011101) and National Social Science Foundation (No.12ZD227).

## References

Amr Ahmed and Eric Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process. 2008. In *SDM*.

- James Allan, Rahul Gupta and Vikas Khandelwal. Temporal summaries of new topics. 2001. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*
- David Blei, Andrew Ng and Micheal Jordan. 2003. Latent dirichlet allocation. In *Journal of Machine Learning Research*.
- David Blei and John Lafferty. Dynamic topic models. 2006. In *Proceedings of the 23rd international conference on Machine learning*.
- Francois Carol, Manuel Davy and Arnaud Doucet. Generalized poly urn for time-varying dirichlet process mixtures. 2007. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*.
- Deepayan Chakrabarti, Ravi Kumar and Andrew Tomkins. Evolutionary Clustering. In *Proceedings of the 12th ACM SIGKDD international conference Knowledge discovery and data mining*.
- Hai-Leong Chieu and Yoong-Keok Lee. Query based event extraction along a timeline. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*
- Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR04*.
- Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha and Yong Yu. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international conference on World wide web*.
- Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference of the NAACL. 2003*.
- Dragomar Radev, Hongyan. Jing, and Malgorzata Stys. 2004. Centroid-based summarization of multiple documents. In *Information Processing and Management*.
- Lu Ren, David Dunson and Lawrence Carin. The dynamic hierarchical Dirichlet process. 2008. In *Proceedings of the 25th international conference on Machine Learning*.
- Xiaojun Wan, Jianwu Yang and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of International Joint Conference on Artificial Intelligence*.
- Xuerui Wang and Andrew MaCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Yee Whye Teh, Michael Jordan, Matthew Beal and David Blei. Hierarchical Dirichlet Processes. In *American Statistical Association*.
- Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li and Yan Zhang. 2011a. Evolutionary Timeline Summarization: a Balanced Optimization Framework via Iterative Substitution. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*.
- Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Jahna Otterbacher, Xiaoming Li and Yan Zhang. Timeline Generation Evolutionary Trans-Temporal Summarization. 2011b. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Jianwen Zhang, Yangqiu Song, Changshui Zhang and Shixia Liu. 2010. Evolutionary Hierarchical Dirichlet Processes for Multiple Correlated Time-varying Corpora. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*.



# Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts

Gerasimos Lampouras and Ion Androutsopoulos

Department of Informatics

Athens University of Economics and Business

Patission 76, GR-104 34 Athens, Greece

<http://nlp.cs.aueb.gr/>

## Abstract

We present an ILP model of concept-to-text generation. Unlike pipeline architectures, our model jointly considers the choices in content selection, lexicalization, and aggregation to avoid greedy decisions and produce more compact texts.

## 1 Introduction

Concept-to-text natural language generation (NLG) generates texts from formal knowledge representations (Reiter and Dale, 2000). With the emergence of the Semantic Web (Antoniou and van Harmelen, 2008), interest in concept-to-text NLG has been revived and several methods have been proposed to express axioms of OWL ontologies (Grau et al., 2008) in natural language (Bontcheva, 2005; Mellish and Sun, 2006; Galanis and Androutsopoulos, 2007; Mellish and Pan, 2008; Schwitter et al., 2008; Schwitter, 2010; Liang et al., 2011; Williams et al., 2011).

NLG systems typically employ a pipeline architecture. They usually start by selecting the logical facts to express. The next stage, text planning, ranges from simply ordering the selected facts to complex decisions about the rhetorical structure of the text. Lexicalization then selects the words and syntactic structures that will realize each fact, specifying how each fact can be expressed as a single sentence. Sentence aggregation then combines sentences into longer ones. Another component generates appropriate referring expressions, and surface realization produces the final text.

Each stage of the pipeline is treated as a local optimization problem, where the decisions of the previous stages cannot be modified. This arrangement produces texts that may not be optimal, since the decisions of the stages have been shown to be co-dependent (Danlos, 1984; Marciniak and Strube, 2005; Belz, 2008). For example, content

selection and lexicalization may lead to more or fewer sentence aggregation opportunities.

We present an Integer Linear Programming (ILP) model that combines content selection, lexicalization, and sentence aggregation. Our model does not consider text planning, nor referring expression generation, which we hope to include in future work, but it is combined with an external simple text planner and a referring expression generation component; we also do not discuss surface realization. Unlike pipeline architectures, our model jointly examines the possible choices in the three NLG stages it considers, to avoid greedy local decisions. Given an individual (entity) or class of an OWL ontology and a set of facts (OWL axioms) about the individual or class, we aim to produce a text that expresses as many of the facts in as few words as possible. This is important when space is limited or expensive (e.g., product descriptions on smartphones, advertisements in search engines).

Although the search space of our model is very large and ILP problems are in general NP-hard, ILP solvers can be used, they are very fast in practice, and they guarantee finding a global optimum. Experiments show that our ILP model outperforms, in terms of compression, an NLG system that uses the same components, but connected in a pipeline, with no deterioration in fluency and clarity.

## 2 Related work

Marciniak and Strube (2005) propose a general ILP approach for language processing applications where the decisions of classifiers that consider particular, but co-dependent, subtasks need to be combined. They also show how their approach can be used to generate multi-sentence route directions, in a setting with very different inputs and processing stages than the ones we consider.

Barzilay and Lapata (2005) treat content selection as an optimization problem. Given a pool of facts and scores indicating the importance of each

fact or pair of facts, they select the facts to express by formulating an optimization problem similar to energy minimization. In other work, Barzilay and Lapata (2006) consider sentence aggregation. Given a set of facts that a content selection stage has produced, aggregation is viewed as the problem of partitioning the facts into optimal subsets. Sentences expressing facts that are placed in the same subset are aggregated to form a longer sentence. An ILP model is used to find the partitioning that maximizes the pairwise similarity of the facts in each subset, subject to constraints limiting the number of subsets and the facts in each subset.

Althaus et al. (2004) show that ordering a set of sentences to maximize sentence-to-sentence coherence is equivalent to the traveling salesman problem and, hence, NP-complete. They also show how an ILP solver can be used in practice.

Joint optimization ILP models have also been used in multi-document text summarization and sentence compression (McDonald, 2007; Clarke and Lapata, 2008; Berg-Kirkpatrick et al., 2011; Galanis et al., 2012; Woodsend and Lapata, 2012), where the input is text, not formal knowledge representations. Statistical methods to jointly perform content selection, lexicalization, and surface realization have also been proposed in NLG (Liang et al., 2009; Konstas and Lapata, 2012a; Konstas and Lapata, 2012b), but they are currently limited to generating single sentences from flat records.

To the best of our knowledge, this article is the first one to consider content selection, lexicalization, and sentence aggregation as an ILP joint optimization problem in the context of multi-sentence concept-to-text generation. It is also the first article to consider ILP in NLG from OWL ontologies.

### 3 Our ILP model of NLG

Let  $F = \{f_1, \dots, f_n\}$  be the set of all the facts  $f_i$  (OWL axioms) about the individual or class to be described. OWL axioms can be represented as sets of RDF triples of the form  $\langle S, R, O \rangle$ , where  $S$  is an individual or class,  $O$  is another individual, class, or datatype value, and  $R$  is a relation (property) that connects  $S$  to  $O$ . Hence, we can assume that each fact  $f_i$  is a triple  $\langle S_i, R_i, O_i \rangle$ .<sup>1</sup>

For each fact  $f_i$ , a set  $P_i = \{p_{i1}, p_{i2}, \dots\}$  of alternative sentence plans is available. Each

sentence plan  $p_{ik}$  specifies how to express  $f_i = \langle S_i, R_i, O_i \rangle$  as an alternative single sentence. In our work, a sentence plan is a sequence of slots, along with instructions specifying how to fill the slots in; and each sentence plan is associated with the relations it can express. For example,  $\langle \text{exhibit12}, \text{foundIn}, \text{athens} \rangle$  could be expressed using a sentence plan like “[ $\text{ref}(S)$ ] [ $\text{find}_{\text{past}}$ ] [ $\text{in}$ ] [ $\text{ref}(O)$ ]”, where square brackets denote slots,  $\text{ref}(S)$  and  $\text{ref}(O)$  are instructions requiring referring expressions for  $S$  and  $O$  in the corresponding slots, and “ $\text{find}_{\text{past}}$ ” requires the simple past form of “find”. In our example, the sentence plan would lead to a sentence like “Exhibit 12 was found in Athens”. We call *elements* the slots with their instructions, but with “ $S$ ” and “ $O$ ” accompanied by the individuals, classes, or datatype values they refer to; in our example, the elements are “[ $\text{ref}(S: \text{exhibit12})$ ]”, “[ $\text{find}_{\text{past}}$ ]”, “[ $\text{in}$ ]”, “[ $\text{ref}(O: \text{athens})$ ]”. Different sentence plans may lead to more or fewer aggregation opportunities; for example, sentences with the same verb are easier to aggregate. We use aggregation rules (Dalianis, 1999) that operate on sentence plans and usually lead to shorter texts.

Let  $s_1, \dots, s_m$  be disjoint subsets of  $F$ , each containing 0 to  $n$  facts, with  $m < n$ . A single sentence is generated for each subset  $s_j$  by aggregating the sentences (more precisely, the sentence plans) expressing the facts of  $s_j$ .<sup>2</sup> An empty  $s_j$  generates no sentence, i.e., the resulting text can be at most  $m$  sentences long. Let us also define:

$$a_i = \begin{cases} 1, & \text{if fact } f_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$l_{ikj} = \begin{cases} 1, & \text{if sentence plan } p_{ik} \text{ is used to express} \\ & \text{fact } f_i, \text{ and } f_i \text{ is in subset } s_j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$b_{tj} = \begin{cases} 1, & \text{if element } e_t \text{ is used in subset } s_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

and let  $B$  be the set of all the distinct elements (no duplicates) from all the available sentence plans that can express the facts of  $F$ . The length of an aggregated sentence resulting from a subset  $s_j$  can be roughly estimated by counting the distinct elements of the sentence plans that have been chosen to express the facts of  $s_j$ ; elements that occur more than once in the chosen sentence plans of  $s_j$

<sup>1</sup>We actually convert the RDF triples to simpler *message triples*, so that each message triple can be easily expressed by a simple sentence, but we do not discuss this conversion here.

<sup>2</sup>All the sentences of every possible subset  $s_j$  can be aggregated, because all the sentences share the same subject, the class or individual being described. If multiple aggregation rules apply, we use the one that leads to a shorter text.

are counted only once, because they will probably be expressed only once, due to aggregation.

Our objective function (4) maximizes the number of selected facts  $f_i$  and minimizes the number of distinct elements in each subset  $s_j$ , i.e., the approximate length of the corresponding aggregated sentence; an alternative explanation is that by minimizing the number of distinct elements in each  $s_j$ , we favor subsets that aggregate well. By  $a$  and  $b$  we jointly denote all the  $a_i$  and  $b_{tj}$  variables. The two parts (sums) of the objective function are normalized to  $[0, 1]$  by dividing by the total number of available facts  $|F|$  and the number of subsets  $m$  times the total number of distinct elements  $|B|$ . In the first part of the objective, we treat all the facts as equally important; if importance scores are also available for the facts, they can be added as multipliers of  $\alpha_i$ . The parameters  $\lambda_1$  and  $\lambda_2$  are used to tune the priority given to expressing many facts vs. generating shorter texts; we set  $\lambda_1 + \lambda_2 = 1$ .

$$\max_{a,b} \lambda_1 \cdot \frac{|F|}{\sum_{i=1}^{|F|} a_i} - \lambda_2 \cdot \sum_{j=1}^m \sum_{t=1}^{|B|} \frac{b_{tj}}{m \cdot |B|} \quad (4)$$

subject to:

$$a_i = \sum_{j=1}^m \sum_{k=1}^{|P_i|} l_{ikj}, \text{ for } i = 1, \dots, n \quad (5)$$

$$\sum_{e_t \in B_{ik}} b_{tj} \geq |B_{ik}| \cdot l_{ikj}, \text{ for } \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, m \\ k = 1, \dots, |P_i| \end{matrix} \quad (6)$$

$$\sum_{p_{ik} \in P(e_t)} l_{ikj} \geq b_{tj}, \text{ for } \begin{matrix} t = 1, \dots, |B| \\ j = 1, \dots, m \end{matrix} \quad (7)$$

$$\sum_{t=1}^{|B|} b_{tj} \leq B_{max}, \text{ for } j = 1, \dots, m \quad (8)$$

$$\sum_{k=1}^{|P_i|} l_{ikj} + \sum_{k'=1}^{|P_{i'}|} l_{i'k'j} \leq 1, \text{ for } \begin{matrix} j = 1, \dots, m, i = 2, \dots, n \\ i' = 1, \dots, n-1; i \neq i' \\ section(f_i) \neq section(f_{i'}) \end{matrix} \quad (9)$$

Constraint 5 ensures that for each selected fact, only one sentence plan in only one subset is selected; if a fact is not selected, no sentence plan for the fact is selected either.  $|\sigma|$  denotes the cardinality of a set  $\sigma$ . In constraint 6,  $B_{ik}$  is the set of distinct elements  $e_t$  of the sentence plan  $p_{ik}$ . This constraint ensures that if  $p_{ik}$  is selected in a subset  $s_j$ , then all the elements of  $p_{ik}$  are also present in  $s_j$ . If  $p_{ik}$  is not selected in  $s_j$ , then some of its elements may still be present in  $s_j$ , if they appear in another selected sentence plan of  $s_j$ .

In constraint 7,  $P(e_t)$  is the set of sentence plans that contain element  $e_t$ . If  $e_t$  is used in a subset  $s_j$ ,

then at least one of the sentence plans of  $P(e_t)$  must also be selected in  $s_j$ . If  $e_t$  is not used in  $s_j$ , then no sentence plan of  $P(e_t)$  may be selected in  $s_j$ . Lastly, constraint 8 limits the number of elements that a subset  $s_j$  can contain to a maximum allowed number  $B_{max}$ , in effect limiting the maximum length of an aggregated sentence.

We assume that each relation  $R$  has been manually mapped to a single *topical section*; e.g., relations expressing the color, body, and flavor of a wine may be grouped in one section, and relations about the wine's producer in another. The section of a fact  $f_i = \langle S_i, R_i, O_i \rangle$  is the section of its relation  $R_i$ . Constraint 9 ensures that facts from different sections will not be placed in the same subset  $s_j$ , to avoid unnatural aggregations.

## 4 Experiments

We used NaturalOWL (Galanis and Androutsopoulos, 2007; Galanis et al., 2009; Androutsopoulos et al., 2013), an NLG system for OWL ontologies that relies on a pipeline of content selection, text planning, lexicalization, aggregation, referring expression generation, and surface realization.<sup>3</sup> We modified content selection, lexicalization, and aggregation to use our ILP model, maintaining the aggregation rules of the original system.<sup>4</sup> For referring expression generation and surface realization, the new system, called ILPNLG, invokes the corresponding components of NaturalOWL.

The original system, called PIPELINE, assumes that each relation has been mapped to a topical section, as in ILPNLG. It also assumes that a manually specified order of the sections and the relations of each section is available, which is used by the text planner to order the selected facts (by their relations). The subsequent components of the pipeline are not allowed to change the order of the facts, and aggregation operates only on sentence plans of adjacent facts from the same section. In ILPNLG, the manually specified order of sections and relations is used to order the sentences of each subset  $s_j$  (before aggregating them), the aggregated sentences in each section (each aggregated sentence inherits the minimum order of its constituents), and the sections (with their sentences).

We used the Wine Ontology, which had been

<sup>3</sup>All the software and data we used are freely available from <http://nlp.cs.aueb.gr/software.html>. We use version 2 of NaturalOWL.

<sup>4</sup>We use the Branch and Cut implementation of GLPK; see [sourceforge.net/projects/winglpk/](http://sourceforge.net/projects/winglpk/).

used in previous experiments with PIPELINE.<sup>5</sup> We kept the 2 topical sections, the ordering of sections and relations, and the sentence plans that had been used in the previous experiments, but we added more sentence plans to ensure that 3 sentence plans were available per fact. We generated texts for the 52 wine individuals of the ontology; we did not experiment with texts describing classes of wines, because we could not think of multiple alternative sentence plans for many of their axioms. For each individual, there were 5 facts on average and a maximum of 6 facts.

PIPELINE has a parameter  $M$  specifying the maximum number of facts it is allowed to report per text. When  $M$  is smaller than the number of available facts  $|F|$  and all the facts are treated as equally important, as in our experiments, it selects randomly  $M$  of the available facts. We repeated the generation of PIPELINE’s texts for the 52 individuals for  $M = 2, 3, 4, 5, 6$ . For each  $M$ , the texts of PIPELINE for the 52 individuals were generated three times, each time using one of the different alternative sentence plans of each relation. We also generated the texts using a variant of PIPELINE, dubbed PIPELINESHORT, which always selects the shortest (in elements) sentence plan among the available ones. In all cases, PIPELINE and PIPELINESHORT were allowed to form aggregated sentences containing up to  $B_{max} = 22$  distinct elements, which was the number of distinct elements of the longest aggregated sentence in the previous experiments, where PIPELINE was allowed to aggregate up to 3 original sentences.

With ILPNLG, we repeated the generation of the texts of the 52 individuals using different values of  $\lambda_1$  ( $\lambda_2 = 1 - \lambda_1$ ), which led to texts expressing from zero to all of the available facts. We set the maximum number of fact subsets to  $m = 3$ , which was the maximum number of aggregated sentences observed in the texts of PIPELINE and PIPELINESHORT. Again, we set  $B_{max} = 22$ .

We compared ILPNLG to PIPELINE and PIPELINESHORT by measuring the average number of facts they reported divided by the average text length (in words). Figure 1 shows this ratio as a function of the average number of reported facts, along with 95% confidence intervals (of sample means). PIPELINESHORT achieved better results than PIPELINE, but the differences were small.

For  $\lambda_1 < 0.2$ , ILPNLG produces empty texts,

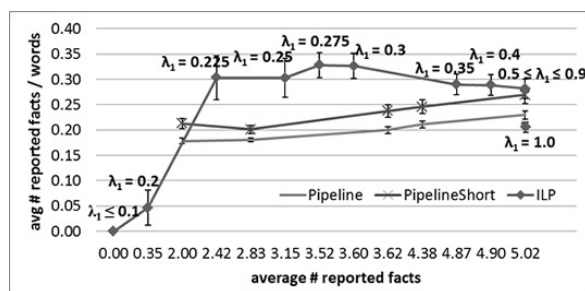


Figure 1: Facts/words ratio of the generated texts.

since it focuses on minimizing the number of distinct elements of each text. For  $\lambda_1 \geq 0.225$ , it performs better than the other systems. For  $\lambda_1 \approx 0.3$ , it obtains the highest fact/words ratio by selecting the facts and sentence plans that lead to the most compressive aggregations. For greater values of  $\lambda_1$ , it selects additional facts whose sentence plans do not aggregate that well, which is why the ratio declines. For small numbers of facts, the two pipeline systems select facts and sentence plans that offer very few aggregation opportunities; as the number of selected facts increases, some more aggregation opportunities arise, which is why the facts/words ratio of the two systems improves. In all the experiments, the ILP solver was very fast (average: 0.08 sec, worst: 0.14 sec). Experiments with human judges also showed that the texts of ILPNLG cannot be distinguished from those of PIPELINESHORT in terms of fluency and text clarity. Hence, the highest compactness of the texts of ILPNLG does *not* come at the expense of lower text quality. Space does not permit a more detailed description of these experiments.

We show below texts produced by PIPELINE ( $M = 4$ ) and ILPNLG ( $\lambda_1 = 0.3$ ).

PIPELINE: This is a strong Sauternes. It is made from Semillon grapes and it is produced by Chateau D’ychem.

ILPNLG: This is a strong Sauternes. It is made from Semillon grapes by Chateau D’ychem.

PIPELINE: This is a full Riesling and it has moderate flavor. It is produced by Volrad.

ILPNLG: This is a full sweet moderate Riesling.

In the first pair, PIPELINE uses different verbs for the grapes and producer, whereas ILPNLG uses the same verb, which leads to a more compressive aggregation; both texts describe the same wine and report 4 facts. In the second pair, ILPNLG has chosen to express the sweetness instead of the producer, and uses the same verb (“be”) for all the facts, leading to a shorter sentence; again both texts describe the same wine and report 4 facts.

<sup>5</sup>See [www.w3.org/TR/owl-guide/wine.rdf](http://www.w3.org/TR/owl-guide/wine.rdf).

In both examples, some facts are not aggregated because they belong in different sections.

## 5 Conclusions

We presented an ILP model for NLG that jointly considers the choices in content selection, lexicalization, and aggregation to avoid greedy local decisions and produce more compact texts. Experiments verified that our model can express more facts per word, compared to a pipeline, which is important when space is scarce. An off-the-shelf ILP solver took approximately 0.1 sec for each text. We plan to extend our model to include text planning and referring expressions generation.

## Acknowledgments

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

## References

- E. Althaus, N. Karamanis, and A. Koller. 2004. Computing locally coherent discourses. In *42nd Annual Meeting of ACL*, pages 399–406, Barcelona, Spain.
- I. Androutsopoulos, G. Lampouras, and D. Galanis. 2013. Generating natural language descriptions from OWL ontologies: the NaturalOWL system. Technical report, Natural Language Processing Group, Department of Informatics, Athens University of Economics and Business.
- G. Antoniou and F. van Harmelen. 2008. *A Semantic Web primer*. MIT Press, 2nd edition.
- R. Barzilay and M. Lapata. 2005. Collective content selection for concept-to-text generation. In *HLT-EMNLP*, pages 331–338, Vancouver, BC, Canada.
- R. Barzilay and M. Lapata. 2006. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*, pages 359–366, New York, NY.
- A. Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.
- T. Berg-Kirkpatrick, D. Gillick, and D. Klein. 2011. Jointly learning to extract and compress. In *49th Annual Meeting of ACL*, pages 481–490, Portland, OR.
- K. Bontcheva. 2005. Generating tailored textual summaries from ontologies. In *2nd European Semantic Web Conf.*, pages 531–545, Heraklion, Greece.
- J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 1(31):399–429.
- H. Dalianis. 1999. Aggregation in natural language generation. *Comput. Intelligence*, 15(4):384–414.
- L. Danlos. 1984. Conceptual and linguistic decisions in generation. In *10th COLING*, pages 501–504, Stanford, CA.
- D. Galanis and I. Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In *11th European Workshop on Natural Lang. Generation*, pages 143–146, Schloss Dagstuhl, Germany.
- D. Galanis, G. Karakatsiotis, G. Lampouras, and I. Androutsopoulos. 2009. An open-source natural language generator for OWL ontologies and its use in Protégé and Second Life. In *12th Conf. of the European Chapter of ACL (demos)*, Athens, Greece.
- D. Galanis, G. Lampouras, and I. Androutsopoulos. 2012. Extractive multi-document summarization with Integer Linear Programming and Support Vector Regression. In *COLING*, pages 911–926, Mumbai, India.
- B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. 2008. OWL 2: The next step for OWL. *Web Semantics*, 6:309–322.
- I. Konstas and M. Lapata. 2012a. Concept-to-text generation via discriminative reranking. In *50th Annual Meeting of ACL*, pages 369–378, Jeju Island, Korea.
- I. Konstas and M. Lapata. 2012b. Unsupervised concept-to-text generation with hypergraphs. In *HLT-NAACL*, pages 752–761, Montréal, Canada.
- P. Liang, M. Jordan, and D. Klein. 2009. Learning semantic correspondences with less supervision. In *47th Meeting of ACL and 4th AFNLP*, pages 91–99, Suntec, Singapore.
- S.F. Liang, R. Stevens, D. Scott, and A. Rector. 2011. Automatic verbalisation of SNOMED classes using OntoVerbal. In *13th Conf. AI in Medicine*, pages 338–342, Bled, Slovenia.
- T. Marciniak and M. Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *9th Conference on Computational Natural Language Learning*, pages 136–143, Ann Arbor, MI.
- R. McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*, pages 557–564, Rome, Italy.

- C. Mellish and J.Z. Pan. 2008. Natural language directed inference from ontologies. *Artificial Intelligence*, 172:1285–1315.
- C. Mellish and X. Sun. 2006. The Semantic Web as a linguistic resource: opportunities for nat. lang. generation. *Knowledge Based Systems*, 19:298–303.
- E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge Univ. Press.
- R. Schwitter, K. Kaljurand, A. Cregan, C. Dolbear, and G. Hart. 2008. A comparison of three controlled nat. languages for OWL 1.1. In *4th OWL Experiences and Directions Workshop*, Washington DC.
- R. Schwitter. 2010. Controlled natural languages for knowledge representation. In *23rd COLING*, pages 1113–1121, Beijing, China.
- S. Williams, A. Third, and R. Power. 2011. Levels of organization in ontology verbalization. In *13th European Workshop on Natural Lang. Generation*, pages 158–163, Nancy, France.
- K. Woodsend and M. Lapata. 2012. Multiple aspect summarization using integer linear programming. In *EMNLP-CoNLL*, pages 233–243, Jesu Island, Korea.

# Sequential Summarization: A New Application for Timely Updated Twitter Trending Topics

Dehong Gao, Wenjie Li, Renxian Zhang

Department of Computing, the Hong Kong Polytechnic University, Hong Kong  
{csdgao, cswjli, csrzhang}@comp.polyu.edu.hk

## Abstract

The growth of the Web 2.0 technologies has led to an explosion of social networking media sites. Among them, Twitter is the most popular service by far due to its ease for real-time sharing of information. It collects millions of tweets per day and monitors what people are talking about in the trending topics updated timely. Then the question is how users can understand a topic in a short time when they are frustrated with the overwhelming and unorganized tweets. In this paper, this problem is approached by sequential summarization which aims to produce a sequential summary, i.e., a series of chronologically ordered short sub-summaries that collectively provide a full story about topic development. Both the number and the content of sub-summaries are automatically identified by the proposed stream-based and semantic-based approaches. These approaches are evaluated in terms of sequence coverage, sequence novelty and sequence correlation and the effectiveness of their combination is demonstrated.

## 1 Introduction and Background

Twitter, as a popular micro-blogging service, collects millions of real-time short text messages (known as tweets) every second. It acts as not only a public platform for posting trifles about users' daily lives, but also a public reporter for real-time news. Twitter has shown its powerful ability in information delivery in many events, like the wildfires in San Diego and the earthquake in Japan. Nevertheless, the side effect is individual users usually sink deep under millions of flooding-in tweets. To alleviate this problem, the applications like *whatthetrend*<sup>1</sup> have evolved from Twitter to provide services that encourage users to edit explanatory tweets about a trending topic, which can be regarded as topic summaries. It is to some extent a good way to help users understand trending topics.

There is also pioneering research in automatic Twitter trending topic summarization. (O'Connor et al., 2010) explained Twitter trending topics by providing a list of significant terms. Users could utilize these terms to drill down to the tweets which are related to the trending topics. (Sharifi et al., 2010) attempted to provide a one-line summary for each trending topic using phrase reinforcement ranking. The relevance model employed by (Harabagiu and Hickl, 2011) generated summaries in larger size, i.e., 250-word summaries, by synthesizing multiple high rank tweets. (Duan et al., 2012) incorporate the user influence and content quality information in timeline tweet summarization and employ reinforcement graph to generate summaries for trending topics.

Twitter summarization is an emerging research area. Current approaches still followed the traditional summarization route and mainly focused on mining tweets of both significance and representativeness. Though, the summaries generated in such a way can sketch the most important aspects of the topic, they are incapable of providing full descriptions of the changes of the focus of a topic, and the temporal information or freshness of the tweets, especially for those newsworthy trending topics, like earthquake and sports meeting. As the main information producer in Twitter, the massive crowd keeps close pace with the development of trending topics and provide the timely updated information. The information dynamics and timeliness is an important consideration for Twitter summarization. That is why we propose sequential summarization in this work, which aims to produce sequential summaries to capture the temporal changes of mass focus.

Our work resembles update summarization promoted by TAC<sup>2</sup> which required creating summaries with new information assuming the reader has already read some previous documents under the same topic. Given two chronologically ordered documents sets about a topic, the systems were asked to generate two

---

<sup>1</sup> [whatthetrend.com](http://whatthetrend.com)

---

<sup>2</sup> [www.nist.gov/tac](http://www.nist.gov/tac)

summaries, and the second one should inform the user of new information only. In order to achieve this goal, existing approaches mainly emphasized the novelty of the subsequent summary (Li and Croft, 2006; Varma et al., 2009; Steinberger and Jezek, 2009). Different from update summarization, we focus more on the temporal change of trending topics. In particular, we need to automatically detect the “update points” among a myriad of related tweets.

It is the goal of this paper to set up a new practical summarization application tailored for timely updated Twitter messages. With the aim of providing a full description of the focus changes and the records of the timeline of a trending topic, the systems are expected to discover the chronologically ordered sets of information by themselves and they are free to generate any number of update summaries according to the actual situations instead of a fixed number of summaries as specified in DUC/TAC. Our main contributions include novel approaches to sequential summarization and corresponding evaluation criteria for this new application. All of them will be detailed in the following sections.

## 2 Sequential Summarization

Sequential summarization proposed here aims to generate a series of chronologically ordered sub-summaries for a given Twitter trending topic. Each sub-summary is supposed to represent one main subtopic or one main aspect of the topic, while a sequential summary, made up by the sub-summaries, should retain the order the information is delivered to the public. In such a way, the sequential summary is able to provide a general picture of the entire topic development.

### 2.1 Subtopic Segmentation

One of the keys to sequential summarization is subtopic segmentation. How many subtopics have attracted the public attention, what are they, and how are they developed? It is important to provide the valuable and organized materials for more fine-grained summarization approaches. We proposed the following two approaches to automatically detect and chronologically order the subtopics.

#### 2.1.1 Stream-based Subtopic Detection and Ordering

Typically when a subtopic is popular enough, it will create a certain level of surge in the tweet stream. In other words, every surge in the tweet

stream can be regarded as an indicator of the appearance of a subtopic that is worthy of being summarized. Our early investigation provides evidence to support this assumption. By examining the correlations between tweet content changes and volume changes in randomly selected topics, we have observed that the changes in tweet volume can really provide the clues of topic development or changes of crowd focus.

The stream-based subtopic detection approach employs the offline peak area detection (Opad) algorithm (Shamma et al., 2010) to locate such surges by tracing tweet volume changes. It regards the collection of tweets at each such surge time range as a new subtopic.

#### Offline Peak Area Detection (Opad) Algorithm

---

```

1: Input:  $TS$  (tweets stream, each  $tw_i$  with timestamp  $t_i$ );
   peak interval window  $\Delta t$  (in hour), and time
   step  $h$  ( $h \ll \Delta t$ );
2: Output: Peak Areas  $PA$ .
3: Initial: two time slots:  $T' = T = t_0 + \Delta t$ ;
   Tweet numbers:  $N' = N = \mathbf{Count}(T)$ 
4: while  $(t_s = T + h) < t_{n-1}$ 
5:   update  $T' = t_s + \Delta t$  and  $N' = \mathbf{Count}(T')$ 
6:   if  $(N' < N$  And up-hilling)
7:     output one peak area  $pa^T$ 
8:     state of down-hilling
9:   else
10:    update  $T = T'$  and  $N = N'$ 
11:    state of up-hilling
12:
13: function  $\mathbf{Count}(T)$ 
14:   Count tweets in time interval  $T$ 

```

---

The subtopics detected by the Opad algorithm are naturally ordered in the timeline.

#### 2.1.2 Semantic-based Subtopic Detection and Ordering

Basically the stream-based approach monitors the changes of the level of user attention. It is easy to implement and intuitively works, but it fails to handle the cases where the posts about the same subtopic are received at different time ranges due to the difference of geographical and time zones. This may make some subtopics scattered into several time slots (peak areas) or one peak area mixed with more than one subtopic.

In order to sequentially segment the subtopics from the semantic aspect, the semantic-based subtopic detection approach breaks the time order of tweet stream, and regards each tweet as an individual short document. It takes advantage of Dynamic Topic Modeling (David and Michael, 2006) to explore the tweet content.



DTM in nature is a clustering approach which can dynamically generate the subtopic underlying the topic. Any clustering approach requires a pre-specified cluster number. To avoid tuning the cluster number experimentally, the subtopic number required by the semantic-based approach is either calculated according to heuristics or determined by the number of the peak areas detected from the stream-based approach in this work.

Unlike the stream-based approach, the subtopics formed by DTM are the sets of distributions of subtopic and word probabilities. They are time independent. Thus, the temporal order among these subtopics is not obvious and needs to be discovered. We use the probabilistic relationships between tweets and topics learned from DTM to assign each tweet to a subtopic that it most likely belongs to. Then the subtopics are ordered temporally according to the mean values of their tweets' timestamps.

## 2.2 Sequential Summary Generation

Once the subtopics are detected and ordered, the tweets belonging to each subtopic are ranked and the most significant one is extracted to generate the sub-summary regarding that subtopic. Two different ranking strategies are adopted to conform to two different subtopic detection mechanisms.

For a tweet in a peak area, the linear combination of two measures is considered to evaluate its significance to be a sub-summary: (1) *subtopic representativeness* measured by the cosine similarity between the tweet and the centroid of all the tweets in the same peak area; (2) *crowding endorsement* measured by the times that the tweet is re-tweeted normalized by the total number of re-tweeting. With the DTM model, the significance of the tweets is evaluated directly by word distribution per subtopic.

MMR (Carbonell and Goldstein, 1998) is used to reduce redundancy in sub-summary generation.

## 3 Experiments and Evaluations

The experiments are conducted on the 24 Twitter trending topics collected using Twitter APIs<sup>3</sup>. The statistics are shown in Table 1.

Due to the shortage of gold-standard sequential summaries, we invite two annotators to read the chronologically ordered tweets, and write a series of sub-summaries for each topic

independently. Each sub-summary is up to 140 characters in length to comply with the limit of tweet, but the annotators are free to choose the number of sub-summaries. It ends up with 6.3 and 4.8 sub-summaries on average in a sequential summary written by the two annotators respectively. These two sets of sequential summaries are regarded as reference summaries to evaluate system-generated summaries from the following three aspects.

| Category      | #TT | Trending Topic Examples             | Tweets Number |
|---------------|-----|-------------------------------------|---------------|
| News          | 6   | <i>Minsk, Libya Release</i>         | 25145         |
| Sports        | 6   | <i>#bbcf1, Lakers/Heat</i>          | 17204         |
| Technology    | 5   | <i>Google Fiber</i>                 | 13281         |
| Science       | 2   | <i>AHINI, Richter</i>               | 10935         |
| Entertainment | 2   | <i>Midnight Club, #ilovemyfans,</i> | 6573          |
| Meme          | 2   | <i>Night Angels</i>                 | 14595         |
| Lifestyle     | 1   | <i>Goose Island</i>                 | 6230          |
| Total         | 24  | -----                               | 93963         |

Table 1. Data Set

- **Sequence Coverage**

Sequence coverage measures the  $N$ -gram match between system-generated summaries and human-written summaries (stopword removed first). Considering temporal information is an important factor in sequential summaries, we propose the *position-aware* coverage measure by accommodating the position information in matching. Let  $S = \{s_1, s_2, \dots, s_k\}$  denote a sequential summary and  $s_i$  the  $i$ th sub-summary,  $N$ -gram coverage is defined as:

$$\text{Coverage} = \frac{1}{|S_{sg}|} \sum_{s_i \in S_{sg}} \frac{\sum_{s_j \in S_{hw}} \sum_{N\text{-gram} \in s_i, s_j} \text{Count}_{\text{Match}}(N\text{-gram})}{\omega_{ij} \cdot \sum_{s_j \in S_{hw}} \sum_{N\text{-gram} \in s_j} \text{Count}(N\text{-gram})}$$

where,  $\omega_{ij} = |j - i| + 1$ ,  $i$  and  $j$  denote the serial numbers of the sub-summaries in the system-generated summary  $S_{sg}$  and the human-written summary  $S_{hw}$ , respectively.  $\omega$  serves as a coefficient to discount long-distance matched sub-summaries. We evaluate unigram, bigram, and skipped bigram matches. Like in ROUGE (Lin, 2004), the skip distance is up to four words.

- **Sequence Novelty**

Sequence novelty evaluates the average novelty of two successive sub-summaries. Information content (IC) has been used to measure the novelty of update summaries by (Aggarwal et al., 2009). In this paper, the novelty of a system-

<sup>3</sup><https://dev.twitter.com/>

generated sequential summary is defined as the average of IC increments of two adjacent sub-summaries,

$$Novelty = \frac{1}{|S|-1} \sum_{i>1} (IC_{s_i} - IC_{s_i, s_{i-1}})$$

where  $|S|$  is the number of sub-summaries in the sequential summary.  $IC_{s_i} = \sum_{w \in s_i} IC_w$ .  $IC_{s_i, s_{i-1}} = \sum_{w \in s_i \cap s_{i-1}} IC_w$  is the overlapped information in the two adjacent sub-summaries.  $IC_w = ITF_w \times Relevance(w, W_{Tw})$  where  $w$  is a word,  $ITF_w$  is the inverse tweet frequency of  $w$ , and  $W_{Tw}$  is all the tweets in the trending topic. The relevance function is introduced to ensure that the information brought by new sub-summaries is not only novel but also related to the topic.

#### • Sequence Correlation

Sequence correlation evaluates the sequential matching degree between system-generated and human-written summaries. In statistics, Kendall's *tau* coefficient is often used to measure the association between two sequences (Lapata, 2006). The basic idea is to count the concordant and discordant pairs which contain the same elements in two sequences. Borrowing this idea, for each sub-summary in a human-generated summary, we find its most matched sub-summary (judged by the cosine similarity measure) in the corresponding system-generated summary and then define the correlation according to the concordance between the two matched sub-summary sequences.

$$\begin{aligned} & \text{Correlation} \\ &= \frac{2(|\#ConcordantPairs| - |\#DiscordantPairs|)}{n(n-1)} \end{aligned}$$

where  $n$  is the number of human-written sub-summaries.

Tables 2 and 3 below present the evaluation results. For the stream-based approach, we set  $\Delta t=3$  hours experimentally. For the semantic-based approach, we compare three different approaches to defining the sub-topic number  $K$ : (1) Semantic-based 1: Following the approach proposed in (Li et al., 2007), we first derive the matrix of tweet cosine similarity. Given the 1-norm of eigenvalues  $\lambda_i^{norm}$  ( $i = 1, 2, \dots, n$ ) of the similarity matrix and the ratios  $\gamma_i = \lambda_i^{norm}/\lambda_2$ , the subtopic number  $K = i + 1$  if  $\gamma_i - \gamma_{i+1} > \delta$  ( $\delta = 0.4$ ). (2) Semantic-based 2: Using the rule of thumb in (Wan and Yang, 2008),  $K = \sqrt{n}$ , where  $n$  is the tweet number. (3) Combined:  $K$  is defined as the number of the peak areas detected from the Opad algorithm, meanwhile we use the tweets within peak areas as the tweets of DTM. This is our new idea.

The experiments confirm the superiority of the semantic-based approach over the stream-based approach in summary content coverage and novelty evaluations, showing that the former is better at subtopic content modeling. The sub-summaries generated by the stream-based approach have comparative sequence (i.e., order) correlation with the human summaries. Combining the advantages the two approaches leads to the best overall results.

| Coverage                        | Unigram       | Bigram        | Skipped Bigram |
|---------------------------------|---------------|---------------|----------------|
| Stream-based( $\Delta t=3$ )    | 0.3022        | 0.1567        | 0.1523         |
| Semantic-based1( $\delta=0.5$ ) | 0.3507        | 0.1684        | 0.1866         |
| Semantic-based 2                | 0.3112        | 0.1348        | 0.1267         |
| Combined( $\Delta t=3$ )        | <b>0.3532</b> | <b>0.1699</b> | <b>0.1791</b>  |

Table 2. N-Gram Coverage Evaluation

| Approaches                        | Novelty       | Correlation   |
|-----------------------------------|---------------|---------------|
| Stream-based ( $\Delta t=3$ )     | 0.3798        | 0.3330        |
| Semantic-based 1 ( $\delta=0.4$ ) | 0.7163        | 0.3746        |
| Semantic-based 2                  | 0.7017        | 0.3295        |
| Combined ( $\Delta t=3$ )         | <b>0.7793</b> | <b>0.3986</b> |

Table 3. Novelty and Correlation Evaluation

## 4 Concluding Remarks

We start a new application for Twitter trending topics, i.e., sequential summarization, to reveal the developing scenario of the trending topics while retaining the order of information presentation. We develop several solutions to automatically detect, segment and order subtopics temporally, and extract the most significant tweets into the sub-summaries to compose sequential summaries. Empirically, the combination of the stream-based approach and the semantic-based approach leads to sequential summaries with high coverage, low redundancy, and good order.

## Acknowledgments

The work described in this paper is supported by a Hong Kong RGC project (PolyU No. 5202/12E) and a National Nature Science Foundation of China (NSFC No. 61272291).

## References

- Aggarwal Gaurav, Sumbaly Roshan and Sinha Shakti. 2009. Update Summarization. Stanford: CS224N Final Projects.

- Blei M. David and Jordan I. Michael. 2006. Dynamic topic models. In Proceedings of the 23rd international conference on Machine learning, 113-120. Pittsburgh, Pennsylvania.
- Carbonell Jaime and Goldstein Jade. 1998. The use of MMR, diversity based reranking for reordering documents and producing summaries. In Proceedings of the 21<sup>st</sup> Annual International Conference on Research and Development in Information Retrieval, 335-336. Melbourne, Australia.
- Duan Yajuan, Chen Zhimin, Wei Furu, Zhou Ming and Heung-Yeung Shum. 2012. Twitter Topic Summarization by Ranking Tweets using Social Influence and Content Quality. In Proceedings of the 24<sup>th</sup> International Conference on Computational Linguistics, 763-780. Mumbai, India.
- Harabagiu Sanda and Hickl Andrew. 2011. Relevance Modeling for Microblog Summarization. In Proceedings of 5<sup>th</sup> International AAAI Conference on Weblogs and Social Media. Barcelona, Spain.
- Lapata Mirella. 2006. Automatic evaluation of information ordering: Kendall's tau. Computational Linguistics, 32(4):1-14.
- Li Wenyuan, Ng Wee-Keong, Liu Ying and Ong Kok-Leong. 2007. Enhancing the Effectiveness of Clustering with Spectra Analysis. IEEE Transactions on Knowledge and Data Engineering, 19(7):887-902.
- Li Xiaoyan and Croft W. Bruce. 2006. Improving novelty detection for general topics using sentence level information patterns. In Proceedings of the 15<sup>th</sup> ACM International Conference on Information and Knowledge Management, 238-247. New York, USA.
- Lin Chin-Yew. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the ACL Workshop on Text Summarization Branches Out, 74-81. Barcelona, Spain.
- Liu Fei, Liu Yang and Weng Fuliang. 2011. Why is "SXSW" trending? Exploring Multiple Text Sources for Twitter Topic Summarization. In Proceedings of the ACL Workshop on Language in Social Media, 66-75. Portland, Oregon.
- O'Connor Brendan, Krieger Michel and Ahn David. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In Proceedings of the 4<sup>th</sup> International AAAI Conference on Weblogs and Social Media, 384-385. Atlanta, Georgia.
- Shamma A. David, Kennedy Lyndon and Churchill F. Elizabeth. 2010. Tweetgeist: Can the Twitter Timeline Reveal the Structure of Broadcast Events? In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, 589-593. Savannah, Georgia, USA.
- Sharifi Beaux, Hutton Mark-Anthony and Kalita Jugal. 2010. Summarizing Microblogs Automatically. In Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 685-688. Los Angeles, California.
- Steinberger Josef and Jezek Karel. 2009. Update summarization based on novel topic distribution. In Proceedings of the 9<sup>th</sup> ACM Symposium on Document Engineering, 205-213. Munich, Germany.
- Varma Vasudeva, Bharat Vijay, Kovelamudi Sudheer, Praveen Bysani, Kumar K. N, Kranthi Reddy, Karuna Kumar and Nitin Maganti. 2009. IIT Hyderabad at TAC 2009. In Proceedings of the 2009 Text Analysis Conference. Gaithersburg, Maryland.
- Wan Xiaojun and Yang Jianjun. 2008. Multi-document summarization using cluster-based link analysis. In Proceedings of the 31<sup>st</sup> Annual International Conference on Research and Development in Information Retrieval, 299-306. Singapore, Singapore.

# A System for Summarizing Scientific Topics Starting from Keywords

**Rahul Jha**

Department of EECS  
University of Michigan  
Ann Arbor, MI, USA  
rahuljha@umich.edu

**Amjad Abu-Jbara**

Department of EECS  
University of Michigan  
Ann Arbor, MI, USA  
amjbara@umich.edu

**Dragomir Radev**

Department of EECS and  
School of Information  
University of Michigan  
Ann Arbor, MI, USA  
radev@umich.edu

## Abstract

In this paper, we investigate the problem of automatic generation of scientific surveys starting from keywords provided by a user. We present a system that can take a topic query as input and generate a survey of the topic by first selecting a set of relevant documents, and then selecting relevant sentences from those documents. We discuss the issues of robust evaluation of such systems and describe an evaluation corpus we generated by manually extracting factoids, or information units, from 47 gold standard documents (surveys and tutorials) on seven topics in Natural Language Processing. We have manually annotated 2,625 sentences with these factoids (around 375 sentences per topic) to build an evaluation corpus for this task. We present evaluation results for the performance of our system using this annotated data.

## 1 Introduction

The rise of the number of publications in all scientific fields is making it more and more difficult to get quickly acquainted with the new developments in a new area. One way to wade through this huge amount of scholarly information is to consult topical surveys written by experts in an area. For example, for machine translation, one might read (Lopez, 2008)<sup>1</sup>. Such surveys can be very helpful when available, but unfortunately, may not be available for all areas. Additionally, the manual surveys quickly go out of date within a few years of publication as additional papers are published in the field.

<sup>1</sup>Adam Lopez. 2008. Statistical machine translation. *ACM Comput. Surv.* 40, 3, Article 8

Thus, a system that can generate such surveys automatically would be a useful tool. Short summaries in the form of abstracts are available for individual papers, but no such information is available for scientific topics. In this paper, we explore strategies for generating and evaluating such surveys of scientific topics automatically starting from a phrase representing a topic area. We evaluate our system on a set of topics in the field of Natural Language Processing. In earlier work, (Teufel and Moens, 2002) have examined the problem of summarizing scientific articles using rhetorical analysis of sentences. Nanba and Okumura (1999) have also discussed the problem of generating surveys of multiple papers. Mohammad et al. (2009) presented experiments on generating surveys of scientific topics starting from papers to be summarized. More recently, Hoang and Kan (2010) have presented initial results on automatically generating related work section for a target paper by taking a hierarchical topic tree as an input.

In this paper, we tackle the more challenging problem of summarizing a topic starting from a topic query. Our system takes as an input a string describing the topic area, selects the relevant papers from a corpus of papers, and then selects sentences from the citing sentences to these papers to generate a survey of the topic. A sample output of our system for the topic of “Word Sense Disambiguation” is shown in Figure 1.

## 2 Candidate Document Selection

Given a query representing the topic to be summarized, our first task is to find the set of relevant documents from the corpus. The simplest way to do this for a corpus of scientific publications is to do a query search using exact match or a standard TF\*IDF system such Lucene, rank the documents using either citation counts or pagerank in the bibliometric citation network, and select the top  $n$  documents. However, comparing

Many corpus based methods have been proposed to deal with the sense disambiguation problem when given definition for each possible sense of a target word or a tagged corpus with the instances of each possible sense, e.g., supervised sense disambiguation (Leacock et al., 1998), and semi-supervised sense disambiguation (Yarowsky, 1995).

Most researchers working on word sense disambiguation (WSD) use manually sense tagged data such as SemCor (Miller et al., 1993) to train statistical classifiers, but also use the information in SemCor on the overall sense distribution for each word as a backoff model.

Yarowsky (1995) has proposed a bootstrapping method for word sense disambiguation.

Training of WSD Classifier Much research has been done on the best supervised learning approach for WSD (Florian and Yarowsky, 2002; Lee and Ng, 2002; Mihalcea and Moldovan, 2001; Yarowsky et al., 2001).

For example, the use of parallel corpora for sense tagging can help with word sense disambiguation (Brown et al., 1991; Dagan, 1991; Dagan and Itai, 1994; Ide, 2000; Resnik and Yarowsky, 1999).

Figure 1: A sample output survey of our system on the topic of “Word Sense Disambiguation” produced by paper selection using Restricted Expansion and sentence selection using Lexrank. In our evaluations, this survey achieved a pyramid score of 0.82 and Unnormalized RU score of 0.31.

| Document selection algorithm                  | $CG_5$      | $CG_{10}$   | $CG_{20}$   |
|---|-------------|-------------|-------------|
| Title match sorted with citation count        | 1.82        | 2.75        | 3.29        |
| Title match sorted with pagerank              | 1.77        | 2.55        | 3.34        |
| Citation expansion sorted with citation count | 0.53        | 1.20        | 2.29        |
| Citation expansion sorted with pagerank       | 0.20        | 0.78        | 1.99        |
| TF*IDF ranked                                 | 0.14        | 0.14        | 0.56        |
| TF*IDF sorted with citation count             | 0.44        | 2.25        | 3.18        |
| TF*IDF sorted with pagerank                   | 1.54        | 2.22        | 2.85        |
| Restricted Expansion                          | <b>2.52</b> | <b>3.91</b> | <b>6.01</b> |

Table 1: Comparison of different methods for document selection by measuring the Cumulative Gain (CG) of top 5, 10 and 20 results.

the results of these techniques with the papers covered by gold standard surveys on a few topics, we found that some important papers are missed by these simple approaches. One reason for this is that early papers in a field might use non-standard terms in the absence of a stable, accepted terminology. Some early Word Sense Disambiguation papers, for example, refer to the problem as Lexical Ambiguity Resolution. Additionally, papers might use alternative forms or abbreviations of topics in their titles and abstracts, e.g. for input query “Semantic Role Labelling”, papers such as (Dahlmeier et al., 2009) titled “Joint Learning of Preposition Senses and Semantic Roles of Prepositional Phrases” and (Che and Liu, 2010) titled “Jointly Modeling WSD and SRL with Markov Logic” might be missed.

To find these papers, we add a simple heuristic called *Restricted Expansion*. In this method, we first create a base set  $B$ , by finding papers with an exact match to the query. This is a high precision set since a paper with a title that contains the exact query phrase is very likely to be relevant to the topic. We then find additional papers by expanding in the citation network around  $B$ , that is, by finding all the papers that are cited by or cite the papers in  $B$ , to create an extended set  $E$ . From this combined set  $(B \cup E)$ , we create a new set  $F$

by filtering out the set of papers that are not cited by or cite a minimum threshold  $t_{init}$  of papers in  $B$ . If the total number of papers is lower than  $f_{min}$  or higher than  $f_{max}$ , we iteratively increase or decrease  $t$  till  $f_{min} \leq |F| \leq f_{max}$ . This method allows us to increase our recall without losing precision. The values for our current experiments are:  $t_{init} = 5$ ,  $f_{min} = 150$ ,  $f_{max} = 250$ .

| Authors                      | Year | Size |
|------------------------------|------|------|
| <b>Surveys</b>               |      |      |
| ACL Wiki                     | 2012 | 4    |
| Roberto Navigli              | 2009 | 68   |
| Eneko Agirre; Philip Edmonds | 2006 | 28   |
| Xiaohua Zhou; Hyouil Han     | 2005 | 6    |
| Nancy Ide; Jean Vronis       | 1998 | 41   |
| <b>Tutorials</b>             |      |      |
| Sanda Harabagiu              | 2011 | 45   |
| Diana McCarthy               | 2011 | 120  |
| Philipp Koehn                | 2008 | 17   |
| Rada Mihalcea                | 2005 | 186  |

Table 2: The set of surveys and tutorials collected for the topic of “Word Sense Disambiguation”. Sizes for surveys are expressed in number of pages, sizes for tutorials are expressed in number of slides.

To evaluate different methods of candidate document selection, we use Cumulative Gain (CG), where the weight for each paper is estimated by the fraction of surveys it appears in. Table 1 shows the average Cumulative Gain of top 5, 10 and 20 documents for each of eight methods we tried. Restricted Expansion outperformed every other method. Once we obtain a set of papers to be summarized, we select the top  $n$  most cited papers in the document set as the papers to be summarized, and extract the set of citing sentences  $S$  from all the papers in the document set to these  $n$  papers.  $S$  is the input for our sentence selection algorithms, described in Section 4.

| Factoid   | S1 | S2 | S3 | S4 | S5 | T1 | T2 | T3 | T4 | Factoid Weight |
|---|----|----|----|----|----|----|----|----|----|----------------|
| definition of wsd                                       | X  | X  | X  | X  | X  | X  | X  | X  | X  | 9              |
| wordnet   | X  | X  | X  |    | X  | X  | X  | X  | X  | 8              |
| knowledge based wsd                                     |    | X  | X  | X  | X  | X  |    | X  | X  | 7              |
| supervised wsd  | X  | X  | X  | X  | X  | X  |    | X  |    | 7              |
| senseval  | X  | X  | X  |    |    | X  | X  | X  | X  | 7              |
| definition of word senses                               | X  |    | X  | X  |    | X  |    | X  | X  | 7              |
| knowledge based wsd using machine readable dictionaries |    | X  | X  |    | X  | X  |    | X  | X  | 6              |
| unsupervised wsd  |    | X  | X  | X  |    | X  | X  | X  |    | 6              |
| bootstrapping algorithms                                | X  | X  | X  |    |    | X  | X  | X  |    | 6              |
| supervised wsd using decision lists                     | X  | X  | X  | X  | X  |    |    | X  |    | 6              |

Table 3: Top 10 factoids for the topic of “Word Sense Disambiguation” and their distribution across various data sources.

### 3 Evaluation Data for Survey Generation

We use the ACL Anthology Network (AAN) as the corpus for our experiments (Radev et al., 2013). We built a factoid inventory for seven topics in NLP based on manual written surveys in the following way. For each topic, we found at least 3 recent tutorials and 3 recent surveys on the topic and extracted the factoids that are covered in each of them. Table 2 shows the complete list of material collected for the topic of “Word Sense Disambiguation”. We found around 80 factoids per topic on an average. Once the factoids were extracted, each factoid was assigned a weight based on the number of documents it appears in, and any factoids with weight one were removed. Table 3 shows the top ten factoids in the topic of Word Sense Disambiguation along with their distribution across the different surveys and tutorials and final weight.

For each of the topics, we used the method described in Section 2 to create a candidate document set and extracted the candidate citing sentences to be used as the input for the content selection component. Each sentence in each topic was then annotated by a human judge against the factoid list for that topic. A sentence is allowed to have zero or more than one factoid. The human assessors were graduate students in Computer Science who have taken a basic “Natural Language Processing” course or an equivalent course. On an average, 375 citing sentences were annotated for each topic, with 2,625 sentences being annotated in total. We present all our experimental results on this large annotated corpora which is also available for download <sup>2</sup>.

### 4 Content Models

Once we have the set of input sentences, our system must select the sentences that should be part

<sup>2</sup>[http://clair.si.umich.edu/corpora/survey\\_data/](http://clair.si.umich.edu/corpora/survey_data/)

of the survey. For this task, we experimented with three content models, described below.

#### 4.1 Centroid

The centroid of a set of documents is a set of words that are statistically important to the cluster of documents. Centroid based summarization of a document set involves first creating the centroid of the documents, and then judging the salience of each document based on its similarity to the centroid of the document set. In our case, the input citing sentences represent the documents from which we extract the centroid. We use the centroid implementation from the publicly available summarization toolkit, MEAD (Radev et al., 2004).

#### 4.2 Lexrank

LexRank (Erkan and Radev, 2004) is a network based content selection algorithm that works by first building a graph of all the documents in a cluster. The edges between corresponding nodes represent the cosine similarity between them. Once the network is built, the algorithm computes the salience of sentences in this graph based on their eigenvector centrality in the network.

#### 4.3 C-Lexrank

C-Lexrank is another network based content selection algorithm that focuses on diversity (Qazvinian and Radev, 2008). Given a set of sentences, it first creates a network using these sentences and then runs a clustering algorithm to partition the network into smaller clusters that represent different aspects of the paper. The motivation behind the clustering is to include more diverse facts in the summary.

### 5 Experiments and Results

To do an evaluation of our different content selection methods, we first select the documents using our Restricted Expansion method, and then pick

| Topic                     | Rand | Cent | LR    | C-LR |
|---------------------------|------|------|-------|------|
| Summarization             | 0.68 | 0.61 | 0.91  | 0.82 |
| Question Answering        | 0.52 | 0.50 | 0.65  | 0.56 |
| Word Sense Disambiguation | 0.78 | 0.73 | 0.82  | 0.76 |
| Named Entity Recognition  | 0.90 | 0.90 | 0.94  | 0.94 |
| Sentiment Analysis        | 0.75 | 0.78 | 0.77  | 0.78 |
| Semantic Role Labeling    | 0.78 | 0.79 | 0.88  | 0.94 |
| Dependency Parsing        | 0.67 | 0.38 | 0.71  | 0.53 |
| Average                   | 0.72 | 0.68 | 0.81* | 0.76 |

Table 4: Results of pyramid evaluation for each of the three methods and the random baseline on each topic.

the citing sentences to be used as the input to the summarization module as described in Section 2. Given this input, we generate 500 word summaries for each of the seven topics using the four methods: Centroid, Lexrank, C-Lexrank and a random baseline.

For each summary, we compute two evaluation metrics. The first is the Pyramid score (Nenkova and Passonneau, 2004) computed by treating the factoids as Summary Content Units (SCU’s). The Pyramid scores for each summary is shown in Table 4. The second metric is an Unnormalized Relative Utility score (Radev and Tam, 2003), computed using the factoid scores of sentences based on the method presented in (Qazvinian, 2012). We call this Unnormalized RU since we are not able to normalize the scores with human generated gold summaries. The results for Unnormalized RU are shown in Table 5. The parameter  $\alpha$  is the RU penalty for including a redundant sentence subsumed by an earlier sentence. If the summary chooses a sentence  $s_i$  with score  $w_{orig}$  that is subsumed by an earlier summary sentence, the score is reduced as  $w_{subsumed} = (\alpha * w_{orig})$ . We approximate subsumption by marking a sentence  $s_j$  as being subsumed by  $s_i$  if  $F_j \subset F_i$ , where  $F_i$  and  $F_j$  are sets of factoids covered in each sentence.

| Topic                     | Rand | Cent | LR    | C-LR |
|---------------------------|------|------|-------|------|
| Summarization             | 0.16 | 0.57 | 0.29  | 0.17 |
| Question Answering        | 0.32 | 0.39 | 0.48  | 0.30 |
| Word Sense Disambiguation | 0.28 | 0.33 | 0.31  | 0.30 |
| Named Entity Recognition  | 0.36 | 0.38 | 0.34  | 0.31 |
| Sentiment Analysis        | 0.23 | 0.34 | 0.48  | 0.33 |
| Semantic Role Labeling    | 0.11 | 0.17 | 0.16  | 0.21 |
| Dependency Parsing        | 0.16 | 0.05 | 0.30  | 0.15 |
| Average                   | 0.23 | 0.32 | 0.34* | 0.25 |

Table 5: Results of Unnormalized Relative Utility evaluation for the three methods and random baseline using  $\alpha = 0.5$ .

The reason for the relatively high scores for the random baseline is that our process to select the initial set of sentences eliminates many bad sen-

tences. For example, for a subset of 5 topics, the total input set contains 1508 sentences, out of which 922 of the sentences (60%) have at least one factoid. This makes it highly likely to pick good content sentences even when we are picking sentences at random.

We find that the Lexrank method outperforms other sentence selection methods on both evaluation metrics. The higher performance of Lexrank compared to Centroid is consistent with earlier published results (Erkan and Radev, 2004). The reason for the low performance of C-Lexrank as compared to Lexrank on this data set can be attributed to the fact that the input sentence set is derived from a much more diverse set of papers which can have a high diversity in lexical choice when describing the same factoid. Thus simple lexical similarity is not enough to find good clusters in this sentence set.

The lower Unnormalized RU scores compared to Pyramid scores indicate that we are selecting sentences containing highly weighted factoids, but we do not select the most informative sentences that contain a large number of factoids. This also shows that we select some redundant factoids, since Unnormalized RU contains a penalty for redundancy. This is again, explained by the fact that the simple lexical diversity based model in C-Lexrank is not able to detect the same factoids being present in two sentences. Despite these shortcomings, our system works quite well in terms of content selection for unseen topics, Figure 2 shows the top 5 sentences for the query “Conditional Random Fields”.

## 6 Conclusion and Future Work

In this paper, we described a pipeline for the generation of scientific surveys starting from a topic query. Our system is divided into two components. The first component finds the set of papers from the corpus relevant to the query using a simple heuristic called *Restricted Expansion*. The second component selects sentences from these papers to generate a survey of the topic. One of the main contributions of this work is a manually annotated data set for evaluating both the tasks. We collected 47 gold standard documents (surveys and tutorials) on seven topics in Natural Language Processing and extracted factoids for each topic. Each factoid is given an importance score based on the number of gold standard documents it appears in.

---

In recent years, conditional random fields (CRFs) (Lafferty et al., 2001) have shown success on a number of natural language processing (NLP) tasks, including shallow parsing (Sha and Pereira, 2003), named entity recognition (McCallum and Li, 2003) and information extraction from research papers (Peng and McCallum, 2004).

---

In natural language processing, two aspects of CRFs have been investigated sufficiently: one is to apply it to new tasks, such as named entity recognition (McCallum and Li, 2003; Li and McCallum, 2003; Settles, 2004), part-of-speech tagging (Lafferty et al., 2001), shallow parsing (Sha and Pereira, 2003), and language modeling (Roark et al., 2004); the other is to exploit new training methods for CRFs, such as improved iterative scaling (Lafferty et al., 2001), L-BFGS (McCallum, 2003) and gradient tree boosting (Dietterich et al., 2004)

---

NP chunks are very similar to the ones of Ramshaw and Marcus (1995).

---

CRFs have shown empirical successes recently in POS tagging (Lafferty et al., 2001), noun phrase segmentation (Sha and Pereira, 2003) and Chinese word segmentation (McCallum and Feng, 2003)

---

CRFs have been successfully applied to a number of real-world tasks, including NP chunking (Sha and Pereira, 2003), Chinese word segmentation (Peng et al., 2004), information extraction (Pinto et al., 2003; Peng and McCallum, 2004), named entity identification (McCallum and Li, 2003; Settles, 2004), and many others.

---

Figure 2: A sample output survey produced by our system on the topic of “Conditional Random Fields” using Restricted Expansion and Lexrank.

Additionally, we manually annotated 2,625 input sentences, about 375 sentences per topic, with the factoids extracted from the gold standard documents for each topic. Using this corpus, we presented experimental results for the performance of our document selection component and three sentence selection strategies.

Our results indicate three main directions for future work. We plan to look at better models of diversity in sentence selection, since methods based on simple lexical similarity do not seem to work well. The low factoid recall shown by low unnormalized RU scores suggests integrating the full text of papers with citation based summaries which might help us find factoids such as topic definitions that are unlikely to be present in citing sentences. A final goal would be to improve the readability and coherence of our system output.

## Acknowledgments

We thank Vahed Qazvinian, Wanchen Lu, Ben King, and Shiwali Mohan for extremely useful discussions and help with the data annotation.

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

- Güneş Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.
- Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10*, pages 427–435, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL '09*, pages 584–592, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 926–931.
- Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (HLT-NAACL '04)*.
- Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Manchester, UK.
- Vahed Qazvinian. 2012. *Using Collective Discourse to Generate Surveys of Scientific Paradigms*. Ph.D. thesis.
- Dragomir R. Radev and Daniel Tam. 2003. Summarization evaluation using relative utility. In *CIKM2003*, pages 508–511.
- Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam



Winkel, and Zhu Zhang. 2004. MEAD - a platform for multidocument multilingual text summarization. In *LREC 2004*, Lisbon, Portugal, May.

Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The acl anthology network corpus. *Language Resources and Evaluation*, pages 1–26.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

# A Unified Morpho-Syntactic Scheme of Stanford Dependencies

Reut Tsarfaty

Uppsala University, Sweden

tsarfaty@stp.lingfil.uu.se

## Abstract

Stanford Dependencies (SD) provide a functional characterization of the grammatical relations in syntactic parse-trees. The SD representation is useful for parser evaluation, for downstream applications, and, ultimately, for natural language understanding, however, the design of SD focuses on structurally-marked relations and under-represents morphosyntactic realization patterns observed in Morphologically Rich Languages (MRLs). We present a novel extension of SD, called Unified-SD (U-SD), which unifies the annotation of structurally- and morphologically-marked relations via an inheritance hierarchy. We create a new resource composed of U-SD-annotated constituency and dependency treebanks for the MRL Modern Hebrew, and present two systems that can automatically predict U-SD annotations, for gold segmented input as well as raw texts, with high baseline accuracy.

## 1 Introduction

*Stanford Dependencies (SD)* provide a functional characterization of the grammatical relations in syntactic trees, capturing the predicate-argument structure of natural language sentences (de Marneffe et al., 2006). The SD representation proved useful in a range of downstream tasks, including Textual Entailments (Dagan et al., 2006) and BioNLP (Fundel and Zimmer., 2007), and in recent years SD structures have also become a de-facto standard for parser evaluation in English (de Marneffe and Manning, 2008a; Cer et al., 2010; Nivre et al., 2010). Efforts now commence towards extending SD for cross-lingual annotation

and evaluation (McDonald et al., 2013; Che et al., 2012; Haverinen et al., 2011). By and large, these efforts aim to remain as close as possible to the original SD scheme. However, the original SD design emphasizes word-tokens and configurational structures, and consequently, these schemes overlook properties and realization patterns observed in a range of languages known as *Morphologically Rich Languages (MRLs)* (Tsarfaty et al., 2010).

MRLs use word-level affixes to express grammatical relations that are typically indicated by structural positions in English. By virtue of word-level morphological marking, word-order in MRLs may be flexible. MRLs have been a focal point for the parsing community due to the challenges that these phenomena pose for systems originally developed for English.<sup>1</sup> Here we argue that the SD hierarchy and design principles similarly emphasize English-like structures and under-represent morphosyntactic argument-marking alternatives. We define an extension of SD, called Unified-SD (U-SD), which unifies the annotation of structurally and morphologically marked relations via an inheritance hierarchy. We extend SD with a functional branch, and provide a principled treatment of morpho-syntactic argument marking.

Based on the U-SD scheme we create a new parallel resource for the MRL Modern Hebrew, whereby aligned constituency and dependency trees reflect equivalent U-SD annotations (cf. Rambow (2010)) for the same set of sentences. We present two systems that can automatically learn U-SD annotations, from the dependency and the constituency versions respectively, delivering high baseline accuracy on the prediction task.

<sup>1</sup>See also the SPMRL line of workshops <https://sites.google.com/site/spsemrml2012/> and the MT-MRL workshop <http://cl.haifa.ac.il/MT/>.

## 2 The Challenge: SD for MRLs

Stanford Dependencies (SD) (de Marneffe et al., 2006; de Marneffe and Manning, 2008b) deliver a functional representation of natural language sentences, inspired by theoretical linguistic work such as studies on Relational Grammars (Postal and Perlmutter, 1977), Lexical Functional Grammars (LFG) (Bresnan, 2000) and the PARC dependency scheme (King et al., 2003). At the same time, the scheme is designed with end-users in mind, allowing them to utilize parser output in a form which is intuitively interpretable and easily processed.

SD basic trees represent sentences as binary relations between word tokens. These relations are labeled using traditional grammatical concepts (*subject*, *object*, *modifier*) that are arranged into an inheritance hierarchy (de Marneffe and Manning, 2008a, Sec. 3). There are different versions of SD annotations: the basic SD scheme, which annotates surface dependency relations as a tree spanning all word tokens in the sentence, and the collapsed SD version, in which function words (such as prepositions) are collapsed and used for specifying a direct relation between content words.

The SD scheme defines a core set of labels and principles which are assumed to be useful for different languages. However, a close examination of the SD label-set and inheritance hierarchy reveals that some of its design principles are geared towards English-like (that is, configurational) phenomena, and conflict with basic properties of MRLs. Let us list three such design principles and outline the challenges that they pose.

**2.1. SD relate input-tokens.** In MRLs, substantial information is expressed as word affixes. One or more morphemes may be appended to a content word, and several morphemes may be contained in a single space-delimited token. For example, the Hebrew token *wkfraiti*<sup>2</sup> in (1) includes the morphemes *w* (and), *kf* (when) and *rait* (saw); the latter segment is a content word, and the former two are functional morphemes.

- (1) *wkfraiti* *at*  
 and-when-saw.1st.Singular acc  
*hsrj hifn*  
 the-movie the-old  
*w/and-1.1 kf/when-1.2 raiti/saw-1.3*  
*at/acc-2 h/the-3.1 srj/movie-3.2 h/the-4.1*  
*ifn/old-4.2*

<sup>2</sup>We use the transliteration of Sima'an et al. (2001).

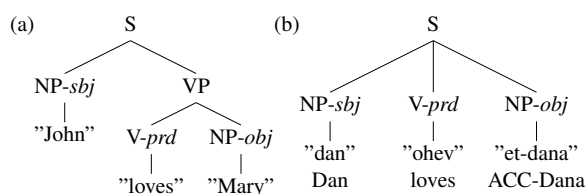


Figure 1: English (a) and Hebrew (b) PS trees decorated with function labels as dash features.

Naïvely taking input tokens as words fails to capture meaningful relations between morphological segments internal to space-delimited tokens.

### 2.2. SD label structurally-marked relations.

Configurational languages like English use function words such as prepositions and auxiliaries to indicate relations between content words and to mark properties of complete structures. In MRLs, such relations and properties may be indicated by word-level morphological marking such as case (Blake, 1994) and agreement (Corbett, 2006). In (1), for instance, the case marker *at* indicates an *accusative* object relation between “see” and “movie”, to be distinguished from, e.g. a *dative* object. Moreover, the agreement in (1) on the *definite* morpheme signals that “old” modifies “movie”. While the original SD scheme label-set covers function words (e.g. *auxpass*, *expl*, *prep*), it misses labels for bound morphemes that mark grammatical relations across languages (such as *accusative*, *dative* or *genitive*). Explicit labeling of such relational morphemes will allow us to benefit from the information that they provide.

### 2.3. SD relations may be inferred using structural cues.

SD relations are extracted from different types of trees for the purpose of, e.g., cross-framework evaluation (Cer et al., 2010). Insofar, recovering SD relations from phrase-structure (PS) trees have used a range of structural cues such as positions and phrase-labels (see, for instance, the software of de Marneffe and Manning (2008a)). In MRLs, positions and phrase types may not suffice for recovering SD relations: an NP under S in Hebrew, for instance, may be a *subject* or an *object*, as shown in Figure 1, and morphological information then determines the function of these constituents. Automatically inferring predicate-argument structures across treebanks thus must rely on both structural and morphological marking, calling for a single annotation scheme that inter-relate the marking alternatives.

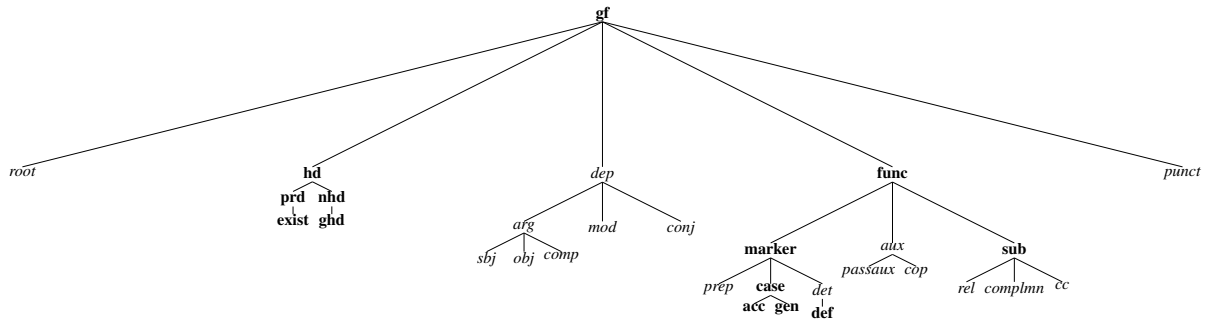


Figure 3: **The Unified SD (U-SD) Ontology.** The architectural changes from the original SD scheme: (i) added a *hd* branch, for implicit head labels; (ii) added a *func* branch where all functional elements (*prep*, *aux*, *cc*, *rel*) as well as morphological markers are moved under; (iii) there is a clear separation between open-class categories (which fall under *hd*, *dep*), closed class elements (under *func*) and non-words (*root* and *punct*). **Boldface** elements are new to U-SD. *Italic* branches spell out further as in the original SD.

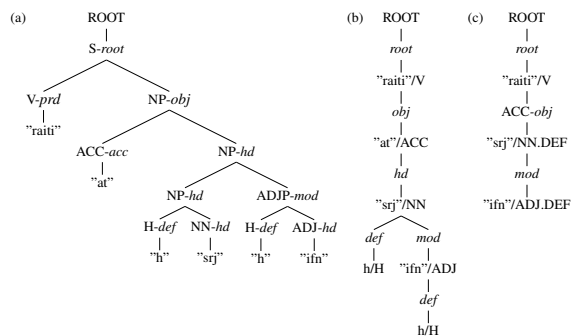


Figure 2: **Sample U-SD Trees** for sentence (1). (a) a phrase-structure tree decorated with U-SD labels, (b) a basic U-SD tree, and (c) a collapsed U-SD tree, where functional nodes are consumed.

### 3 The Proposal: Unified-SD (U-SD)

To address these challenges, we propose an extension of SD called Unified-SD (U-SD) which annotates relations between morphological segments and reflects different types of argument-marking patterns. The SD ontology is re-organized and extended to allow us to annotate morphologically- and structurally-marked relations alike.

**Preliminaries.** We assume that  $\mathcal{M}(w_1 \dots w_n) = s_1 \dots s_m$  is a morphological analysis function that identifies all morphological segments of a sentence  $S = w_1 \dots w_n$ . The U-SD scheme provides the syntactic representation of  $S$  by means of a set of triplets  $(l, s_i, s_j)$  consisting of a label  $l$ , a head  $s_i$  and a dependent  $s_j$  ( $i \neq j$ ). The segments are assumed to be numbered  $x.y$  where  $x$  is the position of the input token, and  $y$  is the position of the segment inside the token. The segmentation numbering is demonstrated in Example (1).

**The U-SD Hierarchy.** Figure 3 shows our proposed U-SD hierarchy. Everything in the ontology is of type *gf* (grammatical function). We define five ontological sub-types: *root*, *hd*, *dep*, *func*, *punct*. The *root* marks a special root dependency. The *dep* branch is used for dependent types, and it retains much of the structure in the original SD scheme (separating *sbj* types, *obj* types, *mod* types, etc.). The new *func* branch contains argument-marking elements, that is, function words and morphemes that play a role in indicating properties or grammatical relations in the syntactic representation. These functional elements may be of types *marker* (prepositions and case), *aux* (auxiliary verbs and copular elements) and *sub* (subordination/conjunction markers). All inherited *func* elements may be consumed (henceforth, collapsed) in order to infer grammatical properties and relations between content words. Head types are implicit in dependency triplets, however, when decorating PS trees with dependency labels as dash features or edge features (as in TigerXML formats (Brants et al., 2002) or via unification-based formalisms) both heads and dependents are labeled with their grammatical types (see Figure 2(a)). The *hd* branch extends the scheme with an inventory of argument-taking elements, to be used when employing SD inside constituency treebanks. The *punct* branch is reserved for punctuation, prosody and other non-verbal speech acts. The complete ontology is given in the appendix.

**Annotation Guidelines.** Anderson (1992) delineates three kinds of properties that are realized by morphology: *structural*, *inherent*, and *agreement* properties. Structural properties (e.g., case) are marked on a content word to indicate its rela-

|       | Segments | Functions |               | Segments   | Functions |      | Segments      | Functions |     |        |               |
|-------|----------|-----------|---------------|------------|-----------|------|---------------|-----------|-----|--------|---------------|
| Gold: | DEP      | 1.00      | 0.8475        | Predicted: | DEP       | 1.00 | 0.8349        | Raw:      | DEP | 0.9506 | 0.7817        |
|       | RR       | 1.00      | <b>0.8984</b> |            | RR        | 1.00 | <b>0.8559</b> |           | RR  | 0.9603 | <b>0.8130</b> |

Table 1: Inferring U-SD trees using different frameworks. All numbers report labeled TedEval accuracy.

tion to other parts of the sentence. Inherent properties (gender, number, etc.) indicate inherent semantic properties of nominals. Agreement properties indicate the semantic properties of nominals on top of other elements (verbs, adjectives, etc.), in order to indicate their relation to the nominals.

We define annotation guidelines that reflect these different properties. Structural morphemes (case) connect words in the arc-structure, linking a head to its semantic dependent, like the case marker “at”-ACC in Figure 2(b). Inherent / agreement properties are annotated as dependents of the content word they add properties to, for instance, the prefixes *def* in Figure 2(b) hang under the modified noun and adjective.

Collapsed U-SD structures interpret *func* elements in order to refine the representation of relations between content words. Case markers can be used for refining the relation between the content words they connect by labeling their direct relation, much like *prep* in the original SD scheme (see, e.g., the ACC-*obj* in Figure 2c). Inherent/agreement features are in fact features of their respective head word (as the X.DEF nodes in Figure 2c).<sup>3</sup> Auxiliaries may further be used to add tense/aspect to the main predicate, and subordinators may propagate information inside the structure (much like conjunction is propagated in SD).

**Universal Aspects of U-SD.** The revised U-SD ontology provides a typological inventory of labels that describe different types of arguments (*dep*), argument-taking elements (*hd*), and argument-marking elements (*func*) in the grammar of different languages. Abstract (universal) concepts reside high in the hierarchy, and more specific distinctions, e.g., morphological markers of particular types, are daughters within more specific branches. Using U-SD for evaluating monolingual parsers is best done with the complete label set relevant for that language. For cross-language evaluation, we can limit the depth of the hierarchy, and convert the more specific notions to their most-specific ancestor in the evaluation set.

<sup>3</sup>Technically, this is done by deleting a line adding a property to the morphology column in the CoNLL format.

## 4 Automatic Annotation of U-SD Trees

Can U-SD structures be automatically predicted? For MRLs, this requires disambiguating both morphological and syntactic information. Here we employ the U-SD scheme for annotating morphosyntactic structures in Modern Hebrew, and use these resources to train two systems that predict U-SD annotations for raw texts.<sup>4</sup>

**Data.** We use the Modern Hebrew treebank (Sima’an et al., 2001), a corpus of 6220 sentences morphologically segmented and syntactically analyzed as PS trees. We infer the function label of each node in the PS trees based on the morphological features, syntactic environment, and dash-feature (if exist), using deterministic grammar rules (Glinert, 1989). Specifically, we compare each edge with a set of templates, and, once finding a template that fits the morphological and syntactic profile of an edge, we assign functions to all daughters. This delivers PS trees where each node is annotated with a U-SD label (Figure 2a). At a second stage we project the inferred labels onto the arcs of the unlabeled dependency trees of Goldberg (2011), using the tree unification operation of Tsarfaty et al. (2012a). The result is a dependency tree aligned with the constituency tree where dependency arcs are labeled with the same function as the respective span in the PS tree.<sup>5</sup>

**Systems.** We present two systems that predict U-SD labels along with morphological and syntactic information, using [DEP], a dependency parser (Nivre et al., 2007), and [RR], a Relational-Realizational (RR) constituency parser (Tsarfaty and Sima’an, 2008). DEP is trained directly on the dependency version of the U-SD resource. Since it cannot predict its own segmentation, automatic segments and tags are predicted using the system of Adler and Elhadad (2006). The constituency-

<sup>4</sup>Despite significant advances in parsing Hebrew, as of yet there has been no functional evaluation of Hebrew parsers. E.g., Goldberg and Elhadad (2010) evaluate on unlabeled dependencies, Tsarfaty (2010) evaluate on constituents. This is largely due to the lack of standard resources and guidelines for annotating functional structures in such a language.

<sup>5</sup>The resources can be downloaded at <http://www.tsarfaty.com/heb-sd/>.

based model is trained on U-SD-labeled RR trees using Petrov et al. (2006). We use the lattice-based extension of Goldberg and Elhadad (2011) to perform joint segmentation and parsing. We evaluate three input scenarios: **[Gold]** gold segmentation and gold tags, **[Predicted]** gold segments, and **[Raw]** raw words. We evaluate parsing results with respect to basic U-SD trees, for 42 labels. We use TedEval for joint segmentation-tree evaluation (Tsarfaty et al., 2012b) and follow the cross-parser evaluation protocol of Tsarfaty et al. (2012a).

**Results.** Since this work focuses on creating a new resource, we report results on the standard devset (Table 1). The gold input scenarios obtain higher accuracy on function labels in all cases, since gold morphological analysis delivers disambiguated functions almost for free. Constituency-based RR structures obtain better accuracy on U-SD annotations than the respective dependency parser. All in all, the U-SD seed we created allows us to infer rich interpretable annotations automatically for raw text, using either a dependency parser or a constituency parser, in good accuracy.

## 5 Conclusion

The contribution of this paper is three-fold. We offer a principled treatment of annotating MRLs via a Unified-SD scheme, which we design to be applicable to many languages. We deliver new U-SD annotated resources for the MRL Modern Hebrew, in different formal types. We finally present two systems that automatically predict U-SD annotations for raw texts. These structures are intended to serve semantic applications. We further intend to use this scheme and computational frameworks to serve a wide cross-parser investigation on inferring functional structures across languages.

## Appendix: The U-SD Ontology

The list in (2) presents the complete U-SD ontology. The hierarchy employs and extends the SD label set of de Marneffe et al. (2006). For readability, we omit here various compound types under *mod*, including *nn*, *mwe*, *predet* and *preconj*.

## Acknowledgements

We thank Joakim Nivre, Yoav Goldberg, Djamel Seddah and anonymous reviewers for comments and discussion. This research was partially funded by the Swedish Research Council. The author is now a researcher at the Weizmann Institute.

- (2) *gf* *root* - root  
*hd* - head (governor, argument-taking)  
*prd* - verbal predicate  
*exist* - head of an existential phrase  
*nhd* - head of a nominal phrase  
*ghd* - genitive head of a nominal phrase  
*dep* - dependent (governed, or an argument)  
*arg* - argument  
*agent* - agent  
*comp* - complement  
*acomp* - adjectival complement  
*ccomp* - comp clause with internal sbj  
*xcomp* - comp clause with external sbj  
*pcomp* - comp clause of a preposition  
*obj* - object  
*dobj* - direct object  
*gobj* - genitive object  
*iobj* - indirect object  
*pobj* - object of a preposition  
*subj* - subject  
*expl* - expletive subject  
*nsubj* - nominal subject  
 — *nsubjpass* - passive nominal sbj  
*csubj* - clausal subject  
 — *csubjpass* - passive clausal sbj  
*mod* - modifier  
*appos* - apposition/parenthetical  
*abbrev* - abbreviation  
*amod* - adjectival modifier  
*advmod* - adverbial modifier  
 — *neg* - negative modifier  
*prepm* - prepositional modifier  
 — *possmod* - possession modifier  
 — *tmod* - temporal modifier  
*remod* - relative clause modifier  
*infmod* - infinitival modifier  
*nummod* - numerical modifier  
*parataxis* - "side-by-side", interjection  
*conj* - conjunct  
*func* - functional (argument marking)  
*marker* - nominal-marking elements  
*prep* - preposition  
*case* - case marker  
 — *acc* - accusative case  
 — *dat* - dative case  
 — *gen* - genitive case  
 — *nom* - nominative case  
*det* - determiner  
 — *def* - definite marker  
 — *dem* - demonstrative  
*sub* - phrase-marking elements  
*complm* - introducing comp phrase  
*rel* - introducing relative phrase  
*cc* - introducing conjunction  
*mark* - introducing an advb phrase  
*aux* - auxiliary verb or a feature-bundle  
*auxpass* - passive auxiliary  
*cop* - copular element  
*modal* - modal verb  
*qaux* - question auxiliary  
*punct* - punctuation

## References

- Meni Adler and Michael Elhadad. 2006. An unsupervised morpheme-based HMM for Hebrew morphological disambiguation. In *Proceedings of COLING-ACL*.
- Stephen R. Anderson. 1992. *A-Morphous Morphology*. Cambridge University Press.
- Barry J. Blake. 1994. *Case*. Cambridge University Press, Cambridge.
- Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of TLT*.
- Joan Bresnan. 2000. *Lexical-Functional Syntax*. Blackwell.
- Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC*.
- Wanxiang Che, Valentin I. Spitzkovsky, and Ting Liu. 2012. A comparison of chinese parsers for stanford dependencies. In *Proceedings of ACL*, pages 11–16.
- Greville G. Corbett. 2006. *Agreement*. Cambridge Textbooks in Linguistics. Cambridge University Press.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *MLCW 2005, LNAI Volume*.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2008a. Stanford dependencies manual. Technical Report.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2008b. The stanford typed dependencies representation. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, pages 449–454.
- Robert Kuffner Fundel, Katrin and Ralf Zimmer. 2007. RelEx relation extraction using dependency parse trees. *Bioinformatics*, (23).
- Lewis Glinert. 1989. *The Grammar of Modern Hebrew*. Cambridge University Press.
- Yoav Goldberg and Michael Elhadad. 2010. Easy-first dependency parsing of Modern Hebrew. In *Proceedings of NAACL/HLT workshop on Statistical Parsing of Morphologically Rich Languages*.
- Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFGLA lattice parser. In *Proceedings of ACL*.
- Yoav Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ph.D. thesis, Ben Gurion University of the Negev.
- Katri Haverinen, Filip Ginter, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. 2011. A dependency-based analysis of treebank annotation errors. In *Proceedings of DepLing*.
- Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 dependency bank. In *The 4th International Workshop on Linguistically Interpreted Corpora*.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL*.
- Joakim Nivre, Jens Nilsson, Johan Hall, Atanas Chaney, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(1):1–41.
- Joakim Nivre, Laura Rimell, Ryan McDonald, and Carlos Gómez-Rodríguez. 2010. Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of COLING*, pages 813–821.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL*.
- Paul M. Postal and David M. Perlmutter. 1977. Toward a universal characterization of passivization. In *BLS* 3.
- Owen Rambow. 2010. The Simple Truth about Dependency and Phrase Structure Representations: An Opinion Piece. In *Proceedings of HLT-ACL*.
- Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In *Traitement Automatique des Langues*.
- Reut Tsarfaty and Khalil Sima'an. 2008. Relational-realizational parsing. In *Proceedings of CoLing*.
- Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kuebler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing for morphologically rich language (SPMRL): What, how and whither. In *Proceedings of the first workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL) at NA-ACL*.
- Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012a. Cross-framework evaluation for statistical parsing. In *Proceeding of EACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson.  
2012b. Joint evaluation for segmentation and parsing. In *Proceedings of ACL*.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.



# Dependency Parser Adaptation with Subtrees from Auto-Parsed Target Domain Data

**Xuezhe Ma**

Department of Linguistics  
University of Washington  
Seattle, WA 98195, USA  
xzma@uw.edu

**Fei Xia**

Department of Linguistics  
University of Washington  
Seattle, WA 98195, USA  
fxia@uw.edu

## Abstract

In this paper, we propose a simple and effective approach to domain adaptation for dependency parsing. This is a feature augmentation approach in which the new features are constructed based on subtree information extracted from the auto-parsed target domain data. To demonstrate the effectiveness of the proposed approach, we evaluate it on three pairs of source-target data, compared with several common baseline systems and previous approaches. Our approach achieves significant improvement on all the three pairs of data sets.

## 1 Introduction

In recent years, several dependency parsing algorithms (Nivre and Scholz, 2004; McDonald et al., 2005a; McDonald et al., 2005b; McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010; Ma and Zhao, 2012) have been proposed and achieved high parsing accuracies on several treebanks of different languages. However, the performance of such parsers declines when training and test data come from different domains. Furthermore, the manually annotated treebanks that these parsers rely on are highly expensive to create. Therefore, developing dependency parsing algorithms that can be easily ported from one domain to another—say, from a resource-rich domain to a resource-poor domain—is of great importance.

Several approaches have been proposed for the task of parser adaptation. McClosky et al. (2006) successfully applied self-training to domain adaptation for constituency parsing using the reranking parser of Charniak and Johnson (2005). Reichart and Rappoport (2007) explored self-training when the amount of the annotated data is small

and achieved significant improvement. Zhang and Wang (2009) enhanced the performance of dependency parser adaptation by utilizing a large-scale hand-crafted HPSG grammar. Plank and van Noord (2011) proposed a data selection method based on effective measures of domain similarity for dependency parsing.

There are roughly two varieties of domain adaptation problem—fully supervised case in which there are a small amount of labeled data in the target domain, and semi-supervised case in which there are no labeled data in the target domain. In this paper, we present a parsing adaptation approach focused on the fully supervised case. It is a feature augmentation approach in which the new features are constructed based on subtree information extracted from the auto-parsed target domain data. For evaluation, we run experiments on three pairs of source-target domains—WSJ-Brown, Brown-WSJ, and WSJ-Genia. Our approach achieves significant improvement on all these data sets.

## 2 Our Approach for Parsing Adaptation

Our approach is inspired by Chen et al. (2009)’s work on semi-supervised parsing with additional subtree-based features extracted from unlabeled data and by the feature augmentation method proposed by Daume III (2007). In this section, we first summarize Chen et al.’s work and explain how we extend that for domain adaptation. We will then highlight the similarity and difference between our work and Daume’s method.

### 2.1 Semi-supervised parsing with subtree-based features

One of the most well-known semi-supervised parsing methods is self-training, where a parser trained from the labeled data set is used to parse unlabeled data, and some of those auto-parsed data are added to the labeled data set to retrain the pars-

ing models. Chen et al. (2009)’s approach differs from self-training in that partial information (i.e., subtrees), instead of the entire trees, from the auto-parsed data is used to re-train the parsing models.

A subtree is a small part of a dependency tree. For example, a first-order subtree is a single edge consisting of a head and a dependent, and a second-order sibling subtree is one that consists of a head and two dependents. In Chen et al. (2009), they first extract all the subtrees in auto-parsed data and store them in a list  $L_{st}$ . Then they count the frequency of these subtrees and divide them into three groups according to their levels of frequency. Finally, they construct new features for the subtrees based on which groups they belongs to and retrain a new parser with feature-augmented training data.<sup>1</sup>

## 2.2 Parser adaptation with subtree-based Features

Chen et al. (2009)’s work is for semi-supervised learning, where the labeled training data and the test data come from the same domain; the subtree-based features collected from auto-parsed data are added to all the labeled training data to retrain the parsing model. In the supervised setting for domain adaptation, there is a large amount of labeled data in the source domain and a small amount of labeled data in the target domain. One intuitive way of applying Chen’s method to this setting is to simply take the union of the labeled training data from both domains and add subtree-based features to all the data in the union when re-training the parsing model. However, it turns out that adding subtree-based features to only the labeled training data in the target domain works better. The steps of our approach are as follows:

1. Train a baseline parser with the small amount of labeled data in the target domain and use the parser to parse the large amount of unlabeled sentences in the target domain.
2. Extract subtrees from the auto-parsed data and add subtree-based features to the labeled training data in the target domain.
3. Retrain the parser with the union of the labeled training data in the two domains, where the instances from the target domain are augmented with the subtree-based features.

<sup>1</sup>If a subtree does not appear in  $L_{st}$ , it falls to the fourth group for “unseen subtrees”.

To state our feature augmentation approach more formally, we use  $X$  to denote the input space, and  $D^s$  and  $D^t$  to denote the labeled data in the source and target domains, respectively. Let  $X'$  be the augmented input space, and  $\Phi^s$  and  $\Phi^t$  be the mappings from  $X$  to  $X'$  for the instances in the source and target domains respectively. The mappings are defined by Eq 1, where  $\mathbf{0} = \langle 0, 0, \dots, 0 \rangle \in X$  is the zero vector.

$$\begin{aligned}\Phi^s(\mathbf{x}_{org}) &= \langle \mathbf{x}_{org}, \mathbf{0} \rangle \\ \Phi^t(\mathbf{x}_{org}) &= \langle \mathbf{x}_{org}, \mathbf{x}_{new} \rangle\end{aligned}\quad (1)$$

Here,  $\mathbf{x}_{org}$  is the original feature vector in  $X$ , and  $\mathbf{x}_{new}$  is the vector of the subtree-based features extracted from auto-parsed data of the target domain. The subtree extraction method used in our approach is the same as in (Chen et al., 2009) except that we use different thresholds when dividing subtrees into three frequency groups: the threshold for the high-frequency level is TOP 1% of the subtrees, the one for the middle-frequency level is TOP 10%, and the rest of subtrees belong to the low-frequency level. These thresholds are chosen empirically on some development data set.

The idea of distinguishing the source and target data is similar to the method in (Daume III, 2007), which did feature augmentation by defining the following mappings:<sup>2</sup>

$$\begin{aligned}\Phi^s(\mathbf{x}_{org}) &= \langle \mathbf{x}_{org}, \mathbf{0} \rangle \\ \Phi^t(\mathbf{x}_{org}) &= \langle \mathbf{x}_{org}, \mathbf{x}_{org} \rangle\end{aligned}\quad (2)$$

Daume III showed that differentiating features from the source and target domains improved performance for multiple NLP tasks. The difference between that study and our approach is that our new features are based on subtree information instead of copies of original features. Since the new features are based on the subtree information extracted from the auto-parsed target data, they represent certain properties of the target domain and that explains why adding them to the target data works better than adding them to both the source and target data.

## 3 Experiments

For evaluation, we tested our approach on three pairs of source-target data and compared it with

<sup>2</sup>The mapping in Eq 2 looks different from the one proposed in (Daume III, 2007), but it can be proved that the two are equivalent.

several common baseline systems and previous approaches. In this section, we first describe the data sets and parsing models used in each of the three experiments in section 3.1. Then we provide a brief introduction to the systems we have reimplemented for comparison in section 3.2. The experimental results are reported in section 3.3.

### 3.1 Data and Tools

In the first two experiments, we used the Wall Street Journal (WSJ) and Brown (B) portions of the English Penn TreeBank (Marcus et al., 1993). In the first experiment denoted by “WSJ-to-B”, WSJ corpus is used as the source domain and Brown corpus as the target domain. In the second experiment, we use the reverse order of the two corpora and denote it by “B-to-WSJ”. The phrase structures in the treebank are converted into dependencies using Penn2Malt tool<sup>3</sup> with the standard head rules (Yamada and Matsumoto, 2003).

For the WSJ corpus, we used the standard data split: sections 2-21 for training and section 23 for test. In the experiment of B-to-WSJ, we randomly selected about 2000 sentences from the training portion of WSJ as the labeled data in the target domain. The rest of training data in WSJ is regarded as the unlabeled data of the target domain.

For Brown corpus, we followed Reichart and Rappoport (2007) for data split. The training and test sections consist of sentences from all of the genres that form the corpus. The training portion consists of 90% (9 of each 10 consecutive sentences) of the data, and the test portion is the remaining 10%. For the experiment of WSJ-to-B, we randomly selected about 2000 sentences from training portion of Brown and use them as labeled data and the rest as unlabeled data in the target domain.

In the third experiment denoted by “WSJ-to-G”, we used WSJ corpus as the source domain and Genia corpus (G)<sup>4</sup> as the target domain. Following Plank and van Noord (2011), we used the training data in CoNLL 2008 shared task (Surdeanu et al., 2008) which are also from WSJ sections 2-21 but converted into dependency structure by the LTH converter (Johansson and Nugues, 2007). The Genia corpus is converted to CoNLL format with LTH converter, too. We randomly selected

<sup>3</sup><http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html>

<sup>4</sup>Genia distribution in Penn Treebank format is available at <http://bllip.cs.brown.edu/download/genia1.0-division-rel1.tar.gz>

|          | Source training | Target   |           |       |
|----------|-----------------|----------|-----------|-------|
|          |                 | training | unlabeled | test  |
| WSJ-to-B | 39,832          | 2,182    | 19,632    | 2,429 |
| B-to-WSJ | 21,814          | 2,097    | 37,735    | 2,416 |
| WSJ-to-G | 39,279          | 1,024    | 13,302    | 1,360 |

Table 1: The number of sentences for each data set used in our experiments

about 1000 sentences from the training portion of Genia data and use them as the labeled data of the target domain, and the rest of training data of Genia as the unlabeled data of the target domain. Table 1 shows the number of sentences of each data set used in the experiments.

The dependency parsing models we used in this study are the graph-based first-order and second-order sibling parsing models (McDonald et al., 2005a; McDonald and Pereira, 2006). To be more specific, we use the implementation of MaxParser<sup>5</sup> with 10-best MIRA (Crammer et al., 2006; McDonald, 2006) learning algorithm and each parser is trained for 10 iterations. The feature sets of first-order and second-order sibling parsing models used in our experiments are the same as the ones in (Ma and Zhao, 2012). The input to MaxParser are sentences with Part-of-Speech tags; we use gold-standard POS tags in the experiments.

Parsing accuracy is measured with unlabeled attachment score (UAS) and the percentage of complete matches (CM) for the first and second experiments. For the third experiment, we also report labeled attachment score (LAS) in order to compare with the results in (Plank and van Noord, 2011).

### 3.2 Comparison Systems

For comparison, we re-implemented the following well-known baselines and previous approaches, and tested them on the three data sets:

**SrcOnly:** Train a parser with the labeled data from the source domain only.

**TgtOnly:** Train a parser with the labeled data from the target domain only.

**Src&Tgt:** Train a parser with the labeled data from the source and target domains.

**Self-Training:** Following Reichart and Rappoport (2007), we train a parser with the union of the source and target labeled data, parse the unlabeled data in the target domain,

<sup>5</sup><http://sourceforge.net/projects/maxparser/>

add the entire auto-parsed trees to the manually labeled data in a single step without checking their parsing quality, and retrain the parser.

**Co-Training:** In the co-training system, we first train two parsers with the labeled data from the source and target domains, respectively. Then we use the parsers to parse unlabeled data in the target domain and select sentences for which the two parsers produce identical trees. Finally, we add the analyses for those sentences to the union of the source and target labeled data to retrain a new parser. This approach is similar to the one used in (Sagae and Tsujii, 2007), which achieved the highest scores in the domain adaptation track of the CoNLL 2007 shared task (Nivre et al., 2007).

**Feature-Augmentation:** This is the approach proposed in (Daume III, 2007).

**Chen et al. (2009):** The algorithm has been explained in Section 2.1. We use the union of the labeled data from the source and target domains as the labeled training data. The unlabeled data needed to construct subtree-based features come from the target domain.

**Plank and van Noord (2011):** This system performs data selection on a data pool consisting of large amount of labeled data to get a training set that is similar to the test domain. The results of the system come from their paper, not from the reimplementations of their system.

**Per-corpus:** The parser is trained with the large training set from the target domain. For example, for the experiment of WSJ-to-B, all the labeled training data from the Brown corpus is used for training, including the subset of data which are treated as unlabeled in our approach and other comparison systems. The results serve as an upper bound of domain adaptation when there is a large amount of labeled data in the target domain.

### 3.3 Results

Table 2 illustrates the results of our approach with the first-order parsing model in the first and second experiments, together with the results of the comparison systems described in section 3.2. The

|                                 | WSJ-to-B    |             | B-to-WSJ    |             |
|---------------------------------|-------------|-------------|-------------|-------------|
|                                 | UAS         | CM          | UAS         | CM          |
| SrcOnly <sup>s</sup>            | 88.8        | 43.8        | 86.3        | 26.5        |
| TgtOnly <sup>t</sup>            | 86.6        | 38.8        | 88.2        | 29.3        |
| Src&Tgt <sup>s,t</sup>          | 89.1        | 44.3        | 89.4        | 31.2        |
| Self-Training <sup>s,t</sup>    | 89.2        | 45.1        | 89.8        | 32.1        |
| Co-Training <sup>s,t</sup>      | 89.2        | 45.1        | 89.8        | 32.7        |
| Feature-Aug <sup>s,t</sup>      | 89.1        | 45.1        | 89.8        | 32.8        |
| Chen (2009) <sup>s,t</sup>      | 89.3        | 45.0        | 89.7        | 31.8        |
| <b>this paper<sup>s,t</sup></b> | <b>89.5</b> | <b>45.5</b> | <b>90.2</b> | <b>33.4</b> |
| Per-corpus <sup>T</sup>         | 89.9        | 47.0        | 92.7        | 42.1        |

Table 2: Results with the first-order parsing model in the first and second experiments. The superscript indicates the source of labeled data used in training.

|                                 | WSJ-to-B    |             | B-to-WSJ    |             |
|---------------------------------|-------------|-------------|-------------|-------------|
|                                 | UAS         | CM          | UAS         | CM          |
| SrcOnly <sup>s</sup>            | 89.8        | 47.3        | 88.0        | 30.4        |
| TgtOnly <sup>t</sup>            | 87.7        | 42.2        | 89.7        | 34.2        |
| Src&Tgt <sup>s,t</sup>          | 90.2        | 48.2        | 90.9        | 36.6        |
| Self-Training <sup>s,t</sup>    | 90.3        | 48.8        | 91.0        | 37.1        |
| Co-Training <sup>s,t</sup>      | 90.3        | 48.5        | 90.9        | 38.0        |
| Feature-Aug <sup>s,t</sup>      | 90.0        | 48.4        | 91.0        | 37.4        |
| Chen (2009) <sup>s,t</sup>      | 90.3        | 49.1        | 91.0        | 37.6        |
| <b>this paper<sup>s,t</sup></b> | <b>90.6</b> | <b>49.6</b> | <b>91.5</b> | <b>38.8</b> |
| Per-corpus <sup>T</sup>         | 91.1        | 51.1        | 93.6        | 47.9        |

Table 3: Results with the second-order sibling parsing model in the first and second experiments.

results with the second-order sibling parsing model is shown in Table 3. The superscript  $s$ ,  $t$  and  $T$  indicates from which domain the labeled data are used in training: tag  $s$  refers to the labeled data in the source domain, tag  $t$  refers to the small amount of labeled data in the target domain, and tag  $T$  indicates that all the labeled training data from the target domain, including the ones that are treated as unlabeled in our approach, are used for training.

Table 4 shows the results in the third experiment with the first-order parsing model. We also include the result from (Plank and van Noord, 2011), which use the same parsing model as ours. Note that this result is not comparable with other numbers in the table as it uses a larger set of labeled data, as indicated by the  $\dagger$  superscript.

All three tables show that our system outperforms the comparison systems in all three

|                                 | WSJ-to-G    |             |
|---------------------------------|-------------|-------------|
|                                 | UAS         | LAS         |
| SrcOnly <sup>s</sup>            | 83.8        | 82.0        |
| TgtOnly <sup>t</sup>            | 87.0        | 85.7        |
| Src&Tgt <sup>s,t</sup>          | 87.2        | 85.9        |
| Self-Training <sup>s,t</sup>    | 87.3        | 86.0        |
| Co-Training <sup>s,t</sup>      | 87.3        | 86.0        |
| Feature-Aug <sup>s,t</sup>      | 87.9        | 86.5        |
| Chen (2009) <sup>s,t</sup>      | 87.5        | 86.2        |
| <b>this paper<sup>s,t</sup></b> | <b>88.4</b> | <b>87.1</b> |
| Plank (2011) <sup>†</sup>       | -           | 86.8        |
| Per-corpus <sup>T</sup>         | 90.5        | 89.7        |

Table 4: Results with first-order parsing model in the third experiment. “Plank (2011)” refers to the approach in Plank and van Noord (2011).

experiments.<sup>6</sup> The improvement of our approach over the feature augmentation approach in Daume III (2007) indicates that adding subtree-based features provides better results than making several copies of the original features. Our system outperforms the system in (Chen et al., 2009), implying that adding subtree-based features to only the target labeled data is better than adding them to the labeled data in both the source and target domains.

Considering the three steps of our approach in Section 2.2, the training data used to train the parser in Step 1 can be from the target domain only or from the source and target domains. Similarly, in Step 3 the subtree-based features can be added to the labeled data from the target domain only or from the source and target domains. Therefore, there are four combinations. Our approach is the one that uses the labeled data from the target domain only in both steps, and Chen’s system uses labeled data from the source and target domains in both steps. Table 5 compares the performance of the final parser in the WSJ-to-Genia experiment when the parser is created with one of the four combinations. The column label and the row label indicate the choice in Step 1 and 3, respectively. The table shows the choice in Step 1 does not have a significant impact on the performance of the final models; in contrast, the choice in Step 3 does matter—adding subtree-based features to the labeled data in the target domain only is much better than adding features to the data in both domains.

<sup>6</sup>The results of Per-corpus are better than ours but it uses a much larger labeled training set in the target domain.

|         | TgtOnly   | Src&Tgt   |
|---------|-----------|-----------|
| TgtOnly | 88.4/87.1 | 88.4/87.1 |
| Src&Tgt | 87.6/86.3 | 87.5/86.2 |

Table 5: The performance (UAS/LAS) of the final parser in the WSJ-to-Genia experiment when different training data are used to create the final parser. The column label and row label indicate the choice of the labeled data used in Step 1 and 3 of the process described in Section 2.2.

## 4 Conclusion

In this paper, we propose a feature augmentation approach for dependency parser adaptation which constructs new features based on subtree information extracted from auto-parsed data from the target domain. We distinguish the source and target domains by adding the new features only to the data from the target domain. The experimental results on three source-target domain pairs show that our approach outperforms all the comparison systems.

For the future work, we will explore the potential benefits of adding other types of features extracted from unlabeled data in the target domain. We will also experiment with various ways of combining our current approach with other domain adaptation methods (such as self-training and co-training) to further improve system performance.

## References

- Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CONLL*, pages 957–961.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine-grained  $n$ -best parsing and discriminative r-ranking. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 132–139.
- Wenliang Chen, Jun’ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 570–579, Singapore, August.
- 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.

- Hal Daume 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, June.
- Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA*, Tartu, Estonia.
- Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of 48th Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1–11, Uppsala, Sweden, July.
- Xuezhe Ma and Hai Zhao. 2012. Fourth-order dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 785–796, Mumbai, India, December.
- Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 337–344, Sydney, Australia, July.
- Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of European Association for Computational Linguistics (EACL-2006)*, pages 81–88, Trento, Italy, April.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL-2005)*, pages 91–98, Ann Arbor, Michigan, USA, June 25-30.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language (HLT/EMNLP 05)*, pages 523–530, Vancouver, Canada, October.
- Ryan McDonald. 2006. *Discriminative learning spanning tree algorithm for dependency parsing*. Ph.D. thesis, University of Pennsylvania.
- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *Proceedings of the 20th international conference on Computational Linguistics (COLING'04)*, pages 64–70, Geneva, Switzerland, August 23-27.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, 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 2007*, pages 915–932, Prague, Czech, June.
- Barbara Plank and Gertjan 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 2011)*, pages 1566–1576, Portland, Oregon, USA, June.
- Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-2007)*, pages 616–623, Prague, Czech Republic, June.
- Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic, June.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Marquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 159–177, Manchester, UK, August.
- Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT-2003)*, pages 195–206, Nancy, France, April.
- Yi Zhang and Rui Wang. 2009. Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009)*, pages 378–386, Suntec, Singapore, August.

# Iterative Transformation of Annotation Guidelines for Constituency Parsing

Xiang Li<sup>1, 2</sup> Wenbin Jiang<sup>1</sup> Yajuan Lü<sup>1</sup> Qun Liu<sup>1, 3</sup>

<sup>1</sup>Key Laboratory of Intelligent Information Processing  
Institute of Computing Technology, Chinese Academy of Sciences  
{lixiang, jiangwenbin, lvyajuan}@ict.ac.cn

<sup>2</sup>University of Chinese Academy of Sciences

<sup>3</sup>Centre for Next Generation Localisation  
Faculty of Engineering and Computing, Dublin City University  
qliu@computing.dcu.ie

## Abstract

This paper presents an effective algorithm of annotation adaptation for constituency treebanks, which transforms a treebank from one annotation guideline to another with an iterative optimization procedure, thus to build a much larger treebank to train an enhanced parser without increasing model complexity. Experiments show that the transformed Tsinghua Chinese Treebank as additional training data brings significant improvement over the baseline trained on Penn Chinese Treebank only.

## 1 Introduction

Annotated data have become an indispensable resource for many natural language processing (NLP) applications. On one hand, the amount of existing labeled data is not sufficient; on the other hand, however there exists multiple annotated data with incompatible annotation guidelines for the same NLP task. For example, the People's Daily corpus (Yu et al., 2001) and Chinese Penn Treebank (CTB) (Xue et al., 2005) are publicly available for Chinese segmentation.

An available treebank is a major resource for syntactic parsing. However, it is often a key bottleneck to acquire credible treebanks. Various treebanks have been constructed based on different annotation guidelines. In addition to the most popular CTB, Tsinghua Chinese Treebank (TCT) (Zhou, 2004) is another real large-scale treebank for Chinese constituent parsing. Figure 1 illustrates some differences between CTB and TCT in grammar category and syntactic structure. Unfortunately, these heterogeneous treebanks can not

be directly merged together for training a parsing model. Such divergences cause a great waste of human effort. Therefore, it is highly desirable to transform a treebank into another compatible with another annotation guideline.

In this paper, we focus on harmonizing heterogeneous treebanks to improve parsing performance. We first propose an effective approach to automatic treebank transformation from one annotation guideline to another. For convenience of reference, a treebank with our desired annotation guideline is named as target treebank, and a treebank with a different annotation guideline is named as source treebank. Our approach proceeds in three steps. A parser is firstly trained on source treebank. It is used to relabel the raw sentences of target treebank, to acquire parallel training data with two heterogeneous annotation guidelines. Then, an annotation transformer is trained on the parallel training data to model the annotation inconsistencies. In the last step, a parser trained on target treebank is used to generate  $k$ -best parse trees with target annotation for source sentences. Then the optimal parse trees are selected by the annotation transformer. In this way, the source treebank is transformed to another with our desired annotation guideline. Then we propose an optimization strategy of iterative training to further improve the transformation performance. At each iteration, the annotation transformation of source-to-target and target-to-source are both performed. The transformed treebank is used to provide better annotation guideline for the parallel training data of next iteration. As a result, the better parallel training data will bring an improved annotation transformer at next iteration.

We perform treebank transformation from TC-

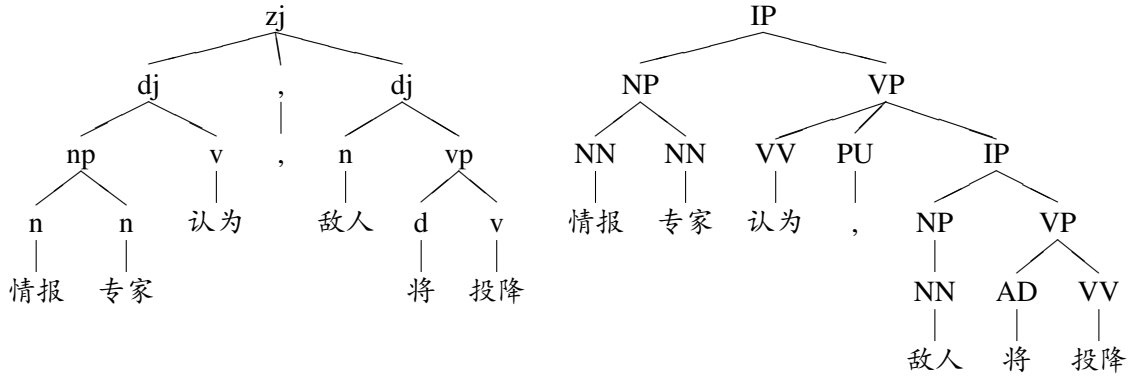


Figure 1: Example heterogeneous trees with TCT (left) and CTB (right) annotation guidelines.

T to CTB, in order to obtain additional treebank to improve a parser. Experiments on Chinese constituent parsing show that, the iterative training strategy outperforms the basic annotation transformation baseline. With additional transformed treebank, the improved parser achieves an F-measure of 0.95% absolute improvement over the baseline parser trained on CTB only.

## 2 Automatic Annotation Transformation

In this section, we present an effective approach that transforms the source treebank to another compatible with the target annotation guideline, then describe an optimization strategy of iterative training that conducts several rounds of bidirectional annotation transformation and improves the transformation performance gradually from a global view.

### 2.1 Principle for Annotation Transformation

In training procedure, the source parser is used to parse the sentences in the target treebank so that there are  $k$ -best parse trees with the source annotation guideline and one gold tree with the target annotation guideline for each sentence in the target treebank. This parallel data is used to train a source-to-target tree transformer. In transformation procedure, the source  $k$ -best parse trees are first generated by a parser trained on the target treebank. Then the optimal source parse trees with target annotation are selected by the annotation transformer with the help of gold source parse trees. By combining the target treebank with the transformed source treebank, it can improve parsing accuracy using a parser trained on the enlarged treebank.

Algorithm 1 shows the training procedure of treebank annotation transformation.  $treebank_s$

and  $treebank_t$  denote the source and target treebank respectively.  $parser_s$  denotes the source parser.  $transformer_{s \rightarrow t}$  denotes the annotation transformer.  $treebank_m^n$  denotes  $m$  treebank re-labeled with  $n$  annotation guideline. Function TRAIN invokes the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007) to train the constituent parsing models. Function PARSE generates  $k$ -best parse trees. Function TRANSFORMTRAIN invokes the perceptron algorithm (Collins, 2002) to train a discriminative annotation transformer. Function TRANSFORM selects the optimal transformed parse trees with the target annotation.

### 2.2 Learning the Annotation Transformer

To capture the transformation information from the source treebank to the target treebank, we use the discriminative reranking technique (Charniak and Johnson, 2005; Collins and Koo, 2005) to train the annotation transformer and to score  $k$ -best parse trees with some heterogeneous features.

In this paper, the averaged perceptron algorithm is used to train the treebank transformation model. It is an online training algorithm and has been successfully used in many NLP tasks, such as parsing (Collins and Roark, 2004) and word segmentation (Zhang and Clark, 2007; Zhang and Clark, 2010).

In addition to the target features which closely follow Sun et al. (2010). We design the following quasi-synchronous features to model the annotation inconsistencies.

- **Bigram constituent relation** For two consecutive fundamental constituents  $s_i$  and  $s_j$  in the target parse tree, we find the minimum categories  $N_i$  and  $N_j$  of the spans of  $s_i$  and  $s_j$  in the source parse tree respectively. Here



---

**Algorithm 1** Basic treebank annotation transformation.

---

```
1: function TRANSFORM-TRAIN(treebanks, treebankt)
2:   parsers ← TRAIN(treebanks)
3:   treebankts ← PARSE(parsers, treebankt)
4:   transformers→t ← TRANSFORMTRAIN(treebankt, treebankts)
5:   treebankst ← TRANSFORM(transformers→t, treebanks)
6:   return treebankst ∪ treebankt
```

---

---

**Algorithm 2** Iterative treebank annotation transformation.

---

```
1: function TRANSFORM-ITERTRAIN(treebanks, treebankt)
2:   parsers ← TRAIN(treebanks)
3:   parsert ← TRAIN(treebankt)
4:   treebankts ← PARSE(parsers, treebankt)
5:   treebankst ← PARSE(parsert, treebanks)
6:   repeat
7:     transformers→t ← TRANSFORMTRAIN(treebankt, treebankts)
8:     transformert→s ← TRANSFORMTRAIN(treebanks, treebankst)
9:     treebankst ← TRANSFORM(transformers→t, treebanks)
10:    treebankts ← TRANSFORM(transformert→s, treebankt)
11:    parsert ← TRAIN(treebankst ∪ treebankt)
12:  until EVAL(parsert) converges
13:  return treebankst ∪ treebankt
```

---

a fundamental constituent is defined to be a pair of word and its POS tag. If  $N_i$  is a sibling of  $N_j$  or each other is identical, we regard the relation between  $s_i$  and  $s_j$  as a positive feature.

- **Consistent relation** If the span of a target constituent can be also parsed as a constituent by the source parser, the combination of target rule and source category is used.
- **Inconsistent relation** If the span of a target constituent cannot be analysed as a constituent by the source parser, the combination of target rule and corresponding treelet in the source parse tree is used.
- **POS tag** The combination of POS tags of same words in the parallel data is used.

### 2.3 Iterative Training for Annotation Transformation

Treebank annotation transformation relies on the parallel training data. Consequently, the accuracy of source parser decides the accuracy of annotation transformer. We propose an iterative training method to improve the transformation accuracy by iteratively optimizing the parallel parse trees. At each iteration of training, the treebank transformation of source-to-target and target-to-source are both performed, and the transformed treebank provides more appropriate annotation for subsequent iteration. In turn, the annotation transformer can be improved gradually along with optimization of the parallel parse trees until convergence.

Algorithm 2 shows the overall procedure of iterative training, which terminates when the performance of a parser trained on the target treebank and the transformed treebank converges.

## 3 Experiments

### 3.1 Experimental Setup

We conduct the experiments of treebank transformation from TCT to CTB. CTB 5.1 is used as the target treebank. We follow the conventional corpus splitting of CTB 5.1: articles 001-270 and 400-1151 are used for training, articles 271-300 are used as test data and articles 301-325 are used as developing data. We use slightly modified version of CTB 5.1 by deleting all the function tags and empty categories, e.g., \*OP\*, using Tsurgon (Levy and Andrew, 2006). The whole TCT 1.0 is taken as the source treebank for training the annotation transformer.

The Berkeley parsing model is trained with 5 split-merge iterations. And we run the Berkeley parser in 100-best mode and construct the 20-fold cross validation training as described in Charniak and Johnson (2005). In this way, we acquire the parallel parse trees for training the annotation transformer.

In this paper, we use bracketing  $F1$  as the ParseVal metric provided by EVALB<sup>1</sup> for all experiments.

---

<sup>1</sup><http://nlp.cs.nyu.edu/evalb/>

| Model                               | F-Measure ( $\leq 40$ words) | F-Measure (all) |
|-------------------------------------|------------------------------|-----------------|
| Self-training                       | 86.11                        | 83.81           |
| Base Annotation Transformation      | 86.56                        | 84.23           |
| Iterative Annotation Transformation | 86.75                        | 84.37           |
| Baseline                            | 85.71                        | 83.42           |

Table 1: The performance of treebank annotation transformation using iterative training.

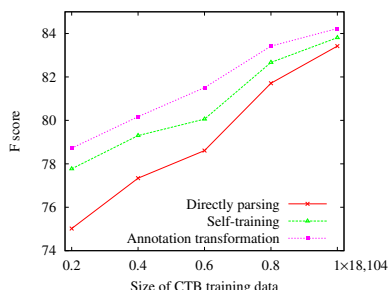


Figure 2: Parsing accuracy with different amounts of CTB training data.

### 3.2 Basic Transformation

We conduct experiments to evaluate the effect of the amount of target training data on transformation accuracy, and how much constituent parsers can benefit from our approach. An enhanced parser is trained on the CTB training data with the addition of transformed TCT by our annotation transformer. As comparison, we build a baseline system (direct parsing) using the Berkeley parser only trained on the CTB training data. In this experiment, the self-training method (McClosky et al., 2006a; McClosky et al., 2006b) is also used to build another strong baseline system, which uses unlabelled TCT as additional data. Figure 2 shows that our approach outperforms the two strong baseline systems. It achieves a 0.69% absolute improvement on the CTB test data over the direct parsing baseline when the whole CTB training data is used for training. We also can find that our approach further extends the advantage over the two baseline systems as the amount of CTB training data decreases in Figure 2. The figure confirms our approach is effective for improving parser performance, specially for the scenario where the target treebank is scarce.

### 3.3 Iterative Transformation

We use the iterative training method for annotation transformation. The CTB developing set is used to determine the optimal training iteration. After each iteration, we test the performance of a parser trained on the combined treebank. Fig-

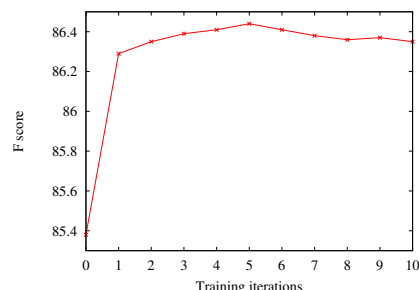


Figure 3: Learning curve of iterative transformation training.

ure 3 shows the performance curve with iteration ranging from 1 to 10. The performance of basic annotation transformation is also included in the curve when iteration is 1. The curve shows that the maximum performance is achieved at iteration 5. Compared to the basic annotation transformation, the iterative training strategy leads to a better parser with higher accuracy. Table 1 reports that the final optimized parsing results on the CTB test set contributes a 0.95% absolute improvement over the directly parsing baseline.

## 4 Related Work

Treebank transformation is an effective strategy to reuse existing annotated data. Wang et al. (1994) proposed an approach to transform a treebank into another with a different grammar using their matching metric based on the bracket information of original treebank. Jiang et al. (2009) proposed annotation adaptation in Chinese word segmentation, then, some work were done in parsing (Sun et al., 2010; Zhu et al., 2011; Sun and Wan, 2012). Recently, Jiang et al. (2012) proposed an advanced annotation transformation in Chinese word segmentation, and we extended it to the more complicated treebank annotation transformation used for Chinese constituent parsing.

Other related work has been focused on semi-supervised parsing methods which utilize labeled data to annotate unlabeled data, then use the additional annotated data to improve the original model (McClosky et al., 2006a; McClosky et

al., 2006b; Huang and Harper, 2009). The self-training methodology enlightens us on getting annotated treebank compatible with another annotation guideline. Our approach places extra emphasis on improving the transformation performance with the help of source annotation knowledge.

Apart from constituency-to-constituency treebank transformation, there also exists some research on dependency-to-constituency treebank transformation. Collins et al. (1999) used transformed constituency treebank from Prague Dependency Treebank for constituent parsing on Czech. Xia and Palmer (2001) explored different algorithms that transform dependency structure to phrase structure. Niu et al. (2009) proposed to convert a dependency treebank to a constituency one by using a parser trained on a constituency treebank to generate  $k$ -best lists for sentences in the dependency treebank. Optimal conversion results are selected from the  $k$ -best lists. Smith and Eisner (2009) and Li et al. (2012) generated rich quasi-synchronous grammar features to improve parsing performance. Some work has been done from the other direction (Daum et al., 2004; Nivre, 2006; Johansson and Nugues, 2007).

## 5 Conclusion

This paper propose an effective approach to transform one treebank into another with a different annotation guideline. Experiments show that our approach can effectively utilize the heterogeneous treebanks and significantly improve the state-of-the-art Chinese constituency parsing performance. How to exploit more heterogeneous knowledge to improve the transformation performance is an interesting future issue.

## Acknowledgments

The authors were supported by National Natural Science Foundation of China (Contracts 61202216), National Key Technology R&D Program (No. 2012BAH39B03), and Key Project of Knowledge Innovation Program of Chinese Academy of Sciences (No. KGZD-EW-501). Qun Liu's work was partially supported by Science Foundation Ireland (Grant No.07/CE/I1142) as part of the CNGL at Dublin City University. Sincere thanks to the three anonymous reviewers for their thorough reviewing and valuable suggestions!

## References

- E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL*, pages 173–180.
- M. Collins and T. Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.
- M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*, volume 2004.
- M. Collins, L. Ramshaw, J. Hajič, and C. Tillmann. 1999. A statistical parser for czech. In *Proceedings of ACL*, pages 505–512.
- M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8.
- M. Daum, K. Foth, and W. Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proceedings of LREC*.
- Z. Huang and M. Harper. 2009. Self-training pcfg grammars with latent annotations across languages. In *Proceedings of EMNLP*, pages 832–841.
- W. Jiang, L. Huang, and Q. Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging: a case study. In *Proceedings of ACL*, pages 522–530.
- Wenbin Jiang, Fandong Meng, Qun Liu, and Yajuan Lü. 2012. Iterative annotation transformation with predict-self reestimation for chinese word segmentation. In *Proceedings of EMNLP*, pages 412–420.
- R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proc. of the 16th Nordic Conference on Computational Linguistics*.
- R. Levy and G. Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234.
- Zhengkua Li, Ting Liu, and Wanxiang Che. 2012. Exploiting multiple treebanks for parsing with quasi-synchronous grammars. In *Proceedings of ACL*, pages 675–684.
- D. McClosky, E. Charniak, and M. Johnson. 2006a. Effective self-training for parsing. In *Proceedings of NAACL*, pages 152–159.
- D. McClosky, E. Charniak, and M. Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of ACL*, pages 337–344.
- Zheng-Yu Niu, Haifeng Wang, and Hua Wu. 2009. Exploiting heterogeneous treebanks for parsing. In *Proceedings of ACL*, pages 46–54.

- J. Nivre. 2006. *Inductive dependency parsing*. Springer Verlag.
- S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL*, pages 404–411.
- S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL*, pages 433–440.
- David A Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*, pages 822–831.
- W. Sun and X. Wan. 2012. Reducing approximation and estimation errors for chinese lexical processing with heterogeneous annotations. In *Proceedings of ACL*.
- W. Sun, R. Wang, and Y. Zhang. 2010. Discriminative parse reranking for chinese with homogeneous and heterogeneous annotations. In *Proceedings of CIPS-SIGHAN*.
- J.N. Wang, J.S. Chang, and K.Y. Su. 1994. An automatic treebank conversion algorithm for corpus sharing. In *Proceedings of ACL*, pages 248–254.
- F. Xia and M. Palmer. 2001. Converting dependency structures to phrase structures. In *Proceedings of the first international conference on Human language technology research*, pages 1–5.
- N. Xue, F. Xia, F.D. Chiou, and M. Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(02):207–238.
- S. Yu, J. Lu, X. Zhu, H. Duan, S. Kang, H. Sun, H. Wang, Q. Zhao, and W. Zhan. 2001. Processing norms of modern chinese corpus. *Technical Report*.
- Y. Zhang and S. Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of ACL*, pages 840–847.
- Y. Zhang and S. Clark. 2010. A fast decoder for joint word segmentation and pos-tagging using a single discriminative model. In *Proceedings of EMNLP*, pages 843–852.
- Q. Zhou. 2004. Annotation scheme for chinese treebank. *Journal of Chinese Information Processing*, 18(4).
- M. Zhu, J. Zhu, and M. Hu. 2011. Better automatic treebank conversion using a feature-based approach. In *Proceedings of ACL*, pages 715–719.

# Nonparametric Bayesian Inference and Efficient Parsing for Tree-adjointing Grammars

Elif Yamangil and Stuart M. Shieber

Harvard University

Cambridge, Massachusetts, USA

{elif,shieber}@seas.harvard.edu

## Abstract

In the line of research extending statistical parsing to more expressive grammar formalisms, we demonstrate for the first time the use of tree-adjointing grammars (TAG). We present a Bayesian non-parametric model for estimating a probabilistic TAG from a parsed corpus, along with novel block sampling methods and approximation transformations for TAG that allow efficient parsing. Our work shows performance improvements on the Penn Treebank and finds more compact yet linguistically rich representations of the data, but more importantly provides techniques in grammar transformation and statistical inference that make practical the use of these more expressive systems, thereby enabling further experimentation along these lines.

## 1 Introduction

There is a deep tension in statistical modeling of grammatical structure between providing good expressivity — to allow accurate modeling of the data with sparse grammars — and low complexity — making induction of the grammars (say, from a treebank) and parsing of novel sentences computationally practical. Tree-substitution grammars (TSG), by expanding the domain of locality of context-free grammars (CFG), can achieve better expressivity, and the ability to model more contextual dependencies; the payoff would be better modeling of the data or smaller (parser) models or both. For instance, constructions that go across levels, like the predicate-argument structure of a verb and its arguments can be modeled by TSGs (Goodman, 2003).

Recent work that incorporated Dirichlet process (DP) nonparametric models into TSGs has provided an efficient solution to the daunting model selection problem of segmenting training data trees into appropriate elementary fragments to form the grammar (Cohn et al., 2009; Post and Gildea, 2009). The elementary trees combined in a TSG are, intuitively, primitives of the language, yet certain linguistic phenomena (notably various forms of modification) “split them up”, preventing their reuse, leading to less sparse grammars than might be ideal (Yamangil and Shieber, 2012; Chiang, 2000; Resnik, 1992).

TSGs are a special case of the more flexible grammar formalism of tree adjointing grammar (TAG) (Joshi et al., 1975). TAG augments TSG with an *adjunction operator* and a set of *auxiliary trees* in addition to the substitution operator and initial trees of TSG, allowing for “splicing in” of syntactic fragments within trees. This functionality allows for better modeling of linguistic phenomena such as the distinction between modifiers and arguments (Joshi et al., 1975; XTAG Research Group, 2001). Unfortunately, TAG’s expressivity comes at the cost of greatly increased complexity. Parsing complexity for unconstrained TAG scales as  $O(n^6)$ , impractical as compared to CFG and TSG’s  $O(n^3)$ . In addition, the model selection problem for TAG is significantly more complicated than for TSG since one must reason about many more combinatorial options with two types of derivation operators. This has led researchers to resort to manual (Doran et al., 1997) or heuristic techniques. For example, one can consider “outsourcing” the auxiliary trees (Shieber, 2007), use template rules and a very small number of grammar categories (Hwa, 1998), or rely on head-words and force lexicalization in order to constrain the problem (Xia et al., 2001; Chiang,

2000; Carreras et al., 2008). However a solution has not been put forward by which a model that maximizes a principled probabilistic objective is sought after.

Recent work by Cohn and Blunsom (2010) argued that under highly expressive grammars such as TSGs where exponentially many derivations may be hypothesized of the data, local Gibbs sampling is insufficient for effective inference and global blocked sampling strategies will be necessary. For TAG, this problem is only more severe due to its mild context-sensitivity and even richer combinatorial nature. Therefore in previous work, Shindo et al. (2011) and Yamangil and Shieber (2012) used tree-insertion grammar (TIG) as a kind of expressive compromise between TSG and TAG, as a substrate on which to build nonparametric inference. However TIG has the constraint of disallowing wrapping adjunction (coordination between material that falls to the left and right of the point of adjunction, such as parentheticals and quotations) as well as left adjunction along the spine of a right auxiliary tree and vice versa.

In this work we formulate a blocked sampling strategy for TAG that is effective and efficient, and prove its superiority against the local Gibbs sampling approach. We show via nonparametric inference that TAG, which contains TSG as a subset, is a better model for treebank data than TSG and leads to improved parsing performance. TAG achieves this by using more compact grammars than TSG and by providing the ability to make finer-grained linguistic distinctions. We explain how our parameter refinement scheme for TAG allows for cubic-time CFG parsing, which is just as efficient as TSG parsing. Our presentation assumes familiarity with prior work on block sampling of TSG and TIG (Cohn and Blunsom, 2010; Shindo et al., 2011; Yamangil and Shieber, 2012).

## 2 Probabilistic Model

In the basic nonparametric TSG model, there is an independent DP for every grammar category (such as  $c = \text{NP}$ ), each of which uses a base distribution  $P_0$  that generates an initial tree by making stepwise decisions and concentration parameter  $\alpha_c$  that controls the level of sparsity (size) of the generated grammars:  $G_c \sim \text{DP}(\alpha_c, P_0(\cdot | c))$ . We extend this model by adding specialized DPs for auxiliary trees  $G_c^{\text{aux}} \sim \text{DP}(\alpha_c^{\text{aux}}, P_0^{\text{aux}}(\cdot | c))$ . Therefore, we have an exchangeable process for generating auxiliary tree  $a_j$  given  $j - 1$  auxiliary

trees previously generated

$$p(a_j | \mathbf{a}_{<j}) = \frac{n_{c,a_j} + \alpha_c^{\text{aux}} P_0^{\text{aux}}(a_j | c)}{j - 1 + \alpha_c^{\text{aux}}} \quad (1)$$

as for initial trees in TSG (Cohn et al., 2009).

We must define base distributions for initial trees and auxiliary trees.  $P_0$  generates an initial tree with root label  $c$  by sampling rules from a CFG  $\tilde{P}$  and making a binary decision at every node generated whether to leave it as a frontier node or further expand (with probability  $\beta_c$ ) (Cohn et al., 2009). Similarly, our  $P_0^{\text{aux}}$  generates an auxiliary tree with root label  $c$  by sampling a CFG rule from  $\tilde{P}$ , flipping an unbiased coin to decide the direction of the spine (if more than a unique child was generated), making a binary decision at the spine whether to leave it as a foot node or further expand (with probability  $\gamma_c$ ), and recurring into  $P_0$  or  $P_0^{\text{aux}}$  appropriately for the off-spine and spinal children respectively.

We glue these two processes together via a set of adjunction parameters  $\mu_c$ . In any derivation for every node labeled  $c$  that is not a frontier node or the root or foot node of an auxiliary tree, we determine the number (perhaps zero) of *simultaneous adjunctions* (Schabes and Shieber, 1994) by sampling a Geometric( $\mu_c$ ) variable; thus  $k$  simultaneous adjunctions would have probability  $(\mu_c)^k (1 - \mu_c)$ . Since we already provide simultaneous adjunction we disallow adjunction at the root of auxiliary trees.

## 3 Inference

Given this model, our inference task is to explore posterior derivations underlying the data. Since TAG derivations are highly structured objects, we design a blocked Metropolis-Hastings sampler that samples derivations per entire parse trees all at once in a joint fashion (Cohn and Blunsom, 2010; Shindo et al., 2011; Yamangil and Shieber, 2012). As in previous work, we use a Goodman-transformed TAG as our proposal distribution (Goodman, 2003) that incorporates additional CFG rules to account for the possibility of backing off to the infinite base distribution  $P_0^{\text{aux}}$ , and use the parsing algorithm described by Shieber et al. (1995) for computing inside probabilities under this TAG model.

The algorithm is illustrated in Table 1 along with Figure 1. Inside probabilities are computed in a bottom-up fashion and a TAG derivation is sampled top-down (Johnson et al., 2007). The

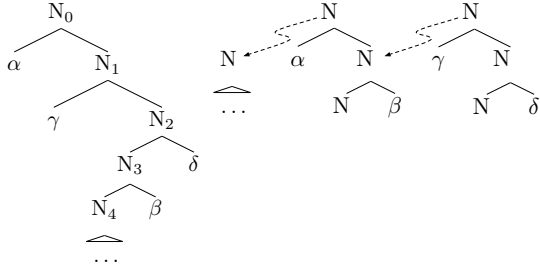


Figure 1: Example used for illustrating blocked sampling with TAG. On the left hand side we have a partial training tree where we highlight the particular nodes (with node labels 0, 1, 2, 3, 4) that the sampling algorithm traverses in post-order. On the right hand side is the TAG grammar fragment that is used to parse these particular nodes: one initial tree and two *wrapping* auxiliary trees where one adjoins into the *spine* of the other for full generality of our illustration. Grammar nodes are labeled with their Goodman indices (letters  $i, j, k, l, m$ ). Greek letters  $\alpha, \beta, \gamma, \delta$  denote entire subtrees. We assume that a subtree in an auxiliary tree (e.g.,  $\alpha$ ) parses the same subtree in a training tree.

sampler visits every node of the tree in post-order ( $O(n)$  operations,  $n$  being the number of nodes), visits every node below it as a potential foot (another  $O(n)$  operations), visits every mid-node in the path between the original node and the potential foot (if spine-adjunction is allowed) ( $O(\log n)$  operations), and forms the appropriate chart items. The complexity is  $O(n^2 \log n)$  if spine-adjunction is allowed,  $O(n^2)$  otherwise.

#### 4 Parameter Refinement

During inference, adjunction probabilities are treated simplistically to facilitate convergence. Only two parameters guide adjunction:  $\mu_c$ , the probability of adjunction; and  $p(a_j \mid \mathbf{a}_{<j}, c)$  (see Equation 1), the probability of the particular auxiliary tree being adjoined given that there is an adjunction. In all of this treatment,  $c$ , the context of an adjunction, is the grammar category label such as S or NP, instead of a unique identifier for the node at which the adjunction occurs as was originally the case in probabilistic TAG literature. However it is possible to experiment with further refinement schemes at parsing time. Once the sampler converges on a grammar, we can re-estimate its adjunction probabilities. Using the  $O(n^6)$  parsing algorithm (Shieber et al., 1995) we experimented with various refinements schemes — ranging from full node identifiers, to Goodman

| Chart item     | Why made?                     | Inside probability   |
|----------------|-------------------------------|--|
| $N_i[4]$       | By assumption.                | —  |
| $N_k[3-4]$     | $N_*[4]$ and $\beta$          | $(1 - \mu_c) \times \pi(\beta)$                                |
| $N_m[2-3]$     | $N_*[3]$ and $\delta$         | $(1 - \mu_c) \times \pi(\delta)$                               |
| $N_l[1-3]$     | $\gamma$ and $N_m[2-3]$       | $(1 - \mu_c) \times \pi(\gamma)$<br>$\times \pi(N_m[2-3])$     |
| $N_{aux}[1-3]$ | $N_l[1-3]$                    | $n_{c,a_l} / (n_c + \alpha_c^{aux})$<br>$\times \pi(N_l[1-3])$ |
| $N_k[1-4]$     | $N_{aux}[1-3]$ and $N_k[3-4]$ | $\mu_c \times \pi(N_{aux}[1-3])$<br>$\times \pi(N_k[3-4])$     |
| $N_j[0-4]$     | $\alpha$ and $N_k[1-4]$       | $(1 - \mu_c) \times \pi(\alpha)$<br>$\times \pi(N_k[1-4])$     |
| $N_{aux}[0-4]$ | $N_j[0-4]$                    | $n_{c,a_j} / (n_c + \alpha_c^{aux})$<br>$\times \pi(N_j[0-4])$ |
| $N_i[0]$       | $N_{aux}[0-4]$ and $N_i[4]$   | $\mu_c \times \pi(N_{aux}[0-4])$<br>$\times \pi(N_i[4])$       |

Table 1: Computation of inside probabilities for TAG sampling. We create two types of chart items: (1) **per-node**, e.g.,  $N_i[\nu]$  denoting the probability of starting at an initial subtree that has Goodman index  $i$  and generating the subtree rooted at node  $\nu$ , and (2) **per-path**, e.g.,  $N_j[\nu-\eta]$  denoting the probability of starting at an auxiliary subtree that has Goodman index  $j$  and generating the subtree rooted at  $\nu$  minus the subtree rooted at  $\eta$ . Above,  $c$  denotes the context of adjunction, which is the nonterminal label of the node of adjunction (here, N),  $\mu_c$  is the probability of adjunction,  $n_{c,a}$  is the count of the auxiliary tree  $a$ , and  $n_c = \sum_a n_{c,a}$  is total number of adjunctions at context  $c$ . The function  $\pi(\cdot)$  retrieves the inside probability corresponding to an item.

index identifiers of the subtree below the adjunction (Hwa, 1998), to simple grammar category labels — and find that using Goodman index identifiers as  $c$  is the best performing option.

Interestingly, this particular refinement scheme also allows for fast cubic-time parsing, which we achieve by approximating the TAG by a TSG with little loss of coverage (*no* loss of coverage under special conditions which we find that are often satisfied) and negligible increase in grammar size, as discussed in the next section.

#### 5 Cubic-time parsing

MCMC training results in a list of sufficient statistics of the final derivation that the TAG sampler converges upon after a number of iterations. Basically, these are the list of initial and auxiliary trees, their cumulative counts over the training data, and their adjunction statistics. An adjunction statistic is listed as follows. If  $\alpha$  is any elementary tree, and  $\beta$  is an auxiliary tree that adjoins  $n$  times at node  $\nu$  of  $\alpha$  that is uniquely reachable at path  $p$ , we write  $\alpha \stackrel{p}{\leftarrow} \beta$  ( $n$  times). We denote  $\nu$  alternatively as  $\alpha[p]$ .

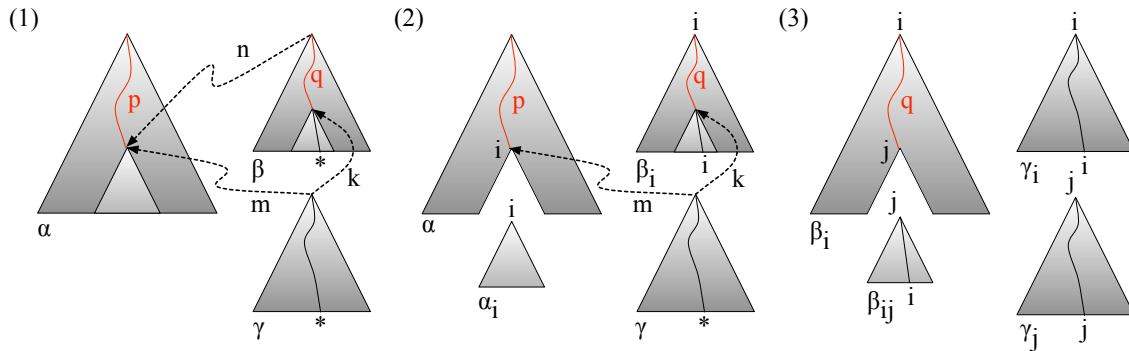


Figure 2: TAG to TSG transformation algorithm. By removing adjunctions in the correct order we end up with a larger yet adjunction-free TSG.

Now imagine that we end up with a small grammar that consists of one initial tree  $\alpha$  and two auxiliary trees  $\beta$  and  $\gamma$ , and the following adjunctions occurring between them

$$\begin{aligned} \alpha &\stackrel{p}{\leftarrow} \beta \text{ (} n \text{ times)} \\ \alpha &\stackrel{m}{\leftarrow} \gamma \text{ (} m \text{ times)} \\ \beta &\stackrel{q}{\leftarrow} \gamma \text{ (} k \text{ times)} \end{aligned}$$

as shown in Figure 2. Assume that  $\alpha$  itself occurs  $l > n + m$  times in total so that there is nonzero probability of no adjunction anywhere within  $\alpha$ . Also assume that the node uniquely identified by  $\alpha[p]$  has Goodman index  $i$ , which we denote as  $i = G(\alpha[p])$ .

The general idea of this TAG-TSG approximation is that, for any auxiliary tree that adjoins at a node  $\nu$  with Goodman index  $i$ , we create an initial tree out of it where the root and foot nodes of the auxiliary tree are both replaced by  $i$ . Further, we split the subtree rooted at  $\nu$  from its parent and rename the substitution site that is newly created at  $\nu$  as  $i$  as well. (See Figure 2.) We can separate the foot subtree from the rest of the initial tree since it is completely remembered by any adjoined auxiliary trees due to the nature of our refinement scheme. However this method fails for adjunctions that occur at *spinal* nodes of auxiliary trees that have foot nodes below them since we would not know in which order to do the initial tree creation. However when the *spine-adjunction relation* is amenable to a *topological sort* (as is the case in Figure 2), we can apply the method by going in this order and doing some extra bookkeeping: updating the list of Goodman indices and redirecting adjunctions as we go along. When there is no such topological sort, we can approximate the TAG by heuristically dropping low-frequency

adjunctions that introduce cycles.<sup>1</sup>

The algorithm is illustrated in Figure 2. In (1) we see the original TAG grammar and its adjunctions ( $n, m, k$  are adjunction counts). Note that the adjunction relation has a topological sort of  $\alpha, \beta, \gamma$ . We process auxiliary trees in this order and iteratively remove their adjunctions by creating specialized initial tree duplicates. In (2) we first visit  $\beta$ , which has adjunctions into  $\alpha$  at the node denoted  $\alpha[p]$  where  $p$  is the unique path from the root to this node. We retrieve the Goodman index of this node  $i = G(\alpha[p])$ , split the subtree rooted at this node as a new initial tree  $\alpha_i$ , relabel its root as  $i$ , and rename the newly-created substitution site at  $\alpha[p]$  as  $i$ . Since  $\beta$  has only this adjunction, we replace it with initial tree version  $\beta_i$  where root/foot labels of  $\beta$  are replaced with  $i$ , and update all adjunctions into  $\beta$  as being into  $\beta_i$ . In (3) we visit  $\gamma$  which now has adjunctions into  $\alpha$  and  $\beta_i$ . For the  $\alpha[p]$  adjunction we create  $\gamma_i$  the same way we created  $\beta_i$  but this time we cannot remove  $\gamma$  as it still has an adjunction into  $\beta_i$ . We retrieve the Goodman index of the node of adjunction  $j = G(\beta_i[q])$ , split the subtree rooted at this node as new initial tree  $\beta_{ij}$ , relabel its root as  $j$ , and rename the newly-created substitution site at  $\beta_i[q]$  as  $j$ . Since  $\gamma$  now has only this adjunction left, we remove it by also creating initial tree version  $\gamma_j$  where root/foot labels of  $\gamma$  are replaced with  $j$ . At this point we have an adjunction-free TSG with elementary trees (and counts)  $\alpha(l), \alpha_i(l), \beta_i(n), \beta_{ij}(n), \gamma_i(m), \gamma_j(k)$  where  $l$  is the count of initial tree  $\alpha$ . These counts, when they are normalized, lead to the appropriate adjunc-

<sup>1</sup>We found that, on average, about half of our grammars have a topological sort of their spine-adjunctions. (On average fewer than 100 spine adjunctions even exist.) When no such sort exists, only a few low-frequency adjunctions have to be removed to eliminate cycles.



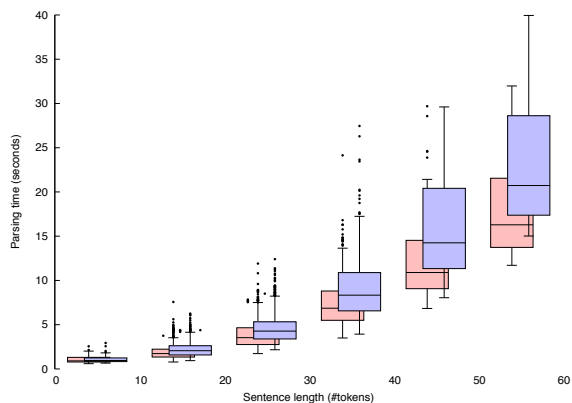


Figure 3: Nonparametric TAG (blue) parsing is efficient and incurs only a small increase in parsing time compared to nonparametric TSG (red).

tion probability refinement scheme of  $\mu_c \times p(a_j | \mathbf{a}_{<j}, c)$  where  $c$  is the Goodman index.

Although this algorithm increases grammar size, the sparsity of the nonparametric solution ensures that the increase is almost negligible: on average the final Goodman-transformed CFG has 173.9K rules for TSG, 189.2K for TAG. Figure 3 demonstrates the comparable Viterbi parsing times for TSG and TAG.

## 6 Evaluation

We use the standard Penn treebank methodology of training on sections 2–21 and testing on section 23. All our data is head-binarized, all hyperparameters are resampled under appropriate vague gamma and beta priors. Samplers are run 1000 iterations each; all reported numbers are averages over 5 runs. For simplicity, parsing results are based on the maximum probability derivation (Viterbi algorithm).

In Table 4, we compare TAG inference schemes and TSG. TAG<sub>Gibbs</sub> operates by locally adding/removing potential adjunctions, similar to Cohn et al. (2009). TAG' is the  $O(n^2)$  algorithm that disallows spine adjunction. We see that TAG' has the best parsing performance, while TAG provides the most compact representation.

| model                | F measure    | # initial trees | # auxiliary trees |
|----------------------|--------------|-----------------|-------------------|
| TSG                  | 84.15        | 69.5K           | -                 |
| TAG <sub>Gibbs</sub> | 82.47        | 69.9K           | 1.7K              |
| TAG'                 | <b>84.87</b> | 66.4K           | 1.5K              |
| TAG                  | 84.82        | <b>66.4K</b>    | <b>1.4K</b>       |

Figure 4: EVALB results. Note that the Gibbs sampler for TAG has poor performance and provides no grammar compaction due to its lack of convergence.

| label            | #adj (spine adj) | ave. depth | #lex. trees | #left trees | #right trees | #wrap trees |
|------------------|------------------|------------|-------------|-------------|--------------|-------------|
| $\overline{VP}$  | 4532 (23)        | 1.06       | 45          | 22          | 65           | 0           |
| $\overline{NP}$  | 2891 (46)        | 1.71       | 68          | 94          | 13           | 1           |
| $\overline{NN}$  | 2160 (3)         | 1.08       | 85          | 16          | 110          | 0           |
| $\overline{NNP}$ | 1478 (2)         | 1.12       | 90          | 19          | 90           | 0           |
| $\overline{NNS}$ | 1217 (1)         | 1.10       | 43          | 9           | 60           | 0           |
| $\overline{VBN}$ | 1121 (1)         | 1.05       | 6           | 18          | 0            | 0           |
| $\overline{VBD}$ | 976 (0)          | 1.0        | 16          | 25          | 0            | 0           |
| $\overline{NP}$  | 937 (0)          | 3.0        | 1           | 5           | 0            | 0           |
| $\overline{VB}$  | 870 (0)          | 1.02       | 14          | 31          | 4            | 0           |
| $\overline{S}$   | 823 (11)         | 1.48       | 42          | 36          | 35           | 3           |
| total            | 23320 (118)      | 1.25       | 824         | 743         | 683          | 9           |

Table 2: Grammar analysis for an estimated TAG, categorized by label. Only the most common top 10 are shown, binarization variables are denoted with overline. A total number of 98 wrapping adjunctions (9 unique wrapping trees) and 118 spine adjunctions occur.

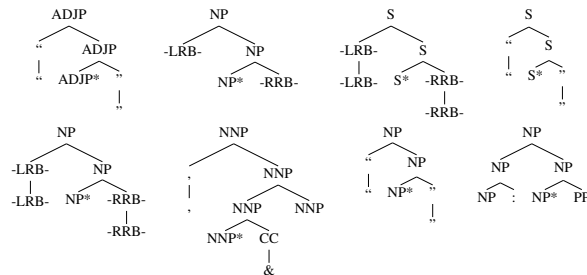


Figure 5: Example wrapping trees from estimated TAGs.

## 7 Conclusion

We described a nonparametric Bayesian inference scheme for estimating TAG grammars and showed the power of TAG formalism over TSG for returning rich, generalizable, yet compact representations of data. The nonparametric inference scheme presents a principled way of addressing the difficult model selection problem with TAG. Our sampler has near quadratic-time efficiency, and our parsing approach remains context-free allowing for fast cubic-time parsing, so that our overall parsing framework is highly scalable.<sup>2</sup>

There are a number of extensions of this work: Experimenting with automatically induced adjunction refinements as well as incorporating substitution refinements can benefit Bayesian TAG (Shindo et al., 2012; Petrov et al., 2006). We are also planning to investigate TAG for more context-sensitive languages, and synchronous TAG for machine translation.

<sup>2</sup>An extensive report of our algorithms and experiments will be provided in the PhD thesis of the first author (Yamangil, 2013). Our code will be made publicly available at [code.seas.harvard.edu/~elif](http://code.seas.harvard.edu/~elif).

## References

- Xavier Carreras, Michael Collins, and Terry Koo. 2008. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning, CoNLL '08*, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00*, pages 456–463, Morristown, NJ, USA. Association for Computational Linguistics.
- Trevor Cohn and Phil Blunsom. 2010. Blocked inference in Bayesian tree substitution grammars. In *Proceedings of the ACL 2010 Conference Short Papers, ACLShort '10*, pages 225–230, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 548–556, Morristown, NJ, USA. Association for Computational Linguistics.
- Christine Doran, Beth Hockey, Philip Hopely, Joseph Rosenzweig, Anoop Sarkar, B. Srinivas, Fei Xia, Alexis Nasr, and Owen Rambow. 1997. Maintaining the forest and burning out the underbrush in xtag. In *Proceedings of the ENVGRAM Workshop*.
- Joshua Goodman. 2003. Efficient parsing of DOP with PCFG-reductions. In Rens Bod, Remko Scha, and Khalil Sima'an, editors, *Data-Oriented Parsing*. CSLI Publications, Stanford, CA.
- Rebecca Hwa. 1998. An empirical evaluation of probabilistic lexicalized tree insertion grammars. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, pages 557–563, Morristown, NJ, USA. Association for Computational Linguistics.
- Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York, April. Association for Computational Linguistics.
- Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163.
- 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.
- Matt Post and Daniel Gildea. 2009. Bayesian learning of a tree substitution grammar. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 45–48, Suntec, Singapore, August. Association for Computational Linguistics.
- Philip Resnik. 1992. Probabilistic tree-adjoining grammar as a framework for statistical natural language processing. In *Proceedings of the 14th conference on Computational linguistics - Volume 2, COLING '92*, pages 418–424, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yves Schabes and Stuart M. Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124. Also available as cmp-1g/9404001.
- Stuart M. Shieber, Yves Schabes, and Fernando C. N. Pereira. 1995. Principles and implementation of deductive parsing. *J. Log. Program.*, 24(1&2):3–36.
- Stuart M. Shieber. 2007. Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries. In Dekai Wu and David Chiang, editors, *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, Rochester, New York, 26 April.
- Hiroyuki Shindo, Akinori Fujino, and Masaaki Nagata. 2011. Insertion operator for Bayesian tree substitution grammars. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages 206–211, Stroudsburg, PA, USA. 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 the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–448, Jeju Island, Korea, July. Association for Computational Linguistics.
- Fei Xia, Chung-hye Han, Martha Palmer, and Aravind Joshi. 2001. Automatically extracting and comparing lexicalized grammars for different languages. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2, IJCAI'01*, pages 1321–1326, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- XTAG Research Group. 2001. A lexicalized tree adjoining grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania.

Elif Yamangil and Stuart Shieber. 2012. Estimating compact yet rich tree insertion grammars. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 110–114, Jeju Island, Korea, July. Association for Computational Linguistics.

Elif Yamangil. 2013. *Rich Linguistic Structure from Large-Scale Web Data*. Ph.D. thesis, Harvard University. Forthcoming.

# Using CCG categories to improve Hindi dependency parsing

**Bharat Ram Ambati**

**Tejaswini Deoskar**

**Mark Steedman**

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

bharat.ambati@ed.ac.uk, {tdeoskar, steedman}@inf.ed.ac.uk

## Abstract

We show that informative lexical categories from a strongly lexicalised formalism such as Combinatory Categorical Grammar (CCG) can improve dependency parsing of Hindi, a free word order language. We first describe a novel way to obtain a CCG lexicon and treebank from an existing dependency treebank, using a CCG parser. We use the output of a supertagger trained on the CCGbank as a feature for a state-of-the-art Hindi dependency parser (Malt). Our results show that using CCG categories improves the accuracy of Malt on long distance dependencies, for which it is known to have weak rates of recovery.

## 1 Introduction

As compared to English, many Indian languages including Hindi have a freer word order and are also morphologically richer. These characteristics pose challenges to statistical parsers. Today, the best dependency parsing accuracies for Hindi are obtained by the shift-reduce parser of Nivre et al. (2007) (Malt). It has been observed that Malt is relatively accurate at recovering short distance dependencies, like arguments of a verb, but is less accurate at recovering long distance dependencies like co-ordination, root of the sentence, etc (McDonald and Nivre, 2007; Ambati et al., 2010).

In this work, we show that using CCG lexical categories (Steedman, 2000), which contain sub-categorization information and capture long distance dependencies elegantly, can help Malt with those dependencies. Section 2 first shows how we extract a CCG lexicon from an existing Hindi dependency treebank (Bhatt et al., 2009) and then use it to create a Hindi CCGbank. In section 3, we develop a supertagger using the CCGbank and explore different ways of providing CCG categories

from the supertagger as features to Malt. Our results show that using CCG categories can help Malt by improving the recovery of long distance relations.

## 2 A CCG Treebank from a Dependency Treebank

There have been some efforts at automatically extracting treebanks of CCG derivations from phrase structure treebanks (Hockenmaier and Steedman, 2007; Hockenmaier, 2006; Tse and Curran, 2010), and CCG lexicons from dependency treebanks (Çakıcı, 2005). Bos et al. (2009) created a CCGbank from an Italian dependency treebank by converting dependency trees into phrase structure trees and then applying an algorithm similar to Hockenmaier and Steedman (2007). In this work, following Çakıcı (2005), we first extract a Hindi CCG lexicon from a dependency treebank. We then use a CKY parser based on the CCG formalism to automatically obtain a treebank of CCG derivations from this lexicon, a novel methodology that may be applicable to obtaining CCG treebanks in other languages as well.

### 2.1 Hindi Dependency Treebank

In this paper, we work with a subset of the Hindi Dependency Treebank (HDT ver-0.5) released as part of Coling 2012 Shared Task on parsing (Bharati et al., 2012). HDT is a multi-layered dependency treebank (Bhatt et al., 2009) annotated with morpho-syntactic (morphological, part-of-speech and chunk information) and syntactico-semantic (dependency) information (Bharati et al., 2006; Bharati et al., 2009). Dependency labels are fine-grained, and mark dependencies that are syntactico-semantic in nature, such as agent (usually corresponding to subject), patient (object), and time and place expressions. There are special labels to mark long distance relations like relative clauses, co-ordination etc

(Bharati et al., 1995; Bharati et al., 2009).

The treebank contains 12,041 training, 1,233 development and 1,828 testing sentences with an average of 22 words per sentence. We used the CoNLL format<sup>1</sup> for our purposes, which contains word, lemma, pos-tag, and coarse pos-tag in the WORD, LEMMA, POS, and CPOS fields respectively and morphological features and chunk information in the FEATS column.

## 2.2 Algorithm

We first made a list of argument and adjunct dependency labels in the treebank. For e.g., dependencies with the label *k1* and *k2* (corresponding to subject and object respectively) are considered to be arguments, while labels like *k7p* and *k7t* (corresponding to place and time expressions) are considered to be adjuncts. For readability reasons, we will henceforth refer to dependency labels with their English equivalents (e.g., SUBJ, OBJ, PURPOSE, CASE for *k1*, *k2*, *rt*, *lwg\_psp* respectively).

Starting from the root of the dependency tree, we traverse each node. The category of a node depends on both its parent and children. If the node is an argument of its parent, we assign the chunk tag of the node (e.g., NP, PP) as its CCG category. Otherwise, we assign it a category of  $X|X$ , where  $X$  is the parent's *result* category and  $|$  is directionality ( $\backslash$  or  $/$ ), which depends on the position of the node w.r.t. its parent. The *result* category of a node is the category obtained once its arguments are resolved. For example,  $S$ , is the result category for  $(S\backslash NP)\backslash NP$ . Once we get the partial category of a node based on the node's parent information, we traverse through the children of the node. If a child is an argument, we add that child's chunk tag, with appropriate directionality, to the node's category. The algorithm is sketched in Figure 1 and an example of a CCG derivation for a simple sentence (marked with chunk tags; NP and VGF are the chunk tags for noun and finite verb chunks respectively.) is shown in Figure 2. Details of some special cases are described in the following subsections.

We created two types of lexicon. In Type 1, we keep morphological information in noun categories and in Type 2, we don't. For example, consider a noun chunk 'raam ne (Ram ERG)'. In Type 1, CCG categories for 'raam' and 'ne' are NP and

```

ModifyTree(DependencyTree tree);
for (each node in tree):
  handlePostPositionMarkers(node);
  handleCoordination(node);
  handleRelativeClauses(node);
  if (node is an argument of parent):
    cat = node.chunkTag;
  else:
    prescat = parent.resultCategory;
    cat = prescat + getDir(node, parent) + prescat;
  for(each child of node):
    if (child is an argument of node):
      cat = cat + getDir(child, node) + child.chunkTag;

```

Figure 1: Algorithm for extracting CCG lexicon from a dependency tree.

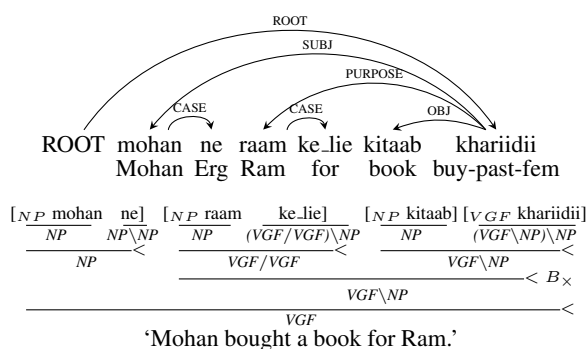


Figure 2: An example dependency tree with its CCG derivation (Erg = Ergative case).

NP [ne]\NP respectively. In Type 2, respective CCG categories for 'raam' and 'ne' are NP and NP\NP. Morphological information such as case (e.g., Ergative case - 'ne') in noun categories is expected to help with determining their dependency labels, but makes the lexicon more sparse.

## 2.3 Morphological Markers

In Hindi, morphological information is encoded in the form of post-positional markers on nouns, and tense, aspect and modality markers on verbs. A post-positional marker following a noun plays the role of a case-marker (e.g., 'raam ne (Ram ERG)', here 'ne' is the ergative case marker) and can also have a role similar to English prepositions (e.g., 'mej par (table on)'). Post-positional markers on nouns can be simple one word expressions like 'ne' or 'par' or can be multiple words as in 'raam ke lie (Ram for)'. Complex post position markers as a whole give information about how the head noun or verb behaves. We merged complex post position markers into single words like 'ke\_lie' so

<sup>1</sup><http://nextens.uvt.nl/depparse-wiki/DataFormat>

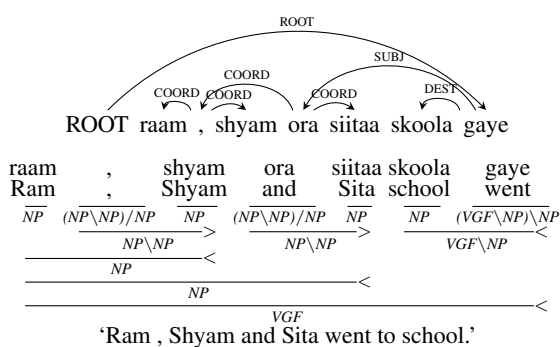
that the entire marker gets a single CCG category.

For an adjunct like ‘raam ke.lie (for Ram)’ in Figure 2, ‘raam’ can have a CCG category  $VG\bar{F}/VG\bar{F}$  as it is the head of the chunk and ‘ke.lie’ a category of  $(VG\bar{F}/VG\bar{F}) \setminus (VG\bar{F}/VG\bar{F})$ . Alternatively, if we pass the adjunct information to the post-position marker (‘ke.lie’), and use the chunk tag ‘NP’ as the category for the head word (‘raam’), then categories of ‘raam’ and ‘ke.lie’ are NP and  $(VG\bar{F}/VG\bar{F}) \setminus NP$  respectively. Though both these analysis give the same semantics, we chose the latter as it leads to a less sparse lexicon. Also, adjuncts that modify adjacent adjuncts are assigned identical categories  $X/X$  making use of CCG’s composition rule and following Çakıcı (2005).

## 2.4 Co-ordination

The CCG category of a conjunction is  $(X \setminus X) / X$ , where a conjunction looks for a child to its right and then a child to its left. To handle conjunction with multiple children, we modified the dependency tree, as follows.

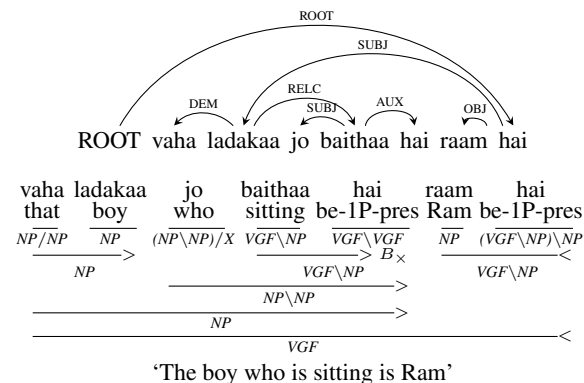
For the example given below, in the original dependency tree, the conjunction **ora** ‘and’ has three children ‘Ram’, ‘Shyam’ and ‘Sita’. We modified the original dependency tree and treat the comma ‘,’ as a conjunction. As a result, ‘,’ will have ‘Ram’ and ‘Shyam’ as children and ‘and’ will have ‘,’ and ‘Sita’ as children. It is straightforward to convert this tree into the original dependency tree for the purpose of evaluation/comparison with other dependency parsers.



## 2.5 Relative Clauses

In English, relative clauses have the category type  $NP \setminus NP$ , where they combine with a noun phrase on the left to give a resulting noun phrase. Hindi, due to its freer word order, has relative clauses of the type  $NP \setminus NP$  or  $NP / NP$  based on the position of the relative clause with respect to the head noun. Similar to English, the relative pronoun has a CCG

category of  $(NP | NP) | X$  where directionality depends on the position of the relative pronoun in the clause and the category  $X$  depends on the grammatical role of the relative pronoun. In the following example,  $X$  is  $VG\bar{F} \setminus NP$



## 2.6 CCG Lexicon to Treebank conversion

We use a CCG parser to convert the CCG lexicon to a CCG treebank as conversion to CCG trees directly from dependency trees is not straightforward. Using the above algorithm, we get one CCG category for every word in a sentence. We then run a non-statistical CKY chart parser based on the CCG formalism<sup>2</sup>, which gives CCG derivations based on the lexical categories. This gives multiple derivations for some sentences. We rank these derivations using two criteria. The first criterion is correct recovery of the gold dependency tree. Derivations which lead to gold dependencies are given higher weight. In the second criterion, we prefer derivations which yield intra-chunk dependencies (e.g., verb and auxiliary) prior to inter-chunk (e.g., verb and its arguments). For example, morphological markers (which lead to intra-chunk dependencies) play a crucial role in identifying correct dependencies. Resolving these dependencies first helps parsers in better identification of inter-chunk dependencies such as argument structure of the verb (Ambati, 2011). We thus extract the best derivation for each sentence and create a CCGbank for Hindi. Coverage, i.e., number of sentences for which we got at least one complete derivation, using this lexicon is 96%. The remaining 4% are either cases of wrong annotations in the original treebank, or rare constructions which are currently not handled by our conversion algorithm.

<sup>2</sup><http://openccg.sourceforge.net/>

### 3 Experiments

In this section, we first describe the method of developing a supertagger using the CCGbank. We then describe different ways of providing CCG categories from the supertagger as features to a state-of-the-art Hindi Dependency parser (Malt).

We did all our experiments using both gold features (pos, chunk and morphological information) provided in the treebank and automatic features extracted using a Hindi shallow parser<sup>3</sup>. We report results with automatic features but we also obtained similar improvements with gold features.

#### 3.1 Category Set

For supertagging, we first obtained a category set from the CCGbank training data. There are 2,177 and 718 category types in Type 1 (with morph. information) and Type 2 (without morph. information) data respectively. Clark and Curran (2004) showed that using a frequency cutoff can significantly reduce the size of the category set with only a small loss in coverage. We explored different cut-off values and finally used a cutoff of 10 for building the tagger. This reduced the category types to 376 and 202 for Type 1 and Type 2 respectively. The percent of category tokens in development data that don't appear in the category set entailed by this cut-off are 1.39 & 0.47 for Type 1 and Type 2 respectively.

#### 3.2 Supertagger

Following Clark and Curran (2004), we used a Maximum Entropy approach to build our supertagger. We explored different features in the context of a 5-word window surrounding the target word. We used features based on WORD ( $w$ ), LEMMA ( $l$ ), POS ( $p$ ), CPOS ( $c$ ) and the FEATS ( $f$ ) columns of the CoNLL format. Table 1 shows the impact of different features on supertagger performance. Experiments 1, 2, 3 have current word ( $w_i$ ) features while Experiments 4, 5, 6 show the impact of contextual and complex bi-gram features.

Accuracy of the supertagger after Experiment 6 is 82.92% and 84.40% for Type 1 and Type 2 data respectively. As the number of category types in Type 1 data (376) are much higher than in Type 2 (202), it is not surprising that the performance of the supertagger is better for Type 2 as compared to Type 1.

<sup>3</sup><http://ltrc.iiit.ac.in/analyzer/hindi/>

| Experiments: Features   | Accuracy     |              |
|---|--------------|--------------|
|   | Type 1       | Type 2       |
| Exp 1: $w_i, p_i$   | 75.14        | 78.47        |
| Exp 2: $Exp 1 + l_i, c_i$   | 77.58        | 80.17        |
| Exp 3: $Exp 2 + f_i$  | 80.43        | 81.88        |
| Exp 4: $Exp 3 + w_{i-1}, w_{i-2}, p_{i-1}, p_{i-2},$<br>$w_{i+1}, w_{i+2}, p_{i+1}, p_{i+2}$  | 82.72        | 84.15        |
| Exp 5: $Exp 4 + w_i p_i, w_i c_i, w_i f_i, p_i f_i$   | 82.81        | 84.29        |
| Exp 6: $Exp 5 + w_{i-2} w_{i-1}, w_{i-1} w_i,$<br>$w_i w_{i+1}, w_{i+1} w_{i+2}, p_{i-2} p_{i-1},$<br>$p_{i-1} p_i, p_i p_{i+1}, p_{i+1} p_{i+2}$ | <b>82.92</b> | <b>84.40</b> |

Table 1: Impact of different features on the supertagger performance for development data.

#### 3.3 Dependency Parsing

There has been a significant amount of work on Hindi dependency parsing in the recent past (Husain, 2009; Husain et al., 2010; Bharati et al., 2012). Out of all these efforts, state-of-the-art accuracy is achieved using the Malt parser. We first run Malt with previous best settings (Bharati et al., 2012) which use the arc-standard parsing algorithm with a liblinear learner, and treat this as our baseline. We compare and analyze results after adding supertags as features with this baseline.

#### 3.4 Using Supertags as Features to Malt

Çakıcı (2009) showed that using gold CCG categories extracted from dependency trees as features to MST parser (McDonald et al., 2006) boosted the performance for Turkish. But using automatic categories from a supertagger radically decreased performance in their case as supertagger accuracy was very low. We have explored different ways of incorporating both gold CCG categories and supertagger-provided CCG categories into dependency parsing. Following Çakıcı (2009), instead of using supertags for all words, we used supertags which occurred at least K times in the training data, and backed off to coarse POS-tags otherwise. We experimented with different values of K and found that K=15 gave the best results.

We first provided gold CCG categories as features to the Malt parser and then provided the output of the supertagger described in section 3.2. We did all these experiments with both Type 1 and Type 2 data. Unlabelled Attachment Scores (UAS) and Labelled Attachment Scores (LAS) for Malt

are shown in Table 2. As expected, gold CCG categories boosted UAS and LAS by around 6% and 7% respectively, for both Type 1 and Type 2 data. This clearly shows that the rich subcategorization information provided by CCG categories can help a shift-reduce parser. With automatic categories from a supertagger, we also got improvements over the baseline, for both Type 1 and Type 2 data. All the improvements are statistically significant (McNemar’s test,  $p < 0.01$ ).

With gold CCG categories, Type 1 data gave slightly better improvements over Type 2 as Type 1 data has richer morphological information. But, in the case of supertagger output, Type 2 data gave more improvements over the baseline Malt as compared to Type 1. This is because the performance of the supertagger on Type 2 data is slightly better than that of Type 1 data (see Table 1).

| <i>Experiment</i>  | <i>Development</i> |               | <i>Testing</i> |               |
|--------------------|--------------------|---------------|----------------|---------------|
|                    | <i>UAS</i>         | <i>LAS</i>    | <i>UAS</i>     | <i>LAS</i>    |
| Malt: Baseline     | 89.09              | 83.46         | 88.67          | 83.04         |
| Malt + Type 1 Gold | 95.87*             | 90.79*        | 95.27*         | 90.22*        |
| Malt + Type 2 Gold | 95.73*             | 90.70*        | 95.26*         | 90.18*        |
| Malt + Type 1 ST   | 89.54*             | 83.68*        | 88.93*         | 83.23*        |
| Malt + Type 2 ST   | <b>89.90*</b>      | <b>83.96*</b> | <b>89.04*</b>  | <b>83.35*</b> |

Table 2: Supertagger impact on Hindi dependency parsing (ST=Supertags). McNemar’s test, \* =  $p < 0.01$ .

It is interesting to notice the impact of using automatic CCG categories from a supertagger on long distance dependencies. It is known that Malt is weak at long-distance relations (McDonald and Nivre, 2007; Ambati et al., 2010). Providing CCG categories as features improved handling of long-distance dependencies for Malt. Figure 3 shows the F-score of the impact of CCG categories on three dependency labels, which take the major share of long distance dependencies, namely, ROOT, COORD, and RELC, the labels for sentence root, co-ordination, and relative clause respectively. For these relations, providing CCG categories gave an increment of 1.2%, 1.4% and 1.6% respectively over the baseline.

We also found that the impact of CCG categories is higher when the span of the dependency is longer. Figure 4 shows the F-score of the impact of CCG categories on dependencies based on the distance between words. Using CCG categories

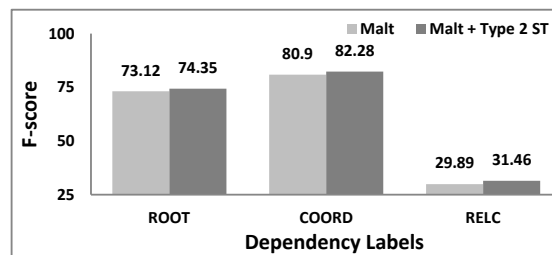


Figure 3: Label-wise impact of supertag features.

does not have much impact on short distance dependencies (1–5), which Malt is already good at. For longer range distances, 6–10, and >10, there is an improvement of 1.8% and 1.4% respectively.

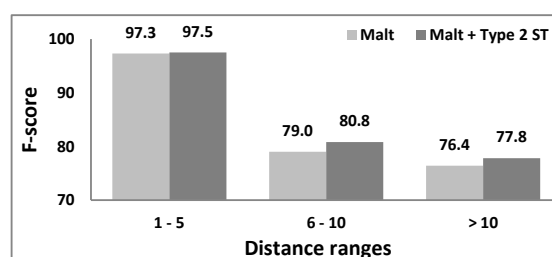


Figure 4: Impact of supertags on distance ranges.

## 4 Conclusion and Future Direction

We have presented an approach for automatically extracting a CCG lexicon from a dependency treebank for Hindi. We have also presented a novel way of creating a CCGbank from a dependency treebank using a CCG parser and the CCG lexicon. Unlike previous work, we obtained improvements in dependency recovery using automatic supertags, as well as gold information. We have shown that informative CCG categories improve the performance of a shift-reduce dependency parser (Malt) in recovering some long distance relations. In future work, we would like to directly train a CCG shift-reduce parser (such as Zhang and Clark (2011)’s English parser) on the Hindi CCGbank. We would also like to see the impact of generalisation of our lexicon using the free-word order formalism for CCG categories of Baldrige (2002).

## Acknowledgements

We would like to thank three anonymous reviewers for their useful suggestions. This work was supported by ERC Advanced Fellowship 249520 GRAMPLUS.



## References

- Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010. 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, Los Angeles, CA, USA, June.
- Bharat Ram Ambati. 2011. *Hindi Dependency Parsing and Treebank Validation*. Master's Thesis, International Institute of Information Technology - Hyderabad, India.
- Jason M. Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorical Grammar*. Ph.D. thesis, University of Edinburgh, UK.
- Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995. Natural Language Processing: A Paninian Perspective. *Prentice-Hall of India*, pages 65–106.
- Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma, and Lakshmi Bai. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. In *Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad*.
- Akshar Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begum, and Rajeev Sangal. 2009. AnnCorra: TreeBanks for Indian Languages, Guidelines for Annotating Hindi TreeBank (version 2.0). <http://ltrc.iiit.ac.in/MachineTrans/research/tb/DS-guidelines/DS-guidelines-ver2-28-05-09.pdf>.
- Akshar Bharati, Prashanth Mannem, and Dipti Misra Sharma. 2012. Hindi Parsing Shared Task. In *Proceedings of Coling Workshop on Machine Translation and Parsing in Indian Languages*, Kharagpur, 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 Third Linguistic Annotation Workshop at 47th ACL and 4th IJCNLP*, pages 186–189, Suntec, Singapore.
- Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a Dependency Treebank to a Categorical Grammar Treebank for Italian. In M. Passarotti, Adam Przepiórkowski, S. Raynaud, and Frank Van Eynde, editors, *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)*, pages 27–38, Milan, Italy.
- Ruken Çakıcı. 2005. Automatic induction of a CCG grammar for Turkish. In *Proceedings of Student Research Workshop, 43rd Annual Meeting of the ACL*, pages 73–78.
- Ruket Çakıcı. 2009. *Parser Models for a Highly Inflected Language*. Ph.D. thesis, University of Edinburgh, UK.
- Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of COLING-04*, pages 282–288.
- Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, September.
- Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44*, pages 505–512, Sydney, Australia.
- Samar Husain, Prashanth Mannem, Bharat Ram Ambati, and Phani Gadde. 2010. The ICON-2010 Tools Contest on Indian Language Dependency Parsing. In *Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing*, Kharagpur, India.
- Samar Husain. 2009. Dependency Parsers for Indian Languages. In *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, India.
- Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*.
- Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 216–220, New York City, New York.
- Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.
- Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.
- Daniel Tse and James R. Curran. 2010. Chinese CCGbank: extracting CCG derivations from the Penn Chinese Treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 1083–1091, Beijing, China.
- Yue Zhang and Stephen Clark. 2011. Shift-Reduce CCG Parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 683–692, Portland, Oregon, USA, June.

# The Effect of Higher-Order Dependency Features in Discriminative Phrase-Structure Parsing

Gregory F. Coppola and Mark Steedman

School of Informatics

The University of Edinburgh

g.f.coppola@sms.ed.ac.uk

steedman@inf.ed.ac.uk

## Abstract

Higher-order dependency features are known to improve dependency parser accuracy. We investigate the incorporation of such features into a cube decoding *phrase-structure* parser. We find considerable gains in accuracy on the range of standard metrics. What is especially interesting is that we find strong, statistically significant gains on dependency recovery on *out-of-domain* tests (Brown vs. WSJ). This suggests that higher-order dependency features are not simply overfitting the training material.

## 1 Introduction

*Higher-order* dependency features encode more complex sub-parts of a dependency tree structure than first-order, bigram head-modifier relationships.<sup>1</sup> The clear trend in dependency parsing has been that the addition of such higher-order features improves parse accuracy (McDonald & Pereira, 2006; Carreras, 2007; Koo & Collins, 2010; Zhang & Nivre, 2011; Zhang & McDonald, 2012). This finding suggests that the same benefits might be observed in phrase-structure parsing. But, this is not necessarily implied. Phrase-structure parsers are generally stronger than dependency parsers (Petrov et al., 2010; Petrov & McDonald, 2012), and make use of more kinds of information. So, it might be that the information modelled by higher-order dependency features adds less of a benefit in the phrase-structure case.

<sup>1</sup>Examples of first-order and higher-order dependency features are given in §3.2.

To investigate this issue, we experiment using Huang’s (2008) *cube decoding* algorithm. This algorithm allows structured prediction with *non-local features*, as discussed in §2. Collins’s (1997) strategy of expanding the phrase-structure parser’s dynamic program to incorporate head-modifier dependency information would not scale to the complex kinds of dependencies we will consider. Using Huang’s algorithm, we can indeed incorporate arbitrary types of dependency feature, using a single, simple dynamic program.

Compared to the baseline, non-local feature set of Collins (2000) and Charniak & Johnson (2005), we find that higher-order dependencies do in fact tend to improve performance significantly on both dependency and constituency accuracy metrics. Our most interesting finding, though, is that higher-order dependency features show a consistent and unambiguous contribution to the dependency accuracy, both labelled and unlabelled, of our phrase-structure parsers on *out-of-domain* tests (which means, here, trained on WSJ, but tested on BROWN). In fact, the gains are even stronger on out-of-domain tests than on in-domain tests. One might have thought that higher-order dependencies, being rather specific by nature, would tend to pick out only very rare events, and so only serve to over-fit the training material, but this is not what we find. We speculate as to what this might mean in §5.2.

The cube decoding paradigm requires a first-stage parser to prune the output space. For this, we use the generative parser of Petrov et al. (2006). We can use this parser’s model score as a feature in our discriminative model at no additional cost. However, doing so conflates the contribution to accuracy of the generative model, on the one hand, and the discriminatively trained, hand-

written, features, on the other. Future systems might use the same or a similar feature set to ours, but in an architecture that does not include any generative parser. On the other hand, some systems might indeed incorporate this generative model's score. So, we need to know exactly what the generative model is contributing to the accuracy of a generative-discriminative model combination. Thus, we conduct experiments in sets: in some cases the generative model score is used, and in others it is not used.

Compared to the faster and more psychologically plausible shift-reduce parsers (Zhang & Nivre, 2011; Zhang & Clark, 2011), cube decoding is a computationally expensive method. But, cube decoding provides a relatively exact environment with which to compare different feature sets, has close connections with modern phrase-based machine translation methods (Huang & Chiang, 2007), and produces very accurate parsers. In some cases, one might want to use a slower, but more accurate, parser during the training stage of a semi-supervised parser training strategy. For example, Petrov et al. (2010) have shown that a fast parser (Nivre et al., 2007) can be profitably trained from the output of a slower but more accurate one (Petrov et al., 2006), in a strategy they call *uptraining*.

We make the source code for these experiments available.<sup>2</sup>

## 2 Phrase-Structure Parsing with Non-Local Features

### 2.1 Non-Local Features

To decode using exact dynamic programming (i.e., CKY), one must restrict oneself to the use of only *local* features. Local features are those that factor according to the individual rule productions of the parse. For example, a feature indicating the presence of the rule  $S \rightarrow NP VP$  is local.<sup>3</sup> But, a feature that indicates that the head word of this  $S$  is, e.g., *joined*, is non-local, because the head word of a phrase cannot be determined by looking at a single rule production. To find a phrase's head word (or tag), we must recursively find the

<sup>2</sup>See <http://gfcoppola.net/code.php>. This software is available for free for non-profit research uses.

<sup>3</sup>A feature indicating that, e.g., the first word dominated by  $S$  is *Pierre* is also local, since the words of the sentence are constant across hypothesized parses, and words can be referred to by their position with respect to a given rule production. See Huang (2008) for more details.

head phrase of each local rule production, until we reach a terminal node (or tag node). This recursion would not be allowed in standard CKY. Many discriminative parsers have used only local features (Taskar et al., 2004; Turian et al., 2007; Finkel et al., 2008). However, Huang (2008) shows that the use of non-local features does in fact contribute substantially to parser performance. And, our desire to make heavy use of head-word dependency relations necessitates the use of non-local features.

### 2.2 Cube Decoding

While the use of non-local features destroys the ability to do exact search, we can still do inexact search using Huang's (2008) *cube decoding* algorithm.<sup>4</sup> A tractable first-stage parser prunes the space of possible parses, and outputs a *forest*, which is a set of rule production instances that can be used to make a parse for the given sentence, and which is significantly pruned compared to the entire space allowed by the grammar. The size of this forest is at most cubic in the length of the sentence (Billot & Lang, 1989), but implicitly represents exponentially many parses. To decode, we fix an beam width of  $k$  (an integer). Then, when parsing, we visit each node  $n$  in the same bottom-up order we would use for Viterbi decoding, and compute a list of the top  $k$  parses to  $n$ , according to a global linear model (Collins, 2002), using the trees that have survived the beam at earlier nodes.

### 2.3 The First-Stage Parser

As noted, we require a first-stage parser to prune the search space.<sup>5</sup> As a by-product of this pruning procedure, we are able to use the model score of the first-stage parser as a feature in our ultimate model at no additional cost. As a first-stage parser, we use Huang et al.'s (2010) implementation of the LA-PCFG parser of Petrov et al. (2006), which uses a generative, latent-variable model.

## 3 Features

### 3.1 Phrase-Structure Features

Our phrase-structure feature set is taken from Collins (2000), Charniak & Johnson (2005), and

<sup>4</sup>This algorithm is closely related to the algorithm for phrase-based machine translation using a language model (Huang & Chiang, 2007).

<sup>5</sup>All work in this paradigm has used a generative parser as the first-stage parser. But, this is arguably a historical accident. We could just as well use a discriminative parser with only local features, like Petrov & Klein (2007a).

Huang (2008). Some features are omitted, with choices made based on the ablation studies of Johnson & Ural (2010). This feature set, which we call  $\Phi_{\text{phrase}}$ , contains the following, mostly non-local, features, which are described and depicted in Charniak & Johnson (2005), Huang (2008), and Johnson & Ural (2010):

- **CoPar** The depth (number of levels) of parallelism between adjacent conjuncts
- **CoParLen** The difference in length between adjacent conjuncts
- **Edges** The words or (part-of-speech) tags on the outside and inside edges of a given XP<sup>6</sup>
- **NGrams** Sub-parts of a given rule production
- **NGramTree** An  $n$ -gram of the input sentence, or the tags, along with the minimal tree containing that  $n$ -gram
- **HeadTree** A sub-tree containing the path from a word to its maximal projection, along with all siblings of all nodes in that path
- **Heads** Head-modifier bigrams
- **Rule** A single rule production
- **Tag** The tag of a given word
- **Word** The tag of and first XP above a word
- **WProj** The tag of and maximal projection of a word

**Heads** is a first-order dependency feature.

### 3.2 Dependency Parsing Features

McDonald et al. (2005) showed that chart-based dependency parsing, based on Eisner’s (1996) algorithm, could be successfully approached in a discriminative framework. In this earliest work, each feature function could only refer to a single, bigram head-modifier relationship, e.g., **Modifier**, below. Subsequent work (McDonald & Pereira, 2006; Carreras, 2007; Koo & Collins, 2010) looked at allowing features to access more complex, *higher-order* relationships, including trigram and 4-gram relationships, e.g., all features apart from **Modifier**, below. With the ability to incorporate non-local phrase-structure parse features (Huang, 2008), we can recognize dependency features of arbitrary order (cf. Zhang & McDonald (2012)). Our dependency feature set, which we call  $\Phi_{\text{deps}}$ , contains:

- **Modifier** head and modifier

<sup>6</sup>The tags outside of a given XP are approximated using the marginally most likely tags given the parse.

- **Sibling** head, modifier  $m$ , and  $m$ ’s nearest inner sibling
- **Grandchild** head, modifier  $m$ , and one of  $m$ ’s modifiers
- **Sibling+Grandchild** head, modifier  $m$ ,  $m$ ’s nearest inner sibling, and one of  $m$ ’s modifiers
- **Grandchild+Grandsibling** head, modifier  $m$ , one of  $m$ ’s modifiers  $g$ , and  $g$ ’s inner sibling

These features are insensitive to arc labels in the present experiments, but future work will incorporate arc labels. Each feature class contains more and less lexicalized versions.

### 3.3 Generative Model Score Feature

Finally, we have a feature set,  $\Phi_{\text{gen}}$ , containing only one feature function. This feature maps a parse to the logarithm of the MAX-RULE-PRODUCT score of that parse according to the LAPCFG parsing model, which is trained separately. This score has the character of a conditional likelihood for the parse (see Petrov & Klein (2007b)).

## 4 Training

We have two feature sets  $\Phi_{\text{phrase}}$  and  $\Phi_{\text{deps}}$ , for which we fix weights using parallel stochastic optimization of a structured SVM objective (Collins, 2002; Taskar et al., 2004; Crammer et al., 2006; Martins et al., 2010; McDonald et al., 2010). To the single feature in the set  $\Phi_{\text{gen}}$  (i.e. the generative model score), we give the weight 1.

The combined models,  $\Phi_{\text{phrase+deps}}$ ,  $\Phi_{\text{phrase+gen}}$ , and  $\Phi_{\text{phrase+deps+gen}}$ , are then *model combinations* of the first three. The combination weights for these combinations are obtained using Och’s (2003) Minimum Error-Rate Training (MERT). The MERT stage helps to avoid feature under-training (Sutton et al., 2005), and avoids the problem of scaling involved in a model that contains mostly boolean features, but one, real-valued, log-scale feature. Training is conducted in three stages (SVM, MERT, SVM), so that there is no influence of any data outside the given training set (WSJ2-21) on the combination weights.

## 5 Experiments

### 5.1 Methods

All models are trained on WSJ2-21, with WSJ22 used to pick the stopping iteration for online

|      |                 | Test Set    |             |             |             |             |             |
|------|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|
|      |                 | WSJ         |             |             | BROWN       |             |             |
| Type | Model           | $F_1$       | UAS         | LAS         | $F_1$       | UAS         | LAS         |
| G    | LA-PCFG         | 90.3        | 93.7        | 91.5        | 85.1        | 88.7        | 85.0        |
| D    | phrase          | 91.2        | 93.9        | 91.0        | 86.1        | 89.4        | 85.1        |
|      | deps            | —           | 93.3        | —           | —           | 89.3        | —           |
|      | phrase+deps     | <b>91.7</b> | <b>94.4</b> | <b>91.5</b> | <b>86.4</b> | <b>90.1</b> | <b>85.9</b> |
| G+D  | phrase+gen      | 92.1        | 94.7        | 92.6        | 87.0        | 90.0        | 86.5        |
|      | phrase+deps+gen | <b>92.4</b> | <b>94.9</b> | <b>92.8</b> | <b>87.4</b> | <b>90.7</b> | <b>87.1</b> |

Table 1: Performance of the various models in cube decoding experiments, on the WSJ test set (in-domain) and the BROWN test set (out-of-domain). G abbreviates *generative*, D abbreviates *discriminative*, and G+D a combination. Some cells are empty because  $\Phi_{\text{deps}}$  features are only sensitive to unlabelled dependencies. Best results in D and G+D conditions appear in bold face.

| Hypothesis      |             | Test Set    |                 |                 |             |             |                 |
|-----------------|-------------|-------------|-----------------|-----------------|-------------|-------------|-----------------|
|                 |             | WSJ         |                 |                 | BROWN       |             |                 |
| Greater         | Lesser      | $F_1$       | UAS             | LAS             | $F_1$       | UAS         | LAS             |
| phrase+deps     | phrase      | <b>.042</b> | <b>.029</b>     | <b>.018</b>     | .140        | <b>.022</b> | <b>.009</b>     |
| phrase+deps     | deps        | —           | <b>&lt;.001</b> | —               | —           | <b>.012</b> | —               |
| phrase+gen      | phrase      | <b>.013</b> | <b>.003</b>     | <b>&lt;.001</b> | <b>.016</b> | .090        | <b>&lt;.001</b> |
| phrase+deps+gen | phrase+gen  | <b>.030</b> | .122            | .151            | .059        | <b>.008</b> | <b>.020</b>     |
| phrase+deps+gen | phrase+deps | <b>.019</b> | <b>.020</b>     | <b>&lt;.001</b> | <b>.008</b> | <b>.040</b> | <b>&lt;.001</b> |

Table 2: Results of statistical significance evaluations of hypotheses of the form  $X$ 's accuracy is greater than  $Y$ 's on the various test sets and metrics. Bold face indicates  $p < .05$ .

optimization, as is standard. The test sets are WSJ23 (in-domain test set), and BROWN9 (out-of-domain test set) from the Penn Treebank (Marcus et al., 1993).<sup>7</sup> We evaluate using harmonic mean between labelled bracket recall and precision (EVALB  $F_1$ ), unlabelled dependency accuracy (UAS), and labelled dependency accuracy (LAS). Dependencies are extracted from full output trees using the algorithm of de Marneffe & Manning (2008). We chose this dependency extractor, firstly, because it is natively meant to be run on the output of phrase-structure parsers, rather than on gold trees with function tags and traces still present, as is, e.g., the *Penn-Converter* of Johansson & Nugues (2007). Also, this is the extractor that was used in a recent shared task (Petrov & McDonald, 2012). We use EVALB and *eval.pl* to calculate scores.

For hypothesis testing, we used the paired bootstrap test recently empirically evaluated in the context of NLP by Berg-Kirkpatrick et al. (2012). This

<sup>7</sup>Following Gildea (2001), the BROWN test set is usually divided into 10 parts. If we start indexing at 0, then the last (test) section has index 9. We received the BROWN data splits from David McClosky, p.c.

involves drawing  $b$  subsamples of size  $n$  with replacement from the test set in question, and checking relative performance of the models on the subsample (see the reference). We use  $b = 10^6$  and  $n = 500$  in all tests.

## 5.2 Results

The performance of the models is shown in Table 1, and Table 2 depicts the results of significance tests of differences between key model pairs.

We find that adding in the higher-order dependency feature set,  $\Phi_{\text{deps}}$ , makes a statistically significant improvement in accuracy on most metrics, in most conditions. On the in-domain WSJ test set, we find that  $\Phi_{\text{phrase+deps}}$  is significantly better than either of its component parts on all metrics. But,  $\Phi_{\text{phrase+deps+gen}}$  is significantly better than  $\Phi_{\text{phrase+gen}}$  only on  $F_1$ , but not on UAS or LAS. However, on the *out-of-domain* BROWN tests, we find that adding  $\Phi_{\text{deps}}$  always adds considerably, and in a statistically significant way, to both LAS and UAS. That is, not only is  $\Phi_{\text{phrase+deps}}$  better at dependency recovery than its component parts, but  $\Phi_{\text{phrase+deps+gen}}$  is also considerably bet-

ter on dependency recovery than  $\Phi_{\text{phrase+gen}}$ , which represents the previous state-of-the-art in this vein of research (Huang, 2008). This result is perhaps counter-intuitive, in the sense that one might have supposed that higher-order dependency features, being highly specific by nature, might only have only served to over-fit the training material. However, this result shows otherwise. Note that the dependency features include various levels of lexicalization. It might be that the more unlexicalized features capture something about the structure of correct parses, that transfers well out-of-domain. Future work should investigate this. And, it of course remains to be seen how this result will transfer to other train-test domain pairs.

To our knowledge, this is the first work to specifically separate the role of the generative model feature from the other features of Collins (2000) and Charniak & Johnson (2005). We note that, even without the  $\Phi_{\text{gen}}$  feature, the discriminative parsing models are very strong, but adding  $\Phi_{\text{gen}}$  nevertheless yields considerable gains. Thus, while a fully discriminative model, perhaps implemented using a shift-reduce algorithm, can be expected to do very well, if the best accuracy is necessary (e.g., in a semi-supervised training strategy), it still seems to pay to use the generative-discriminative model combination. Note that the LAS scores of our models without  $\Phi_{\text{gen}}$  are relatively weak. This is presumably largely because our dependency features are, at present, not sensitive to arc labels, so our results probably underestimate the capability of our general framework with respect to labelled dependency recovery.

Table 3 compares our work with Huang’s (2008). Note that our model  $\Phi_{\text{phrase+gen}}$  uses essentially the same features as Huang (2008), so the fact that our  $\Phi_{\text{phrase+gen}}$  is noticeably more accurate on  $F_1$  is presumably due to the benefits in reduced feature under-training achieved by the MERT combination strategy. Also, our  $\Phi_{\text{phrase+deps}}$  model is as accurate as Huang’s, without even using the generative model score feature. Table 4 compares our work to McClosky et al.’s (2006) domain adaptation work with the Charniak & Johnson (2005) parser. Their three models shown have been trained on: i) the WSJ (supervised, out-of-domain), ii) the WSJ plus 2.5 million sentences of automatically labelled NANC newswire text (semi-supervised, out-of-domain), and iii) the BROWN corpus (supervised, in-domain). We test

| Type | Model           | WSJ         |
|------|-----------------|-------------|
| G+D  | Huang (2008)    | 91.7        |
| D    | phrase+deps     | 91.7        |
| G+D  | phrase+gen      | 92.1        |
| G+D  | phrase+deps+gen | <b>92.4</b> |

Table 3: Comparison of constituency parsing results in the cube decoding framework, on the WSJ test set. On G+D, D, see Table 1.

| Parser   | Training Data | BROWN $F_1$ |
|----------|---------------|-------------|
| CJ       | WSJ           | 85.2        |
| CJ       | WSJ+NANC      | 87.8        |
| CJ       | BROWN         | <b>88.4</b> |
| Our Best | WSJ           | <u>87.4</u> |

Table 4: Comparison of our best model,  $\Phi_{\text{phrase+deps+gen}}$ , on BROWN, with the Charniak & Johnson (2005) parser, denoted CJ, as reported in McClosky et al. (2006). Underline indicates best trained on WSJ, bold face indicates best overall.

on BROWN. We see that our best (WSJ-trained) model is over 2% more accurate (absolute  $F_1$  difference) than the Charniak & Johnson (2005) parser trained on the same data. In fact, our best model is nearly as good as McClosky et al.’s (2006) self-trained, semi-supervised model. Of course, the self-training strategy is orthogonal to the improvements we have made.

## 6 Conclusion

We have shown that the addition of higher-order dependency features into a cube decoding phase-structure parser leads to statistically significant gains in accuracy. The most interesting finding is that these gains are clearly observed on out-of-domain tests. This seems to imply that higher-order dependency features do not merely over-fit the training material. Future work should look at other train-test domain pairs, as well as look at exactly which higher-order dependency features are most important to out-of-domain accuracy.

## Acknowledgments

This work was supported by the Scottish Informatics and Computer Science Alliance, The University of Edinburgh’s School of Informatics, and ERC Advanced Fellowship 249520 GRAMPLUS. We thank Zhongqiang Huang for his extensive help in getting started with his LA-PCFG parser.

## References

- Berg-Kirkpatrick, T., Burkett, D., & Klein, D. (2012). An empirical investigation of statistical significance in NLP. In *EMNLP*, 995–1005.
- Billot, S., & Lang, B. (1989). The structure of shared forests in ambiguous parsing. In *ACL*, 143–151.
- Carreras, X. (2007). Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, 957–961.
- Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *ACL*, 173–180.
- Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *ACL*, 16–23.
- Collins, M. (2000). Discriminative reranking for natural language parsing. In *ICML*, 175–182.
- Collins, M. (2002). Discriminative training methods for Hidden Markov Models: theory and experiments with perceptron algorithms. In *EMNLP*, 1–8.
- Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *JMLR*, 7, 551–585.
- Eisner, J. (1996). Three new probabilistic models for dependency parsing: An exploration. In *COLING*, 340–345.
- Finkel, J. R., Kleeman, A., & Manning, C. D. (2008). Efficient, feature-based, conditional random field parsing. In *ACL*, 959–967.
- Gildea, D. (2001). Corpus variation and parser performance. In *EMNLP*, 167–202.
- Huang, L. (2008). Forest reranking: Discriminative parsing with non-local features. In *ACL*, 586–594.
- Huang, L., & Chiang, D. (2007). Forest rescoring: Faster decoding with integrated language models. In *ACL*.
- Huang, Z., Harper, M., & Petrov, S. (2010). Self-training with products of latent variable grammars. In *EMNLP*, 12–22.
- Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*, 105–112.
- Johnson, M., & Ural, A. E. (2010). Reranking the Berkeley and Brown parsers. In *HLT-NAACL*, 665–668.
- Koo, T., & Collins, M. (2010). Efficient third-order dependency parsers. In *ACL*, 1–11.
- Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313–330.
- de Marneffe, M.-C., & Manning, C. D. (2008). The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1–8.
- Martins, A. F., Gimpel, K., Smith, N. A., Xing, E. P., Figueiredo, M. A., & Aguiar, P. M. (2010). Learning structured classifiers with dual coordinate ascent. Technical report, DTIC Document.
- McClosky, D., Charniak, E., & Johnson, M. (2006). Reranking and self-training for parser adaptation. In *ACL*, 337–344.
- McDonald, R., & Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *EACL*, 81–88.
- McDonald, R. T., Crammer, K., & Pereira, F. C. N. (2005). Online large-margin training of dependency parsers. In *ACL*, 91–98.
- McDonald, R. T., Hall, K., & Mann, G. (2010). Distributed training strategies for the structured perceptron. In *HLT-NAACL*, 456–464.
- Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., & Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2), 95–135.
- Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *ACL*, 160–167.
- Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *ACL*, 433–440.
- Petrov, S., Chang, P.-C., Ringgaard, M., & Alshawi, H. (2010). Uptraining for accurate deterministic question parsing. In *EMNLP*, 705–713.
- Petrov, S., & Klein, D. (2007a). Discriminative log-linear grammars with latent variables. In *NIPS*.

- Petrov, S., & Klein, D. (2007b). Improved inference for unlexicalized parsing. In *HLT-NAACL*, 404–411.
- Petrov, S., & McDonald, R. (2012). Overview of the 2012 shared task on parsing the web. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).
- Sutton, C., Sindelar, M., & McCallum, A. (2005). Feature bagging: Preventing weight undertraining in structured discriminative learning. In *HLT-NAACL*.
- Taskar, B., Klein, D., Collins, M., Koller, D., & Manning, C. D. (2004). Max-margin parsing. In *EMNLP*, 1–8.
- Turian, J., Wellington, B., & Melamed, I. D. (2007). Scalable discriminative learning for natural language parsing and translation. In *NIPS*, 1409–1416.
- Zhang, H., & McDonald, R. (2012). Generalized higher-order dependency parsing with cube pruning. In *EMNLP*, 238–242.
- Zhang, Y., & Clark, S. (2011). Shift-reduce CCG parsing. In *ACL*, 683–692.
- Zhang, Y., & Nivre, J. (2011). Transition-based dependency parsing with rich non-local features. In *ACL*, 188–293.



# Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers

André F. T. Martins\*<sup>†</sup> Miguel B. Almeida\*<sup>†</sup> Noah A. Smith<sup>#</sup>

\*Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal

<sup>†</sup>Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal

<sup>#</sup>School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

{atm,mba}@priberam.pt, nasmith@cs.cmu.edu

## Abstract

We present fast, accurate, direct non-projective dependency parsers with third-order features. Our approach uses AD<sup>3</sup>, an accelerated dual decomposition algorithm which we extend to handle specialized head automata and sequential head bigram models. Experiments in fourteen languages yield parsing speeds competitive to projective parsers, with state-of-the-art accuracies for the largest datasets (English, Czech, and German).

## 1 Introduction

**Dependency parsing** has become a prominent approach to syntax in the last few years, with increasingly fast and accurate models being devised (Kübler et al., 2009; Huang and Sagae, 2010; Zhang and Nivre, 2011; Rush and Petrov, 2012).

In projective parsing, the arcs in the dependency tree are constrained to be nested, and the problem of finding the best tree can be addressed with dynamic programming. This results in cubic-time decoders for arc-factored and sibling second-order models (Eisner, 1996; McDonald and Pereira, 2006), and quartic-time for grandparent models (Carreras, 2007) and third-order models (Koo and Collins, 2010). Recently, Rush and Petrov (2012) trained third-order parsers with vine pruning cascades, achieving runtimes only a small factor slower than first-order systems. Third-order features have also been included in transition systems (Zhang and Nivre, 2011) and graph-based parsers with cube-pruning (Zhang and McDonald, 2012).

Unfortunately, **non-projective dependency parsers** (appropriate for languages with a more flexible word order, such as Czech, Dutch, and German) lag behind these recent advances. The main obstacle is that non-projective parsing is NP-hard beyond arc-factored models (McDonald

and Satta, 2007). Approximate parsers have therefore been introduced, based on belief propagation (Smith and Eisner, 2008), dual decomposition (Koo et al., 2010), or multi-commodity flows (Martins et al., 2009, 2011). These are all instances of **turbo parsers**, as shown by Martins et al. (2010): the underlying approximations come from the fact that they run global inference in factor graphs ignoring loop effects. While this line of research has led to accuracy gains, none of these parsers use third-order contexts, and their speeds are well behind those of projective parsers.

This paper bridges the gap above by presenting the following contributions:

- We apply the third-order feature models of Koo and Collins (2010) to **non-projective** parsing.
- This extension is non-trivial since exact dynamic programming is not applicable. Instead, we adapt AD<sup>3</sup>, the dual decomposition algorithm proposed by Martins et al. (2011), to handle third-order features, by introducing specialized head automata.
- We make our parser substantially faster than the many-components approach of Martins et al. (2011). While AD<sup>3</sup> requires solving quadratic subproblems as an intermediate step, recent results (Martins et al., 2012) show that they can be addressed with the same oracles used in the sub-gradient method (Koo et al., 2010). This enables AD<sup>3</sup> to exploit combinatorial subproblems like the the head automata above.

Along with this paper, we provide a free distribution of our parsers, including training code.<sup>1</sup>

## 2 Dependency Parsing with AD<sup>3</sup>

**Dual decomposition** is a class of optimization techniques that tackle the dual of combinatorial

<sup>1</sup>Released as TurboParser 2.1, and publicly available at <http://www.ark.cs.cmu.edu/TurboParser>.

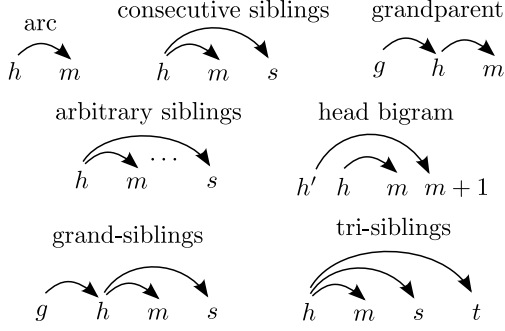


Figure 1: Parts considered in this paper. First-order models factor over arcs (Eisner, 1996; McDonald et al., 2005), and second-order models include also consecutive siblings and grandparents (Carreras, 2007). Our parsers add also *arbitrary* siblings (not necessarily consecutive) and head bigrams, as in Martins et al. (2011), in addition to third-order features for grand- and tri-siblings (Koo and Collins, 2010).

problems in a modular and extensible manner (Komodakis et al., 2007; Rush et al., 2010). In this paper, we employ **alternating directions dual decomposition** (AD<sup>3</sup>; Martins et al., 2011). Like the subgradient algorithm of Rush et al. (2010), AD<sup>3</sup> splits the original problem into **local subproblems**, and seeks an agreement on the overlapping variables. The difference is that the AD<sup>3</sup> subproblems have an additional *quadratic* term to accelerate consensus. Recent analysis (Martins et al., 2012) has shown that: (i) AD<sup>3</sup> converges at a faster rate,<sup>2</sup> and (ii) the quadratic subproblems can be solved using the same combinatorial machinery that is used in the subgradient algorithm. This opens the door for larger subproblems (such as the combination of trees and head automata in Koo et al., 2010) instead of a many-components approach (Martins et al., 2011), while still enjoying faster convergence.

## 2.1 Our Setup

Given a sentence with  $L$  words, to which we prepend a root symbol  $\$$ , let  $A := \{\langle h, m \rangle \mid h \in \{0, \dots, L\}, m \in \{1, \dots, L\}, h \neq m\}$  be the set of possible dependency arcs. We parameterize a dependency tree via an indicator vector  $\mathbf{u} := \langle u_a \rangle_{a \in A}$ , where  $u_a$  is 1 if the arc  $a$  is in the tree, and 0 otherwise, and we denote by  $\mathcal{Y} \subseteq \mathbb{R}^{|A|}$  the set of such vectors that are indicators of well-

<sup>2</sup>Concretely, AD<sup>3</sup> needs  $O(1/\epsilon)$  iterations to converge to a  $\epsilon$ -accurate solution, while subgradient needs  $O(1/\epsilon^2)$ .

formed trees. Let  $\{A_s\}_{s=1}^S$  be a cover of  $A$ , where each  $A_s \subseteq A$ . We assume that the score of a parse tree  $\mathbf{u} \in \mathcal{Y}$  decomposes as  $f(\mathbf{u}) := \sum_{s=1}^S f_s(\mathbf{z}_s)$ , where each  $\mathbf{z}_s := \langle z_{s,a} \rangle_{a \in A_s}$  is a “partial view” of  $\mathbf{u}$ , and each local score function  $f_s$  comes from a feature-based linear model.

Past work in dependency parsing considered either (i) a few “large” components, such as **trees** and **head automata** (Smith and Eisner, 2008; Koo et al., 2010), or (ii) many “small” components, coming from a multi-commodity flow formulation (Martins et al., 2009, 2011). Let  $\mathcal{Y}_s \subseteq \mathbb{R}^{|A_s|}$  denote the set of feasible realizations of  $\mathbf{z}_s$ , *i.e.*, those that are partial views of an actual parse tree. A tuple of views  $\langle \mathbf{z}_1, \dots, \mathbf{z}_S \rangle \in \prod_{s=1}^S \mathcal{Y}_s$  is said to be *globally consistent* if  $z_{s,a} = z_{s',a}$  holds for every  $a, s$  and  $s'$  such that  $a \in A_s \cap A_{s'}$ . We assume each parse  $\mathbf{u} \in \mathcal{Y}$  corresponds uniquely to a globally consistent tuple of views, and vice-versa. Following Martins et al. (2011), the problem of obtaining the best-scored tree can be written as follows:

$$\begin{aligned} & \text{maximize} && \sum_{s=1}^S f_s(\mathbf{z}_s) \\ & \text{w.r.t. } \mathbf{u} \in \mathbb{R}^{|A|}, && \mathbf{z}_s \in \mathcal{Y}_s, \forall s \\ & && \text{s.t. } z_{s,a} = u_a, \forall s, \forall a \in A_s, \end{aligned} \quad (1)$$

where the equality constraint ensures that the partial views “glue” together to form a coherent parse tree.<sup>3</sup>

## 2.2 Dual Decomposition and AD<sup>3</sup>

Dual decomposition methods dualize out the equality constraint in Eq. 1 by introducing **Lagrange multipliers**  $\lambda_{s,a}$ . In doing so, they solve a relaxation where the combinatorial sets  $\mathcal{Y}_s$  are replaced by their convex hulls  $\mathcal{Z}_s := \text{conv}(\mathcal{Y}_s)$ .<sup>4</sup> All that is necessary is the following assumption:

**Assumption 1** (Local-Max Oracle). *Every  $s \in \{1, \dots, S\}$  has an oracle that solves efficiently any instance of the following subproblem:*

$$\begin{aligned} & \text{maximize} && f_s(\mathbf{z}_s) + \sum_{a \in A_s} \lambda_{s,a} z_{s,a} \\ & \text{w.r.t. } \mathbf{z}_s \in \mathcal{Y}_s. \end{aligned} \quad (2)$$

Typically, Assumption 1 is met whenever the maximization of  $f_s$  over  $\mathcal{Y}_s$  is tractable, since the objective in Eq. 2 just adds a linear function to  $f_s$ .

<sup>3</sup>Note that any tuple  $\langle \mathbf{z}_1, \dots, \mathbf{z}_S \rangle \in \prod_{s=1}^S \mathcal{Y}_s$  satisfying the equality constraints will be globally consistent; this fact, due the assumptions above, will imply  $\mathbf{u} \in \mathcal{Y}$ .

<sup>4</sup>Let  $\Delta^{|\mathcal{Y}_s|} := \{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{Y}_s|} \mid \boldsymbol{\alpha} \geq \mathbf{0}, \sum_{\mathbf{y}_s \in \mathcal{Y}_s} \alpha_{\mathbf{y}_s} = 1\}$  be the probability simplex. The convex hull of  $\mathcal{Y}_s$  is the set  $\text{conv}(\mathcal{Y}_s) := \{\sum_{\mathbf{y}_s \in \mathcal{Y}_s} \alpha_{\mathbf{y}_s} \mathbf{y}_s \mid \boldsymbol{\alpha} \in \Delta^{|\mathcal{Y}_s|}\}$ . Its members represent marginal probabilities over the arcs in  $A_s$ .

The AD<sup>3</sup> algorithm (Martins et al., 2011) alternates among the following iterative updates:

- **z-updates**, which decouple over  $s = 1, \dots, S$ , and solve a penalized version of Eq. 2:

$$\begin{aligned} \mathbf{z}_s^{(t+1)} &:= \arg \max_{\mathbf{z}_s \in \mathcal{Z}_s} f_s(\mathbf{z}_s) + \sum_{a \in A_s} \lambda_{s,a}^{(t)} z_{s,a} \\ &\quad - \frac{\rho}{2} \sum_{a \in A_s} (z_{s,a} - u_a^{(t)})^2. \end{aligned} \quad (3)$$

Above,  $\rho$  is a constant and the quadratic term penalizes deviations from the current global solution (stored in  $\mathbf{u}^{(t)}$ ).<sup>5</sup> We will see (Prop. 2) that this problem can be solved iteratively using only the Local-Max Oracle (Eq. 2).

- **u-updates**, a simple averaging operation:

$$u_a^{(t+1)} := \frac{1}{\{|s : a \in A_s\}} \sum_{s : a \in A_s} z_{s,a}^{(t+1)}. \quad (4)$$

- **$\lambda$ -updates**, where the Lagrange multipliers are adjusted to penalize disagreements:

$$\lambda_{s,a}^{(t+1)} := \lambda_{s,a}^{(t)} - \rho(z_{s,a}^{(t+1)} - u_a^{(t+1)}). \quad (5)$$

In sum, the only difference between AD<sup>3</sup> and the subgradient method is in the  $\mathbf{z}$ -updates, which in AD<sup>3</sup> require solving a quadratic problem. While closed-form solutions have been developed for some specialized components (Martins et al., 2011), this problem is in general more difficult than the one arising in the subgradient algorithm. However, the following result, proved in Martins et al. (2012), allows to expand the scope of AD<sup>3</sup> to any problem which satisfies Assumption 1.

**Proposition 2.** *The problem in Eq. 3 admits a solution  $\mathbf{z}_s^*$  which is spanned by a sparse basis  $\mathcal{W} \subseteq \mathcal{Y}_s$  with cardinality at most  $|\mathcal{W}| \leq O(|A_s|)$ . In other words, there is a distribution  $\alpha$  with support in  $\mathcal{W}$  such that  $\mathbf{z}_s^* = \sum_{\mathbf{y}_s \in \mathcal{W}} \alpha_{\mathbf{y}_s} \mathbf{y}_s$ .<sup>6</sup>*

Prop. 2 has motivated an active set algorithm (Martins et al., 2012) that maintains an estimate of  $\mathcal{W}$  by iteratively adding and removing elements computed through the oracle in Eq. 2.<sup>7</sup> Typically, very few iterations are necessary and great speed-ups are achieved by warm-starting  $\mathcal{W}$  with the active set computed in the previous AD<sup>3</sup> iteration. This has a huge impact in practice and is crucial to obtain the fast runtimes in §4 (see Fig. 2).

<sup>5</sup>In our experiments (§4), we set  $\rho = 0.05$ .

<sup>6</sup>Note that  $|\mathcal{Y}_s| = O(2^{|A_s|})$  in general. What Prop. 2 tells us is that the solution of Eq. 3 can be represented as a distribution over  $\mathcal{Y}_s$  with a very sparse support.

<sup>7</sup>The algorithm is a specialization of Nocedal and Wright (1999), §16.4, which effectively exploits the sparse representation of  $\mathbf{z}_s^*$ . For details, see Martins et al. (2012).

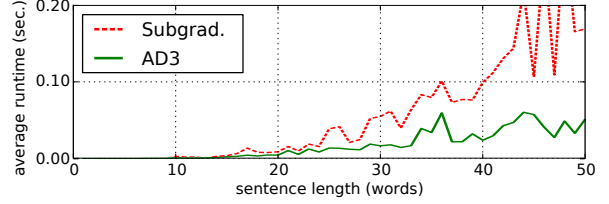


Figure 2: Comparison between AD<sup>3</sup> and subgradient. We show averaged runtimes in PTB §22 as a function of the sentence length. For subgradient, we chose for each sentence the most favorable stepsize in  $\{0.001, 0.01, 0.1, 1\}$ .

### 3 Solving the Subproblems

We next describe the actual components used in our third-order parsers.

**Tree component.** We use an arc-factored score function (McDonald et al., 2005):  $f^{\text{TREE}}(\mathbf{z}) = \sum_{m=1}^L \sigma_{\text{ARC}}(\pi(m), m)$ , where  $\pi(m)$  is the parent of the  $m$ th word according to the parse tree  $\mathbf{z}$ , and  $\sigma_{\text{ARC}}(h, m)$  is the score of an individual arc. The parse tree that maximizes this function can be found in time  $O(L^3)$  via the Chu-Liu-Edmonds' algorithm (Chu and Liu, 1965; Edmonds, 1967).<sup>8</sup>

**Grand-sibling head automata.** Let  $A_h^{\text{in}}$  and  $A_h^{\text{out}}$  denote respectively the sets of *incoming* and *outgoing* candidate arcs for the  $h$ th word, where the latter subdivides into arcs pointing to the right,  $A_{h,\rightarrow}^{\text{out}}$ , and to the left,  $A_{h,\leftarrow}^{\text{out}}$ . Define the sets  $A_{h,\rightarrow}^{\text{GSIB}} = A_h^{\text{in}} \cup A_{h,\rightarrow}^{\text{out}}$  and  $A_{h,\leftarrow}^{\text{GSIB}} = A_h^{\text{in}} \cup A_{h,\leftarrow}^{\text{out}}$ . We describe right-side grand-sibling head automata; their left-side counterparts are analogous. For each head word  $h$  in the parse tree  $\mathbf{z}$ , define  $g := \pi(h)$ , and let  $\langle m_0, m_1, \dots, m_{p+1} \rangle$  be the sequence of right modifiers of  $h$ , with  $m_0 = \text{START}$  and  $m_{p+1} = \text{END}$ . Then, we have the following grand-sibling component:

$$\begin{aligned} f_{h,\rightarrow}^{\text{GSIB}}(\mathbf{z}|_{A_{h,\rightarrow}^{\text{GSIB}}}) &= \sum_{k=1}^{p+1} (\sigma_{\text{SIB}}(h, m_{k-1}, m_k) \\ &\quad + \sigma_{\text{GP}}(g, h, m_k) + \sigma_{\text{GSIB}}(g, h, m_{k-1}, m_k)), \end{aligned}$$

where we use the shorthand  $\mathbf{z}|_B$  to denote the subvector of  $\mathbf{z}$  indexed by the arcs in  $B \subseteq A$ . Note that this score function absorbs grandparent and consecutive sibling scores, in addition to the grand-sibling scores.<sup>9</sup> For each  $h$ ,  $f_{h,\rightarrow}^{\text{GSIB}}$  can be

<sup>8</sup>In fact, there is an asymptotically faster  $O(L^2)$  algorithm (Tarjan, 1977). Moreover, if the set of possible arcs is reduced to a subset  $B \subseteq A$  (via pruning), then the fastest known algorithm (Gabow et al., 1986) runs in  $O(|B| + L \log L)$  time.

<sup>9</sup>Koo et al. (2010) used an identical automaton for their second-order model, but leaving out the grand-sibling scores.

|      | No pruning | $ A_m^{\text{in}}  \leq K$ | same, + $ A_h^{\text{out}}  \leq J$ |
|------|------------|----------------------------|-------------------------------------|
| TREE | $O(L^2)$   | $O(KL + L \log L)$         | $O(KL + L \log L)$                  |
| GSIB | $O(L^4)$   | $O(K^2 L^2)$               | $O(JK^2 L)$                         |
| TSIB | $O(L^4)$   | $O(KL^3)$                  | $O(J^2 KL)$                         |
| SEQ  | $O(L^3)$   | $O(K^2 L)$                 | $O(K^2 L)$                          |
| ASIB | $O(L^3)$   | $O(KL^2)$                  | $O(JKL)$                            |

Table 1: Theoretical runtimes of each subproblem without pruning, limiting the number of candidate heads, and limiting (in addition) the number of modifiers. Note the  $O(L \log L)$  total runtime per  $\text{AD}^3$  iteration in the latter case.

maximized in time  $O(L^3)$  with dynamic programming, yielding  $O(L^4)$  total runtime.

**Tri-sibling head automata.** In addition, we define left and right-side tri-sibling head automata that remember the previous two modifiers of a head word. This corresponds to the following component function (for the right-side case):

$$f_{h,\rightarrow}^{\text{TSIB}}(z|_{A_{h,\rightarrow}^{\text{out}}}) = \sum_{k=2}^{p+1} \sigma_{\text{TSIB}}(h, m_{k-2}, m_{k-1}, m_k).$$

Again, each of these functions can be maximized in time  $O(L^3)$ , yielding  $O(L^4)$  runtime.

**Sequential head bigram model.** Head bigrams can be captured with a simple sequence model:

$$f^{\text{SEQ}}(z) = \sum_{m=2}^L \sigma_{\text{HB}}(m, \pi(m), \pi(m-1)).$$

Each score  $\sigma_{\text{HB}}(m, h, h')$  is obtained via features that look at the heads of consecutive words (as in Martins et al. (2011)). This function can be maximized in time  $O(L^3)$  with the Viterbi algorithm.

**Arbitrary siblings.** We handle arbitrary siblings as in Martins et al. (2011), defining  $O(L^3)$  component functions of the form  $f_{h,m,s}^{\text{ASIB}}(z_{\langle h,m \rangle}, z_{\langle h,s \rangle}) = \sigma_{\text{ASIB}}(h, m, s)$ . In this case, the quadratic problem in Eq. 3 can be solved directly in constant time.

Tab. 1 details the time complexities of each subproblem. Without pruning, each iteration of  $\text{AD}^3$  has  $O(L^4)$  runtime. With a simple strategy that limits the number of candidate heads per word to a constant  $K$ , this drops to cubic time.<sup>10</sup> Further speed-ups are possible with more pruning: by limiting the number of possible modifiers to a constant  $J$ , the runtime would reduce to  $O(L \log L)$ .

<sup>10</sup>In our experiments, we employed this strategy with  $K = 10$ , by pruning with a first-order probabilistic model. Following Koo and Collins (2010), for each word  $m$ , we also pruned away incoming arcs  $\langle h, m \rangle$  with posterior probability less than 0.0001 times the probability of the most likely head.

|  | UAS          | Tok/sec          |
|--|--------------|------------------|
| PTB-YM §22, 1st ord                    | 91.38        | 4,063            |
| PTB-YM §22, 2nd ord                    | 93.15        | 1,338            |
| PTB-YM §22, 2nd ord, +ASIB, +HB        | 93.28        | 1,018            |
| PTB-YM §22, 3rd ord                    | 93.29        | 709              |
| PTB-YM §22, 3rd ord, gold tags         | 94.01        | 722              |
| <b>This work (PTB-YM §23, 3rd ord)</b> | <b>93.07</b> | 735              |
| Koo et al. (2010)                      | 92.46        | 112 <sup>†</sup> |
| Huang and Sagae (2010)                 | 92.1–        | 587 <sup>†</sup> |
| Zhang and Nivre (2011)                 | 92.9–        | 680 <sup>†</sup> |
| Martins et al. (2011)                  | 92.53        | 66 <sup>†</sup>  |
| Zhang and McDonald (2012)              | 93.06        | 220              |
| <b>This work (PTB-S §23, 3rd ord)</b>  | <b>92.82</b> | 604              |
| Rush and Petrov (2012)                 | 92.7–        | 4,460            |

Table 2: Results for the projective English dataset. We report unlabeled attachment scores (UAS) ignoring punctuation, and parsing speeds in tokens per second. Our speeds include the time necessary for pruning, evaluating features, and decoding, as measured on a Intel Core i7 processor @3.4 GHz. The others are speeds reported in the cited papers; those marked with <sup>†</sup> were converted from times per sentence.

## 4 Experiments

We first evaluated our non-projective parser in a *projective* English dataset, to see how its speed and accuracy compares with recent projective parsers, which can take advantage of dynamic programming. To this end, we converted the Penn Treebank to dependencies through (i) the head rules of Yamada and Matsumoto (2003) (PTB-YM) and (ii) basic dependencies from the Stanford parser 2.0.5 (PTB-S).<sup>11</sup> We trained by running 10 epochs of cost-augmented MIRA (Crammer et al., 2006). To ensure valid parse trees at test time, we rounded fractional solutions as in Martins et al. (2009)—yet, solutions were integral  $\approx 95\%$  of the time.

Tab. 2 shows the results in the dev-set (top block) and in the test-set (two bottom blocks). In the dev-set, we see consistent gains when more expressive features are added, the best accuracies being achieved with the full third-order model; this comes at the cost of a 6-fold drop in runtime compared with a first-order model. By looking at the two bottom blocks, we observe that our parser has slightly better accuracies than recent projective parsers, with comparable speed levels (with the exception of the highly optimized vine cascade approach of Rush and Petrov, 2012).

<sup>11</sup>We train on sections §02–21, use §22 as validation data, and test on §23. We trained a simple 2nd-order tagger with 10-fold jackknifing to obtain automatic part-of-speech tags for §22–23, with accuracies 97.2% and 96.9%, respectively.

|            | First Ord. |         | Sec. Ord. |         | Third Ord.   |         | Best published UAS |         |      | RP12 |         | ZM12         |
|------------|------------|---------|-----------|---------|--------------|---------|--------------------|---------|------|------|---------|--------------|
|            | UAS        | Tok/sec | UAS       | Tok/sec | UAS          | Tok/sec | UAS                | Tok/sec |      | UAS  | Tok/sec | UAS          |
| Arabic     | 77.23      | 2,481   | 78.50     | 388     | 79.64        | 197     | 81.12              | -       | Ma11 | -    | -       | -            |
| Bulgarian  | 91.76      | 5,678   | 92.82     | 2,049   | 93.10        | 1,273   | 93.50              | -       | Ma11 | 91.9 | 3,980   | 93.08        |
| Chinese    | 88.49      | 18,094  | 90.14     | 4,284   | 89.98        | 2,592   | 91.89              | -       | Ma10 | 90.9 | 7,800   | -            |
| Czech      | 87.66      | 1,840   | 90.00     | 751     | <b>90.32</b> | 501     | 89.46              | -       | Ma11 | -    | -       | -            |
| Danish     | 89.42      | 4,110   | 91.20     | 1,053   | 91.48        | 650     | 91.86              | -       | Ma11 | -    | -       | -            |
| Dutch      | 83.61      | 3,884   | 86.37     | 1,294   | <b>86.19</b> | 599     | 85.81              | 121     | Ko10 | -    | -       | -            |
| German     | 90.52      | 5,331   | 91.85     | 1,788   | <b>92.41</b> | 965     | 91.89              | -       | Ma11 | 90.8 | 2,880   | 91.35        |
| English    | 91.21      | 3,127   | 93.03     | 1,317   | <b>93.22</b> | 785     | 92.68              | -       | Ma11 | -    | -       | -            |
| Japanese   | 92.78      | 23,895  | 93.14     | 5,660   | 93.52        | 2,996   | 93.72              | -       | Ma11 | 92.3 | 8,600   | 93.24        |
| Portuguese | 91.14      | 4,273   | 92.71     | 1,316   | 92.69        | 740     | 93.03              | 79      | Ko10 | 91.5 | 2,900   | 91.69        |
| Slovene    | 82.81      | 4,315   | 85.21     | 722     | 86.01        | 366     | 86.95              | -       | Ma11 | -    | -       | -            |
| Spanish    | 83.61      | 4,347   | 84.97     | 623     | 85.59        | 318     | 87.48              | -       | ZM12 | -    | -       | <b>87.48</b> |
| Swedish    | 89.36      | 5,622   | 90.98     | 1,387   | 91.14        | 684     | 91.44              | -       | ZM12 | 90.1 | 5,320   | <b>91.44</b> |
| Turkish    | 75.98      | 6,418   | 76.50     | 1,721   | 76.90        | 793     | 77.55              | 258     | Ko10 | -    | -       | -            |

Table 3: Results for the CoNLL-2006 datasets and the *non-projective* English dataset of CoNLL-2008. “Best Published UAS” includes the most accurate parsers among Nivre et al. (2006), McDonald et al. (2006), Martins et al. (2010, 2011), Koo et al. (2010), Rush and Petrov (2012), Zhang and McDonald (2012). The last two are shown separately in the rightmost columns.

In our second experiment (Tab. 3), we used 14 datasets, most of which are non-projective, from the CoNLL 2006 and 2008 shared tasks (Buchholz and Marsi, 2006; Surdeanu et al., 2008). Our third-order model achieved the best reported scores for English, Czech, German, and Dutch—which includes the three largest datasets and the ones with the most non-projective dependencies—and is on par with the state of the art for the remaining languages. To our knowledge, the speeds are the highest reported among higher-order non-projective parsers, and only about 3–4 times slower than the vine parser of Rush and Petrov (2012), which has lower accuracies.

## 5 Conclusions

We presented new third-order non-projective parsers which are both fast and accurate. We decoded with AD<sup>3</sup>, an accelerated dual decomposition algorithm which we adapted to handle large components, including specialized head automata for the third-order features, and a sequence model for head bigrams. Results are above the state of the art for large datasets and non-projective languages. In the hope that other researchers may find our implementation useful or are willing to contribute with further improvements, we made our parsers publicly available as open source software.

## Acknowledgments

We thank all reviewers for their insightful comments and Lingpeng Kong for help in converting the Penn Treebank to Stanford dependencies. This

work was partially supported by the EU/FEDER programme, QREN/POR Lisboa (Portugal), under the Intelligo project (contract 2012/24803), by a FCT grant PTDC/EEI-SII/2312/2012, and by NSF grant IIS-1054319.

## References

- S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *International Conference on Natural Language Learning*.
- X. Carreras. 2007. Experiments with a higher-order projective dependency parser. In *International Conference on Natural Language Learning*.
- Y. J. Chu and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.
- K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.
- J. Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.
- J. M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of International Conference on Computational Linguistics*, pages 340–345.
- H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan. 1986. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6(2):109–122.

- L. Huang and K. Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086.
- N. Komodakis, N. Paragios, and G. Tziritas. 2007. MRF optimization via dual decomposition: Message-passing revisited. In *Proc. of International Conference on Computer Vision*.
- T. Koo and M. Collins. 2010. Efficient third-order dependency parsers. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 1–11.
- T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proc. of Empirical Methods for Natural Language Processing*.
- S. Kübler, R. McDonald, and J. Nivre. 2009. *Dependency parsing*. Morgan & Claypool Publishers.
- A. F. T. Martins, N. A. Smith, and E. P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proc. of Annual Meeting of the Association for Computational Linguistics*.
- A. F. T. Martins, N. A. Smith, E. P. Xing, M. A. T. Figueiredo, and P. M. Q. Aguiar. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Proc. of Empirical Methods for Natural Language Processing*.
- A. F. T. Martins, N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo. 2011. Dual decomposition with many overlapping components. In *Proc. of Empirical Methods for Natural Language Processing*.
- A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing. 2012. Alternating directions dual decomposition. Arxiv preprint arXiv:1212.6550.
- R. T. McDonald and F. C. N. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of Annual Meeting of the European Chapter of the Association for Computational Linguistics*.
- R. McDonald and G. Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proc. of International Conference on Parsing Technologies*.
- R. T. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of Empirical Methods for Natural Language Processing*.
- R. McDonald, K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proc. of International Conference on Natural Language Learning*.
- J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Procs. of International Conference on Natural Language Learning*.
- J. Nocedal and S. J. Wright. 1999. *Numerical optimization*. Springer-Verlag.
- Alexander M Rush and Slav Petrov. 2012. Vine pruning for efficient multi-pass dependency parsing. In *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics*.
- A. Rush, D. Sontag, M. Collins, and T. Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proc. of Empirical Methods for Natural Language Processing*.
- D. Smith and J. Eisner. 2008. Dependency parsing by belief propagation. In *Proc. of Empirical Methods for Natural Language Processing*.
- M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. *Proc. of International Conference on Natural Language Learning*.
- R.E. Tarjan. 1977. Finding optimum branchings. *Networks*, 7(1):25–36.
- H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proc. of International Conference on Parsing Technologies*.
- H. Zhang and R. McDonald. 2012. Generalized higher-order dependency parsing with cube pruning. In *Proc. of Empirical Methods in Natural Language Processing*.
- Y. Zhang and J. Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

# A Lattice-based Framework for Joint Chinese Word Segmentation, POS Tagging and Parsing

Zhiguo Wang<sup>1</sup>, Chengqing Zong<sup>1</sup> and Nianwen Xue<sup>2</sup>

<sup>1</sup>National Laboratory of Pattern Recognition,

Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190

<sup>2</sup>Computer Science Department, Brandeis University, Waltham, MA 02452

{zgwang, cqzong}@nlpr.ia.ac.cn    xuen@brandeis.edu

## Abstract

For the cascaded task of Chinese word segmentation, POS tagging and parsing, the pipeline approach suffers from error propagation while the joint learning approach suffers from inefficient decoding due to the large combined search space. In this paper, we present a novel lattice-based framework in which a Chinese sentence is first segmented into a word lattice, and then a lattice-based POS tagger and a lattice-based parser are used to process the lattice from two different viewpoints: sequential POS tagging and hierarchical tree building. A strategy is designed to exploit the complementary strengths of the tagger and parser, and encourage them to predict agreed structures. Experimental results on Chinese Treebank show that our lattice-based framework significantly improves the accuracy of the three sub-tasks.

## 1 Introduction

Previous work on syntactic parsing generally assumes a processing pipeline where an input sentence is first tokenized, POS-tagged and then parsed (Collins, 1999; Charniak, 2000; Petrov and Klein, 2007). This approach works well for languages like English where automatic tokenization and POS tagging can be performed with high accuracy without the guidance of the high-level syntactic structure. Such an approach, however, is not optimal for languages like Chinese where there are no natural delimiters for word boundaries, and word segmentation (or tokenization) is a non-trivial research problem by itself. Errors in word segmentation would propagate to later processing stages such as POS tagging and syntactic parsing. More importantly, Chinese is a language that lacks the morphological clues that help determine the POS tag of a word. For example, 调查 (“investigate/investigation”) can either be a verb (“investigate”) or a noun (“investigation”), and there is no morphological variation between its verbal form and nominal form.

This contributes to the relatively low accuracy (95% or below) in Chinese POS tagging when evaluated as a stand-alone task (Sun and Uszkoreit, 2012), and the noun/verb ambiguity is a major source of error.

More recently, joint inference approaches have been proposed to address the shortcomings of the pipeline approach. Qian and Liu (2012) proposed a joint inference approach where syntactic parsing can provide feedback to word segmentation and POS tagging and showed that the joint inference approach leads to improvements in all three sub-tasks. However, a major challenge for joint inference approach is that the large combined search space makes efficient decoding and parameter estimation very hard.

In this paper, we present a novel lattice-based framework for Chinese. An input Chinese sentence is first segmented into a word lattice, which is a compact representation of a small set of high-quality word segmentations. Then, a lattice-based POS tagger and a lattice-based parser are used to process the word lattice from two different viewpoints. We next employ the dual decomposition method to exploit the complementary strengths of the tagger and parser, and encourage them to predict agreed structures. Experimental results show that our lattice-based framework significantly improves the accuracies of the three sub-tasks

## 2 The Lattice-based Framework

Figure 1 gives the organization of the framework. There are four types of linguistic structures: a Chinese sentence, the word lattice, tagged word sequence and parse tree of the Chinese sentence. An example for each structure is provided in Figure 2. We can see that the terminals and pre-terminals of a parse tree constitute a tagged word sequence. Therefore, we define a comparator between a tagged word sequence and a parse tree: if they contain the same word sequence and POS tags, they are equal, otherwise unequal.

Figure 1 also shows the workflow of the framework. First, the Chinese sentence is segmented into a word lattice using the word segmentation system. Then the word lattice is fed into the lattice-based POS tagger to produce a tagged word sequence  $S$  and into the lattice-based parser to separately produce a parse tree  $T$ . We then compare  $S$  with  $T$  to see whether they are equal. If they are equal, we output  $T$  as the final result. Otherwise, the guidance generator generates some guidance orders based on the difference between  $S$  and  $T$ , and guides the tagger and the parser to process the lattice again. This procedure may iterate many times until the tagger and parser predict equal structures.

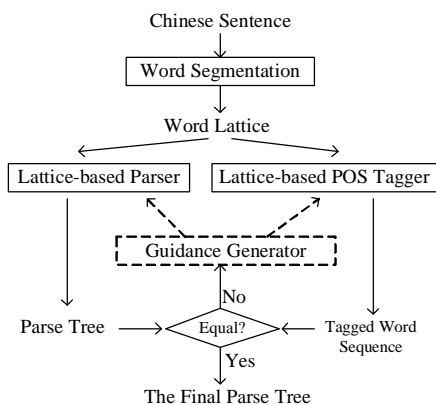
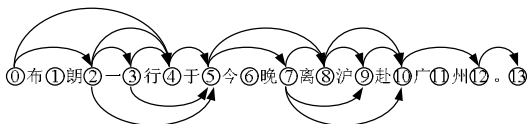


Figure 1: The lattice-based framework.

布朗一行于今晚离沪赴广州。  
Brown's group will leave Shanghai to Guangzhou tonight.  
(a) Chinese Sentence



NR — NN — P — NT — P — NR — VV — NR — PU  
布朗 一行 于 今晚 离 沪 赴 广州  
Brown group in tonight leave Shanghai go Guangzhou .

(c) Tagged Word Sequence

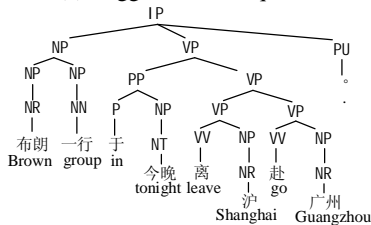


Figure 2: Linguistic structure examples.

The motivation to design such a framework is as follows. First, state-of-the-art word segmentation systems can now perform with high accuracy. We can easily get an F1 score greater than 96%, and an oracle (upper bound) F1 score greater than 99% for the word lattice (Jiang et

al., 2008). Therefore, a word lattice provides us a good enough search space to allow sufficient interaction among word segmentation, POS tagging and parsing systems. Second, both the lattice-based POS tagger and the lattice-based parser can select word segmentation from the word lattice and predict POS tags, but they do so from two different perspectives. The lattice-based POS tagger looks at a path in a word lattice as a sequence and performs sequence labeling based on linear local context, while the lattice-based parser builds the parse trees in a hierarchical manner. They have different strengths with regard to word segmentation and POS tagging. We hypothesize that exploring the complementary strengths of the tagger and parser would improve each of the sub-tasks.

We build a character-based model (Xue, 2003) for the word segmentation system, and treat segmentation as a sequence labeling task, where each Chinese character is labeled with a tag. We use the tag set provided in Wang et al. (2011) and use the same feature templates. We use the Maximum Entropy (ME) model to estimate the feature weights. To get a word lattice, we first generate N-best word segmentation results, and then compact the N-best lists into a word lattice by collapsing all the identical words into one edge. We also assign a probability to each edge, which is calculated by multiplying the tagging probabilities of each character in the word.

The goal of the lattice-based POS tagger is to predict a tagged word sequence  $S$  for an input word lattice  $L$ :

$$\hat{S} = \underset{S \in \text{cand}(L)}{\text{argmax}} \mathbf{w} \cdot \mathbf{f}(S)$$

where  $\text{cand}(L)$  represents the set of all possible tagged word sequences derived from the word lattice  $L$ .  $\mathbf{f}(S)$  is used to map  $S$  onto a global feature vector, and  $\mathbf{w}$  is the corresponding weight vector. We use the same non-local feature templates used in Jiang et al. (2008) and a similar decoding algorithm. We use the perceptron algorithm (Collins, 2002) for parameter estimation.

Goldberg and Elhadad (2011) proposed a lattice-based parser for Heberw based on the PCFG-LA model (Matsuzaki et al., 2005). We adopted their approach, but found the un-weighted word lattice their parser takes as input to be ineffective for our Chinese experiments. Instead, we use a weighted lattice as input and weigh each edge in the lattice with the word probability. In our model, each syntactic category  $A$  is split into multiple subcategories  $A[x]$  by labeling a latent annotation  $x$ . Then, a parse tree



$T$  is refined into  $T[\mathbf{X}]$ , where  $\mathbf{X}$  is the latent annotation vector for all non-terminals in  $T$ . The probability of  $T[\mathbf{X}]$  is calculated as:

$$p(T[\mathbf{X}]) = \prod p(A[x] \rightarrow B[y]C[z]) \times \prod p(D[x] \rightarrow w) \times \prod p(w)$$

where the three terms are products of all syntactic rule probabilities, lexical rule probabilities and word probabilities in  $T[\mathbf{X}]$  respectively.

### 3 Combined Optimization Between The Lattice-based POS Tagger and The Lattice-based Parser

We first define some variables to make it easier to compare a tagged word sequence  $S$  with a parse tree  $T$ . We define  $\mathcal{P}$  as the set of all POS tags. For  $S$ , we define  $s(i, j, p)=1$  if  $S$  contains a POS tag  $p \in \mathcal{P}$  spanning from the  $i$ -th character to the  $j$ -th character, otherwise  $s(i, j, p) = 0$ . We also define  $s(i, j, \#) = 1$  if  $S$  contains the word spanning from the  $i$ -th character to the  $j$ -th character, otherwise  $s(i, j, \#) = 0$ . Similarly, for  $T$ , we define  $t(i, j, p)=1$  if  $T$  contains a POS tag  $p \in \mathcal{P}$  spanning from the  $i$ -th character to the  $j$ -th character, otherwise  $t(i, j, p) = 0$ . We also define  $t(i, j, \#) = 1$  if  $T$  contains the word spanning from the  $i$ -th character to the  $j$ -th character, otherwise  $t(i, j, \#) = 0$ . Therefore,  $S$  and  $T$  are equal, only if  $s(i, j, p) = t(i, j, p)$  for all  $i \in [0, n]$ ,  $j \in [i + 1, n]$  and  $p \in \mathcal{P} \cup \#$ , otherwise unequal.

Our framework expects the tagger and the parser to predict equal structures and we formulate it as a constraint optimization problem:

$$(\hat{S}, \hat{T}) = \underset{S, T}{\operatorname{argmax}} f_1(S) + f_2(T)$$

Such that for all  $i \in [0, n]$ ,  $j \in [i + 1, n]$  and  $p \in \mathcal{P} \cup \#$ :

$$s(i, j, p) = t(i, j, p)$$

where  $f_1(S) = \mathbf{w} \cdot \mathbf{f}(S)$  is a scoring function from the viewpoint of the lattice-based POS tagger, and  $f_2(T) = \log p(T)$  is a scoring function from the viewpoint of the lattice-based parser.

The dual decomposition (a special case of Lagrangian relaxation) method introduced in Komodakis et al. (2007) is suitable for this problem. Using this method, we solve the primal constraint optimization problem by optimizing the dual problem. First, we introduce a vector of Lagrange multipliers  $\mu(i, j, p)$  for each equality constraint. Then, the Lagrangian is formulated as:

$$L(S, T, \mu) = f_1(S) + f_2(T) + \sum_{i, j, p} \mu(i, j, p)(s(i, j, p) - t(i, j, p))$$

By grouping the terms that depend on  $S$  and  $T$ , we rewrite the Lagrangian as

$$L(S, T, \mu) = \left( f_1(S) + \sum_{i, j, p} \mu(i, j, p)s(i, j, p) \right) + \left( f_2(T) - \sum_{i, j, p} \mu(i, j, p)t(i, j, p) \right)$$

Then, the dual objective is

$$L(\mu) = \max_{S, T} L(S, T, \mu) = \max_S \left( f_1(S) + \sum_{i, j, p} \mu(i, j, p)s(i, j, p) \right) + \max_T \left( f_2(T) - \sum_{i, j, p} \mu(i, j, p)t(i, j, p) \right)$$

The dual problem is to find  $\min_{\mu} L(\mu)$ .

We use the subgradient method (Boyd et al., 2003) to minimize the dual. Following Rush et al. (2010), we define the subgradient of  $L(\mu)$  as:

$$\gamma(i, j, p) = s(i, j, p) - t(i, j, p) \text{ for all } (i, j, p)$$

Then, adjust  $\mu(i, j, p)$  as follows:

$$\mu'(i, j, p) = \mu(i, j, p) - \delta(s(i, j, p) - t(i, j, p))$$

where  $\delta > 0$  is a step size.

---

#### Algorithm 1: Combined Optimization

---

- 1: Set  $\mu^{(0)}(i, j, p)=0$ , for all  $\mu(i, j, p)$
  - 2: **For**  $k=1$  **to**  $K$
  - 3:  $\hat{S}^{(k)} \leftarrow \operatorname{argmax}_S (f_1(S) + \sum_{i, j, p} \mu^{(k-1)}(i, j, p)s(i, j, p))$
  - 4:  $\hat{T}^{(k)} \leftarrow \operatorname{argmax}_T (f_2(T) - \sum_{i, j, p} \mu^{(k-1)}(i, j, p)t(i, j, p))$
  - 5: **If**  $s^{(k)}(i, j, p) = t^{(k)}(i, j, p)$  for all  $(i, j, p)$
  - 6: **Return**  $(\hat{S}^{(k)}, \hat{T}^{(k)})$
  - 7: **Else**
  - 8:  $\mu^{(k)}(i, j, p) = \mu^{(k-1)}(i, j, p) - \delta(s^{(k)}(i, j, p) - t^{(k)}(i, j, p))$
- 

Algorithm 1 presents the subgradient method to solve the dual problem. The algorithm initializes the Lagrange multiplier values with 0 (line 1) and then iterates many times. In each iteration, the algorithm finds the best  $\hat{S}^{(k)}$  and  $\hat{T}^{(k)}$  by running the lattice-based POS tagger (line 3) and the lattice-based parser (line 4). If  $\hat{S}^{(k)}$  and  $\hat{T}^{(k)}$  share the same tagged word sequence (line 5), then the algorithm returns the solution (line 6). Otherwise, the algorithm adjusts the Lagrange multiplier values based on the differences between  $\hat{S}^{(k)}$  and  $\hat{T}^{(k)}$  (line 8). A crucial point is that the **argmax** problems in line 3 and line 4 can be solved efficiently using the original decoding algorithms, because the Lagrange multiplier can be regarded as adjustments for lexical rule probabilities and word probabilities.

## 4 Experiments

We conduct experiments on the Chinese Treebank Version 5.0 and use the standard data split

(Petrov and Klein, 2007). The traditional evaluation metrics for POS tagging and parsing are not suitable for the joint task. Following with Qian and Liu (2012), we redefine *precision* and *recall* by computing the span of a constituent based on character offsets rather than word offsets.

#### 4.1 Performance of the Basic Sub-systems

We train the word segmentation system with 100 iterations of the Maximum Entropy model using the OpenNLP toolkit. Table 1 shows the performance. It shows that our word segmentation system is comparable with the state-of-the-art systems and the upper bound F1 score of the word lattice exceeds 99.6%. This indicates that our word segmentation system can provide a good search space for the lattice-based POS tagger and the lattice-based parser.

|                           | P     | R     | F     |
|---------------------------|-------|-------|-------|
| (Kruengkrai et al., 2009) | 97.46 | 98.29 | 97.87 |
| (Zhang and Clark, 2010)   | -     | -     | 97.78 |
| (Qian and Liu, 2012)      | 97.45 | 98.24 | 97.85 |
| (Sun, 2011)               | -     | -     | 98.17 |
| Our Word Seg. System      | 96.97 | 98.06 | 97.52 |
| Word Lattice Upper Bound  | 99.55 | 99.75 | 99.65 |

Table 1: Word segmentation evaluation.

To train the lattice-based POS tagger, we generate the word lattice for each sentence in the training set using cross validation approach. We divide the entire training set into 18 folds on average (each fold contains 1,000 sentences). For each fold, we segment each sentence in the fold into a word lattice by compacting 20-best segmentation list produced with a model trained on the other 17 folds. Then, we train the lattice-based POS tagger with 20 iterations of the average perceptron algorithm. Table 2 presents the joint word segmentation and POS tagging performance and shows that our lattice-based POS tagger obtains results that are comparable with state-of-the-art systems.

|                           | P     | R     | F     |
|---------------------------|-------|-------|-------|
| (Kruengkrai et al., 2009) | 93.28 | 94.07 | 93.67 |
| (Zhang and Clark, 2010)   | -     | -     | 93.67 |
| (Qian and Liu, 2012)      | 93.1  | 93.96 | 93.53 |
| (Sun, 2011)               | -     | -     | 94.02 |
| Lattice-based POS tagger  | 93.64 | 93.87 | 93.75 |

Table 2: POS tagging evaluation.

We implement the lattice-based parser by modifying the Berkeley Parser, and train it with 5 iterations of the split-merge-smooth strategy (Petrov et al., 2006). Table 3 shows the performance, where the “Pipeline Parser” represents the system taking one-best segmentation result

from our word segmentation system as input and “Lattice-based Parser” represents the system taking the compacted word lattice as input. We find the lattice-based parser gets better performance than the pipeline system among all three sub-tasks.

|                      |       | P     | R     | F     |
|----------------------|-------|-------|-------|-------|
| Pipeline Parser      | Seg.  | 96.97 | 98.06 | 97.52 |
|                      | POS   | 92.01 | 93.04 | 92.52 |
|                      | Parse | 80.86 | 81.47 | 81.17 |
| Lattice-based Parser | Seg.  | 97.73 | 97.66 | 97.70 |
|                      | POS   | 93.24 | 93.18 | 93.21 |
|                      | Parse | 81.83 | 81.71 | 81.77 |

Table 3: Parsing evaluation.

#### 4.2 Performance of the Framework

For the lattice-based framework, we set the maximum iteration in Algorithm 1 as  $K = 20$ . The step size  $\delta$  is tuned on the development set and empirically set to be 0.8. Table 4 shows the parsing performance on the test set. It shows that the lattice-based framework achieves improvement over the lattice-based parser alone among all three sub-tasks: 0.16 points for word segmentation, 1.19 points for POS tagging and 1.65 points for parsing. It also outperforms the lattice-based POS tagger by 0.65 points on POS tagging accuracy. Our lattice-based framework also improves over the best joint inference parsing system (Qian and Liu, 2012) by 0.57 points.

|                         |       | P     | R     | F            |
|-------------------------|-------|-------|-------|--------------|
| (Qian and Liu, 2012)    | Seg.  | 97.56 | 98.36 | 97.96        |
|                         | POS   | 93.43 | 94.2  | 93.81        |
|                         | Parse | 83.03 | 82.66 | 82.85        |
| Lattice-based Framework | Seg.  | 97.82 | 97.9  | 97.86        |
|                         | POS   | 94.36 | 94.44 | <b>94.40</b> |
|                         | Parse | 83.34 | 83.5  | <b>83.42</b> |

Table 4: Lattice-based framework evaluation.

## 5 Conclusion

In this paper, we present a novel lattice-based framework for the cascaded task of Chinese word segmentation, POS tagging and parsing. We first segment a Chinese sentence into a word lattice, then process the lattice using a lattice-based POS tagger and a lattice-based parser. We also design a strategy to exploit the complementary strengths of the tagger and the parser and encourage them to predict agreed structures. Experimental results show that the lattice-based framework significantly improves the accuracies of the three tasks. The parsing accuracy of the framework also outperforms the best joint parsing system reported in the literature.

## Acknowledgments

The research work has been funded by the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2011AA01A207, 2012AA011101, and 2012AA011102 and also supported by the Key Project of Knowledge Innovation Program of Chinese Academy of Sciences under Grant No.KGZD-EW-501. This work is also supported in part by the DAPRA via contract HR0011-11-C-0145 entitled "Linguistic Resources for Multilingual Processing".

## References

- S. Boyd, L. Xiao and A. Mutapcic. 2003. Subgradient methods. Lecture notes of EE392o, Stanford University.
- E. Charniak. 2000. A maximum-entropy-inspired parser. In NAACL '00, page 132-139.
- Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.
- Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proc. of EMNLP2002, pages 1-8.
- Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In Proc. of ACL2011.
- Wenbin Jiang, Haitao Mi and Qun Liu. 2008. Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In Proc. of Coling 2008, pages 385-392.
- Komodakis, N., Paragios, N., and Tziritas, G. 2007. MRF optimization via dual decomposition: Message-passing revisited. In ICCV 2007.
- C. Kruengkrai, K. Uchimoto, J. Kazama, Y. Wang, K. Torisawa and H. Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In Proc. of ACL2009, pages 513-521.
- Takuya Matsuzaki, Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In Proc. of ACL2005, pages 75-82.
- Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In Proc. of ACL2006, pages 433-440.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In Proc. of NAACL2007, pages 404-411.
- Xian Qian and Yang Liu. 2012. Joint Chinese Word segmentation, POS Tagging Parsing. In Proc. of EMNLP 2012, pages 501-511.
- Alexander M. Rush, David Sontag, Michael Collins and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In Proc. of EMNLP2010, pages 1-11.
- Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In Proc. of ACL2011, pages 1385-1394.
- Weiwei Sun and Hans Uszkoreit. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate Chinese part-of-speech tagging. In Proc. of ACL2012.
- Yiyou 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 Proc. of IJCNLP2011, pages 309-317.
- Nianwen Xue. 2003. Chinese word segmentation as character tagging. Computational Linguistics and Chinese Language Processing, 8 (1). pages 29-48.
- Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In Proc. of EMNLP2010, pages 843-852.

# Efficient Implementation of Beam-Search Incremental Parsers\*

Yoav Goldberg

Dept. of Computer Science

Bar-Ilan University

Ramat Gan, Tel Aviv, 5290002 Israel

yoav.goldberg@gmail.com

Kai Zhao

Liang Huang

Graduate Center and Queens College

City University of New York

{kzhao@gc, lhuang@cs.gc}.cuny.edu

{kzhao.hf, liang.huang.sh}.gmail.com

## Abstract

Beam search incremental parsers are accurate, but not as fast as they could be. We demonstrate that, contrary to popular belief, most current implementations of beam parsers in fact run in  $O(n^2)$ , rather than linear time, because each state-transition is actually implemented as an  $O(n)$  operation. We present an improved implementation, based on *Tree Structured Stack* (TSS), in which a transition is performed in  $O(1)$ , resulting in a real linear-time algorithm, which is verified empirically. We further improve parsing speed by sharing feature-extraction and dot-product across beam items. Practically, our methods combined offer a speedup of  $\sim 2x$  over strong baselines on Penn Treebank sentences, and are orders of magnitude faster on much longer sentences.

## 1 Introduction

Beam search incremental parsers (Roark, 2001; Collins and Roark, 2004; Zhang and Clark, 2008; Huang et al., 2009; Huang and Sagae, 2010; Zhang and Nivre, 2011; Zhang and Clark, 2011) provide very competitive parsing accuracies for various grammar formalisms (CFG, CCG, and dependency grammars). In terms of parsing strategies, they can be broadly divided into two categories: the first group (Roark, 2001; Collins and Roark, 2004) uses soft (aka probabilistic) beams borrowed from bottom-up parsers (Charniak, 2000; Collins, 1999) which has no control of complexity, while the second group (the rest and many more recent ones) employs hard beams borrowed from machine translation (Koehn, 2004) which guarantee (as they claim) a linear runtime  $O(kn)$  where  $k$  is the beam width. However, we will demonstrate below that, contrary to popular

belief, in most standard implementations their actual runtime is in fact  $O(kn^2)$  rather than linear. Although this argument in general also applies to dynamic programming (DP) parsers,<sup>1</sup> in this paper we only focus on the standard, non-dynamic programming approach since it is arguably still the dominant practice (e.g. it is easier with the popular arc-eager parser with a rich feature set (Kuhlmann et al., 2011; Zhang and Nivre, 2011)) and it benefits more from our improved algorithms.

The dependence on the beam-size  $k$  is because one needs to do  $k$ -times the number of basic operations (feature-extractions, dot-products, and state-transitions) relative to a greedy parser (Nivre and Scholz, 2004; Goldberg and Elhadad, 2010). Note that in a beam setting, the same state can expand to several new states in the next step, which is usually achieved by *copying* the state prior to making a transition, whereas greedy search only stores one state which is modified *in-place*.

Copying amounts to a large fraction of the slowdown of beam-based with respect to greedy parsers. Copying is expensive, because the state keeps track of (a) a stack and (b) the set of dependency-arcs added so far. Both the arc-set and the stack can grow to  $O(n)$  size in the worst-case, making the state-copy (and hence state-transition) an  $O(n)$  operation. Thus, beam search implementations that copy the entire state are in fact quadratic  $O(kn^2)$  and not linear, with a slowdown factor of  $O(kn)$  with respect to greedy parsers, which is confirmed empirically in Figure 4.

We present a way of decreasing the  $O(n)$  transition cost to  $O(1)$  achieving strictly linear-time parsing, using a data structure of **Tree-Structured Stack** (TSS) that is inspired by but simpler than the graph-structured stack (GSS) of Tomita (1985) used in dynamic programming (Huang and Sagae, 2010).<sup>2</sup> On average Treebank sentences, the TSS

<sup>1</sup>The Huang-Sagae DP parser (<http://acl.cs.gc.edu>) does run in  $O(kn)$ , which inspired this paper when we experimented with simulating non-DP beam search using GSS.

<sup>2</sup>Our notion of TSS is crucially different from the data

\*Supported in part by DARPA FA8750-13-2-0041 (DEFT).

|                       |   |
|-----------------------|---|
| input:                | $w_0 \dots w_{n-1}$   |
| axiom                 | $0 : \langle 0, \epsilon \rangle : \emptyset$   |
| SHIFT                 | $\frac{\ell : \langle j, S \rangle : A}{\ell + 1 : \langle j + 1, S   w_j \rangle : A} \quad j < n$                                     |
| REDUCE <sub>CEL</sub> | $\frac{\ell : \langle j, S   s_1   s_0 \rangle : A}{\ell + 1 : \langle j, S   s_0 \rangle : A \cup \{s_1 \hat{\curvearrowright} s_0\}}$ |
| REDUCER               | $\frac{\ell : \langle j, S   s_1   s_0 \rangle : A}{\ell + 1 : \langle j, S   s_1 \rangle : A \cup \{s_1 \hat{\curvearrowright} s_0\}}$ |
| goal                  | $2n - 1 : \langle n, s_0 \rangle : A$   |

Figure 1: An abstraction of the arc-standard deductive system Nivre (2008). The stack  $S$  is a list of heads,  $j$  is the index of the token at the front of the buffer, and  $\ell$  is the step number (beam index).  $A$  is the **arc-set** of dependency arcs accumulated so far, which we will get rid of in Section 4.1.

version, being linear time, leads to a speedup of  $2x \sim 2.7x$  over the naive implementation, and about  $1.3x \sim 1.7x$  over the optimized baseline presented in Section 5.

Having achieved efficient state-transitions, we turn to feature extraction and dot products (Section 6). We present a simple scheme of sharing repeated scoring operations across different beam items, resulting in an additional 7 to 25% speed increase. On Treebank sentences, the methods combined lead to a speedup of  $\sim 2x$  over strong baselines ( $\sim 10x$  over naive ones), and on longer sentences they are orders of magnitude faster.

## 2 Beam Search Incremental Parsing

We assume familiarity with transition-based dependency parsing. The unfamiliar reader is referred to Nivre (2008). We briefly describe a standard shift-reduce dependency parser (which is called “arc-standard” by Nivre) to establish notation. Parser *states* (sometimes called configurations) are composed of a stack, a buffer, and an arc-set. Parsing *transitions* are applied to states, and result in new states. The arc-standard system has three kinds of transitions: SHIFT, REDUCE<sub>CEL</sub>,

structure with the same name in an earlier work of Tomita (1985). In fact, Tomita’s TSS merges the top portion of the stacks (more like GSS) while ours merges the bottom portion. We thank Yue Zhang for informing us that TSS was already implemented for the CCG parser in *zpar* (<http://sourceforge.net/projects/zpar/>) though it was not mentioned in his paper (Zhang and Clark, 2011).

and REDUCER, which are summarized in the deductive system in Figure 1. The SHIFT transition removes the first word from the buffer and pushes it to the stack, and the REDUCE<sub>CEL</sub> and REDUCER actions each add a dependency relation between the two words on the top of the stack (which is achieved by adding the arc  $s_1 \hat{\curvearrowright} s_0$  or  $s_1 \hat{\curvearrowright} s_0$  to the arc-set  $A$ ), and pops the new dependent from the stack. When reaching the goal state the parser returns a tree composed of the arcs in the arc-set.

At parsing time, transitions are chosen based on a trained scoring model which looks at features of the state. In a *beam* parser,  $k$  items (hypotheses) are maintained. Items are composed of a state and a score. At step  $i$ , each of the  $k$  items is extended by applying all possible transitions to the given state, resulting in  $k \times a$  items,  $a$  being the number of possible transitions. Of these, the top scoring  $k$  items are kept and used in step  $i + 1$ . Finally, the tree associated with the highest-scoring item is returned.

## 3 The Common Implementation of State

The *stack* is usually represented as a list or an array of token indices, and the *arc-set* as an array *heads* of length  $n$  mapping the word at position  $m$  to the index of its parent. In order to allow for fast feature extraction, additional arrays are used to map each token to its left-most and right-most modifier, which are used in most incremental parsers, e.g. (Huang and Sagae, 2010; Zhang and Nivre, 2011). The buffer is usually implemented as a pointer to a *shared* sentence object, and an index  $j$  to the current front of the buffer. Finally, it is common to keep an additional array holding the transition sequence leading to the current state, which can be represented compactly as a pointer to the previous state and the current action. The state structure is summarized below:

```
class state
  stack[n] of token_ids
  array[n] heads
  array[n] leftmost_modifiers
  array[n] rightmost_modifiers
  int j
  int last_action
  state previous
```

In a greedy parser, state transition is performed in-place. However, in a beam parser the states cannot be modified in place, and a state transition operation needs to result in a new, independent state object. The common practice is to copy the current state, and then update the needed fields in the copy. Copying a stack and arrays of size  $n$  is an

$O(n)$  operation. In what follows, we present a way to perform transitions in  $O(1)$ .

## 4 Efficient State Transitions

### 4.1 Distributed Representation of Trees

The state needs to keep track of the set of arcs added to the tree so far for two reasons:

- (a) In order to return the complete tree at the end.
- (b) In order to compute features when parsing.

Observe that we do not in fact need to store any arc in order to achieve (a) – we could reconstruct the entire set by backtracking once we reach the final configuration. Hence, the arc-set in Figure 1 is only needed for computing features. Instead of storing the entire arc-set, we could keep only the information needed for feature computation. In the feature set we use (Huang and Sagae, 2010), we need access to (1) items on the buffer, (2) the 3 top-most elements of the stack, and (3) the current left-most and right-most modifiers of the two topmost stack elements. The left-most and right-most modifiers are already kept in the state representation, but store more information than needed: we only need to keep track of the modifiers of current stack items. Once a token is removed from the stack it will never return, and we will not need access to its modifiers again. We can therefore remove the left/rightmost modifier arrays, and instead have the stack store triplets (token, leftmost\_mod, rightmost\_mod). The heads array is no longer needed. Our new state representation becomes:

```
class state
  stack[n] of (tok, left, right)
  int j
  int last_action
  state previous
```

### 4.2 Tree Structured Stack: TSS

We now turn to handle the stack. Notice that the buffer, which is also of size  $O(n)$ , is represented as a pointer to an immutable shared object, and is therefore very efficient to copy. We would like to treat the stack in a similar fashion.

An *immutable stack* can be implemented functionally as a *cons list*, where the head is the top of the stack and the tail is the rest of the stack. Pushing an item to the stack amounts to adding a new head link to the list and returning it. Popping an item from the stack amounts to returning the tail of the list. Notice that, crucially, a pop operation does not change the underlying list at all, and

a push operation only adds to the front of a list. Thus, the stack operations are non-destructive, in the sense that once you hold a reference to a stack, the view of the stack through this reference does not change regardless of future operations that are applied to the stack. Moreover, push and pop operations are very efficient. This stack implementation is an example of a persistent data structure – a data structure inspired by functional programming which keeps the old versions of itself intact when modified (Okasaki, 1999).

While each client sees the stack as a list, the underlying representation is a tree, and clients hold pointers to nodes in the tree. A push operation adds a branch to the tree and returns the new pointer, while a pop operation returns the pointer of the parent, see Figure 3 for an example. We call this representation a tree-structured stack (TSS).

Using this stack representation, we can replace the  $O(n)$  stack by an integer holding the item at the top of the stack ( $s_0$ ), and a pointer to the tail of the stack ( $tail$ ). As discussed above, in addition to the top of the stack we also keep its leftmost and rightmost modifiers  $s_{0L}$  and  $s_{0R}$ . The simplified state representation becomes:

```
class state
  int s0, s0L, s0R
  state tail
  int j
  int last_action
  state previous
```

State is now reduced to seven integers, and the transitions can be implemented very efficiently as we show in Figure 2. The parser state is transformed into a compact object, and state transitions are  $O(1)$  operations involving only a few pointer lookups and integer assignments.

### 4.3 TSS vs. GSS; Space Complexity

TSS is inspired by the graph-structured stack (GSS) used in the dynamic-programming parser of Huang and Sagae (2010), but without reentrancy (see also Footnote 2). More importantly, the state signature in TSS is much slimmer than that in GSS. Using the notation of Huang and Sagae, instead of maintaining the full DP signature of

$$\tilde{\mathbf{f}}_{\text{DP}}(j, S) = (j, \mathbf{f}_d(s_d), \dots, \mathbf{f}_0(s_0))$$

where  $s_d$  denotes the  $d^{\text{th}}$  tree on stack, in non-DP TSS we only need to store the features  $\mathbf{f}_0(s_0)$  for the final tree on the stack,

$$\tilde{\mathbf{f}}_{\text{noDP}}(j, S) = (j, \mathbf{f}_0(s_0)),$$

```

def Shift(state)
    newstate.s0 = state.j
    newstate.s0L = None
    newstate.s0R = None
    newstate.tail = state
    newstate.j = state.j + 1
    return newstate

def ReduceL(state)
    newstate.s0 = state.s0
    newstate.s0L = state.tail.s0
    newstate.s0R = state.s0R
    newstate.tail = state.tail.tail
    newstate.j = j
    return newstate

def ReduceR(state)
    newstate.s0 = state.tail.s0
    newstate.s0L = state.tail.s0L
    newstate.s0R = state.s0
    newstate.tail = state.tail.tail
    newstate.j = j
    return newstate

```

Figure 2: State transitions implementation in the TSS representation (see Fig. 3 for the `tail` pointers). The two lines on `s0L` and `s0R` are specific to feature set design, and can be expanded for richer feature sets. To conserve space, we do not show the obvious assignments to `last_action` and `previous`.

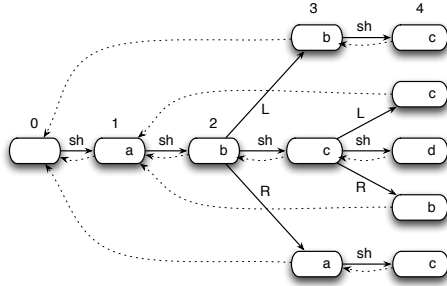


Figure 3: Example of tree-structured stack. The forward arrows denote state transitions, and the dotted backward arrows are the `tail` pointers to the stack tail. The boxes denote the top-of-stack at each state. Notice that for  $b = \text{shift}(a)$  we perform a single *push* operation getting  $b.\text{tail} = a$ , while for  $b = \text{reduce}(a)$  transition we perform two pops and a push, resulting in  $b.\text{tail} = a.\text{tail}.\text{tail}$ .

thanks to the uniqueness of `tail` pointers (“left-pointers” in Huang and Sagae).

In terms of space complexity, each state is reduced from  $O(n)$  in size to  $O(d)$  with GSS and to  $O(1)$  with TSS,<sup>3</sup> making it possible to store the entire beam in  $O(kn)$  space. Moreover, the constant state-size makes memory management easier and reduces fragmentation, by making it possible to pre-allocate the entire beam upfront. We did not explore its empirical implications in this work, as our implementation language, Python, does not support low-level memory management.

#### 4.4 Generality of the Approach

We presented a concrete implementation for the arc-standard system with a relatively simple (yet state-of-the-art) feature set. As in Kuhlmann et al. (2011), our approach is also applicable to other transitions systems and richer feature-sets with some additional book-keeping. A well-

<sup>3</sup>For example, a GSS state in Huang and Sagae’s experiments also stores `s1`, `s1L`, `s1R`, `s2` besides the  $\mathbf{f}_0(s_0)$  features (`s0`, `s0L`, `s0R`) needed by TSS.  $d$  is treated as a constant by Huang and Sagae but actually it could be a variable.

documented Python implementation for the *labeled arc-eager* system with the rich feature set of Zhang and Nivre (2011) is available on the first author’s homepage.

## 5 Fewer Transitions: Lazy Expansion

Another way of decreasing state-transition costs is making less transitions to begin with: instead of performing all possible transitions from each beam item and then keeping only  $k$  of the resulting states, we could perform only transitions that are sure to end up on the next step in the beam. This is done by first computing transition scores from each beam item, then keeping the top  $k$  highest scoring  $(\text{state}, \text{action})$  pairs, performing only those  $k$  transitions. This technique is especially important when the number of possible transitions is large, such as in labeled parsing. The technique, though never mentioned in the literature, was employed in some implementations (e.g., Yue Zhang’s `zpar`). We mention it here for completeness since it’s not well-known yet.

## 6 (Partial) Feature Sharing

After making the state-transition efficient, we turn to deal with the other major expensive operation: feature-extractions and dot-products. While we can’t speed up the process, we observe that some computations are repeated in different parts of the beam, and propose to share these computations. Notice that relatively few token indices from a state can determine the values of many features. For example, knowing the buffer index  $j$  determines the words and tags of items after location  $j$  on the buffer, as well as features composed of combinations of these values.

Based on this observation we propose the notion of a *state signature*, which is a set of token indices. An example of a state signature would be  $\text{sig}(\text{state}) = (\text{s0}, \text{s0L}, \text{s1}, \text{s1L})$ , indicating the indices of the two tokens at the top of the stack together with their leftmost modifiers. Given a sig-

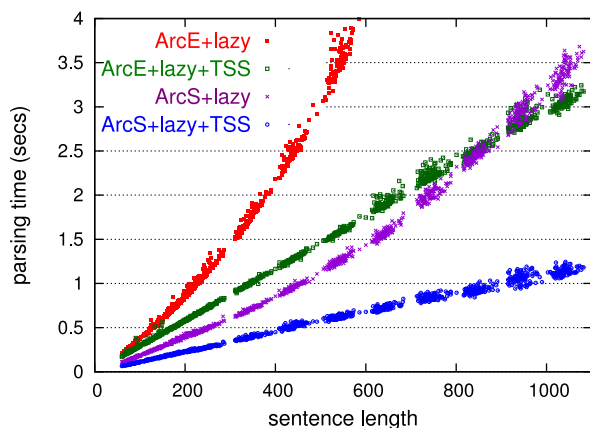


Figure 4: Non-linearity of the standard beam search compared to the linearity of our TSS beam search for labeled arc-eager and unlabeled arc-standard parsers on long sentences (running times vs. sentence length). All parsers use beam size 8.

nature, we decompose the feature function  $\phi(x)$  into two parts  $\phi(x) = \phi_s(\text{sig}(x)) + \phi_o(x)$ , where  $\phi_s(\text{sig}(x))$  extracts all features that depend exclusively on signature items, and  $\phi_o(x)$  extracts all other features.<sup>4</sup> The scoring function  $\mathbf{w} \cdot \phi(x)$  decomposes into  $\mathbf{w} \cdot \phi_s(\text{sig}(x)) + \mathbf{w} \cdot \phi_o(x)$ . During beam decoding, we maintain a cache mapping seen signatures  $\text{sig}(\text{state})$  to (partial) transition scores  $\mathbf{w} \cdot \phi_s(\text{sig}(\text{state}))$ . We now need to calculate  $\mathbf{w} \cdot \phi_o(x)$  for each beam item, but  $\mathbf{w} \cdot \phi_s(\text{sig}(x))$  only for one of the items sharing the signature. Defining the signature involves a natural balance between signatures that repeat often and signatures that cover many features. In the experiments in this paper, we chose the signature function for the arc-standard parser to contain all core elements participating in feature extraction<sup>5</sup>, and for the arc-eager parser a signature containing only a partial subset.<sup>6</sup>

## 7 Experiments

We implemented beam-based parsers using the traditional approach as well as with our proposed extension and compared their runtime.

The first experiment highlights the non-linear behavior of the standard implementation, compared to the linear behavior of the TSS method.

<sup>4</sup>One could extend the approach further to use several signatures and further decompose the feature function. We did not pursue this idea in this work.

<sup>5</sup> $s_0, s_0L, s_0R, s_1, s_1L, s_1R, s_2, j$ .

<sup>6</sup> $s_0, s_0L, s_0R, s_0h, b_0L, j$ , where  $s_0h$  is the parent of  $s_0$ , and  $b_0L$  is the leftmost modifier of  $j$ .

| system | plain<br>(sec 3) | plain<br>+TSS<br>(sec 4) | plain<br>+lazy<br>(sec 5) | plain<br>+TSS<br>+lazy | +TSS+lazy<br>+feat-share<br>(sec 6) |
|--------|------------------|--------------------------|---------------------------|------------------------|-------------------------------------|
| ArcS-U | 20.8             | 38.6                     | 24.3                      | 41.1                   | 47.4                                |
| ArcE-U | 25.4             | 48.3                     | 38.2                      | 58.2                   | 72.3                                |
| ArcE-L | 1.8              | 4.9                      | 11.1                      | 14.5                   | 17.3                                |

Table 1: Parsing speeds for the different techniques measured in sentences/sec (beam size 8). All parsers are implemented in Python, with dot-products in C. ArcS/ArcE denotes arc-standard vs. arc-eager, L/U labeled (stanford deps, 49 labels) vs. unlabeled parsing. ArcS use feature set of Huang and Sagae (2010) (50 templates), and ArcE that of Zhang and Nivre (2011) (72 templates).

As parsing time is dominated by score computation, the effect is too small to be measured on natural language sentences, but it is noticeable for longer sentences. Figure 4 plots the runtime for synthetic examples with lengths ranging from 50 to 1000 tokens, which are generated by concatenating sentences from Sections 22–24 of Penn Treebank (PTB), and demonstrates the non-linear behavior (dataset included). We argue parsing longer sentences is by itself an interesting and potentially important problem (e.g. for other languages such as Arabic and Chinese where word or sentence boundaries are vague, and for parsing beyond sentence-level, e.g. discourse parsing or parsing with inter-sentence dependencies).

Our next set of experiments compares the actual speedup observed on English sentences. Table 1 shows the speed of the parsers (sentences/second) with the various proposed optimization techniques. We first train our parsers on Sections 02–21 of PTB, using Section 22 as the test set. The accuracies of all our parsers are at the state-of-the-art level. The final speedups are up to 10x against naive baselines and  $\sim 2x$  against the lazy-transitions baselines.

## 8 Conclusions

We demonstrated in both theory and experiments that the standard implementation of beam search parsers run in  $O(n^2)$  time, and have presented improved algorithms which run in  $O(n)$  time. Combined with other techniques, our method offers significant speedups ( $\sim 2x$ ) over strong baselines, or 10x over naive ones, and is orders of magnitude faster on much longer sentences. We have demonstrated that our approach is general and we believe it will benefit many other incremental parsers.



## References

- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*.
- Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Yoav Goldberg and Michael Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of HLT-NAACL*, pages 742–750.
- Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL 2010*.
- Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of EMNLP*.
- Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA*, pages 115–124.
- Marco Kuhlmann, Carlos Gmez-Rodrguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of ACL*.
- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of english text. In *Proceedings of COLING*, Geneva.
- Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.
- Chris Okasaki. 1999. *Purely functional data structures*. Cambridge University Press.
- Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.
- Masaru Tomita. 1985. An efficient context-free parsing algorithm for natural languages. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 2*, pages 756–764.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*.
- Yue Zhang and Stephen Clark. 2011. Shift-reduce ccg parsing. In *Proceedings of ACL*.
- Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL*, pages 188–193.

# Simpler unsupervised POS tagging with bilingual projections

Long Duong,<sup>1,2</sup> Paul Cook,<sup>1</sup> Steven Bird,<sup>1</sup> and Pavel Pecina<sup>2</sup>

1 Department of Computing and Information Systems, The University of Melbourne

2 Charles University in Prague, Czech Republic

lduong@student.unimelb.edu.au, paulcook@unimelb.edu.au,

sbird@unimelb.edu.au, pecina@ufal.mff.cuni.cz

## Abstract

We present an unsupervised approach to part-of-speech tagging based on projections of tags in a word-aligned bilingual parallel corpus. In contrast to the existing state-of-the-art approach of Das and Petrov, we have developed a substantially simpler method by automatically identifying “good” training sentences from the parallel corpus and applying self-training. In experimental results on eight languages, our method achieves state-of-the-art results.

## 1 Unsupervised part-of-speech tagging

Currently, part-of-speech (POS) taggers are available for many highly spoken and well-resourced languages such as English, French, German, Italian, and Arabic. For example, Petrov et al. (2012) build supervised POS taggers for 22 languages using the TNT tagger (Brants, 2000), with an average accuracy of 95.2%. However, many widely-spoken languages — including Bengali, Javanese, and Lahnda — have little data manually labelled for POS, limiting supervised approaches to POS tagging for these languages.

However, with the growing quantity of text available online, and in particular, multilingual parallel texts from sources such as multilingual websites, government documents and large archives of human translations of books, news, and so forth, *unannotated parallel data* is becoming more widely available. This parallel data can be exploited to bridge languages, and in particular, transfer information from a highly-resourced language to a lesser-resourced language, to build unsupervised POS taggers.

In this paper, we propose an unsupervised approach to POS tagging in a similar vein to the work of Das and Petrov (2011). In this approach,

a parallel corpus for a more-resourced language having a POS tagger, and a lesser-resourced language, is word-aligned. These alignments are exploited to infer an unsupervised tagger for the target language (i.e., a tagger not requiring manually-labelled data in the target language). Our approach is substantially simpler than that of Das and Petrov, the current state-of-the-art, yet performs comparably well.

## 2 Related work

There is a wealth of prior research on building unsupervised POS taggers. Some approaches have exploited similarities between typologically similar languages (e.g., Czech and Russian, or Telugu and Kannada) to estimate the transition probabilities for an HMM tagger for one language based on a corpus for another language (e.g., Hana et al., 2004; Feldman et al., 2006; Reddy and Sharoff, 2011). Other approaches have simultaneously tagged two languages based on alignments in a parallel corpus (e.g., Snyder et al., 2008).

A number of studies have used *tag projection* to copy tag information from a resource-rich to a resource-poor language, based on word alignments in a parallel corpus. After alignment, the resource-rich language is tagged, and tags are projected from the source language to the target language based on the alignment (e.g., Yarowsky and Ngai, 2001; Das and Petrov, 2011). Das and Petrov (2011) achieved the current state-of-the-art for unsupervised tagging by exploiting high confidence alignments to copy tags from the source language to the target language. Graph-based label propagation was used to automatically produce more labelled training data. First, a graph was constructed in which each vertex corresponds to a unique trigram, and edge weights represent the syntactic similarity between vertices. Labels were then propagated by optimizing a convex function to favor the same tags for closely related nodes

| Model                            | Coverage | Accuracy |
|----------------------------------|----------|----------|
| Many-to-1 alignments             | 88%      | 68%      |
| 1-to-1 alignments                | 68%      | 78%      |
| 1-to-1 alignments: Top 60k sents | 91%      | 80%      |

Table 1: Token coverage and accuracy of many-to-one and 1-to-1 alignments, as well as the top 60k sentences based on alignment score for 1-to-1 alignments, using directly-projected labels only.

while keeping a uniform tag distribution for unrelated nodes. A tag dictionary was then extracted from the automatically labelled data, and this was used to constrain a feature-based HMM tagger.

The method we propose here is simpler to that of Das and Petrov in that it does not require convex optimization for label propagation or a feature based HMM, yet it achieves comparable results.

### 3 Tagset

Our tagger exploits the idea of projecting tag information from a resource-rich to resource-poor language. To facilitate this mapping, we adopt Petrov et al.’s (2012) twelve universal tags: 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). These twelve basic tags are common across taggers for most languages.

Adopting a universal tagset avoids the need to map between a variety of different, language-specific tagsets. Furthermore, it makes it possible to apply unsupervised tagging methods to languages for which no tagset is available, such as Telugu and Vietnamese.

## 4 A Simpler Unsupervised POS Tagger

Here we describe our proposed tagger. The key idea is to maximize the amount of information gleaned from the source language, while limiting the amount of noise. We describe the seed model and then explain how it is successively refined through self-training and revision.

### 4.1 Seed Model

The first step is to construct a seed tagger from directly-projected labels. Given a parallel corpus for a source and target language, Algorithm 1 provides a method for building an unsupervised tagger for the target language. In typical applications,

the source language would be a better-resourced language having a tagger, while the target language would be lesser-resourced, lacking a tagger and large amounts of manually POS-labelled data.

---

#### Algorithm 1 Build seed model

---

- 1: Tag source side.
  - 2: Word align the corpus with Giza++ and remove the many-to-one mappings.
  - 3: Project tags from source to target using the remaining 1-to-1 alignments.
  - 4: Select the top  $n$  sentences based on sentence alignment score.
  - 5: Estimate emission and transition probabilities.
  - 6: Build seed tagger T.
- 

We eliminate many-to-one alignments (Step 2). Keeping these would give more POS-tagged tokens for the target side, but also introduce noise. For example, suppose English and French were the source and target language, respectively. In this case alignments such as English *laws* (NNS) to French *les* (DT) *lois* (NNS) would be expected (Yarowsky and Ngai, 2001). However, in Step 3, where tags are projected from the source to target language, this would incorrectly tag French *les* as NN. We build a French tagger based on English–French data from the Europarl Corpus (Koehn, 2005). We also compare the accuracy and coverage of the tags obtained through direct projection using the French Melt POS tagger (Denis and Sagot, 2009). Table 1 confirms that the one-to-one alignments indeed give higher accuracy but lower coverage than the many-to-one alignments. At this stage of the model we hypothesize that high-confidence tags are important, and hence eliminate the many-to-one alignments.

In Step 4, in an effort to again obtain higher quality target language tags from direct projection, we eliminate all but the top  $n$  sentences based on their alignment scores, as provided by the aligner via IBM model 3. We heuristically set this cutoff to 60k to balance the accuracy and size of the seed model.<sup>1</sup> Returning to our preliminary English–French experiments in Table 1, this process gives improvements in both accuracy and coverage.<sup>2</sup>

<sup>1</sup>We considered values in the range 60–90k, but this choice had little impact on the accuracy of the model.

<sup>2</sup>We also considered using all projected labels for the top 60k sentences, not just 1-to-1 alignments, but in preliminary experiments this did not perform as well, possibly due to the previously-observed problems with many-to-one alignments.

The number of parameters for the emission probability is  $|V| \times |T|$  where  $V$  is the vocabulary and  $T$  is the tag set. The transition probability, on the other hand, has only  $|T|^3$  parameters for the trigram model we use. Because of this difference in number of parameters, in step 5, we use different strategies to estimate the emission and transition probabilities. The emission probability is estimated from all 60k selected sentences. However, for the transition probability, which has less parameters, we again focus on “better” sentences, by estimating this probability from only those sentences that have (1) token coverage  $> 90\%$  (based on direct projection of tags from the source language), and (2) length  $> 4$  tokens. These criteria aim to identify longer, mostly-tagged sentences, which we hypothesize are particularly useful as training data. In the case of our preliminary English–French experiments, roughly 62% of the 60k selected sentences meet these criteria and are used to estimate the transition probability. For unaligned words, we simply assign a random POS and very low probability, which does not substantially affect transition probability estimates.

In Step 6 we build a tagger by feeding the estimated emission and transition probabilities into the TNT tagger (Brants, 2000), an implementation of a trigram HMM tagger.

## 4.2 Self training and revision

For self training and revision, we use the seed model, along with the large number of target language sentences available that have been partially tagged through direct projection, in order to build a more accurate tagger. Algorithm 2 describes this process of self training and revision, and assumes that the parallel source–target corpus has been word aligned, with many-to-one alignments removed, and that the sentences are sorted by alignment score. In contrast to Algorithm 1, all sentences are used, not just the 60k sentences with the highest alignment scores.

We believe that sentence alignment score might correspond to difficulty to tag. By sorting the sentences by alignment score, sentences which are more difficult to tag are tagged using a more mature model. Following Algorithm 1, we divide sentences into blocks of 60k.

In step 3 the tagged block is revised by comparing the tags from the tagger with those obtained through direct projection. Suppose source

---

### Algorithm 2 Self training and revision

---

- 1: Divide target language sentences into blocks of  $n$  sentences.
  - 2: Tag the first block with the seed tagger.
  - 3: Revise the tagged block.
  - 4: Train a new tagger on the tagged block.
  - 5: Add the previous tagger’s lexicon to the new tagger.
  - 6: Use the new tagger to tag the next block.
  - 7: Goto 3 and repeat until all blocks are tagged.
- 

language word  $w_i^s$  is aligned with target language word  $w_j^t$  with probability  $p(w_j^t|w_i^s)$ ,  $T_i^s$  is the tag for  $w_i^s$  using the tagger available for the source language, and  $T_j^t$  is the tag for  $w_j^t$  using the tagger learned for the target language. If  $p(w_j^t|w_i^s) > S$ , where  $S$  is a threshold which we heuristically set to 0.7, we replace  $T_j^t$  by  $T_i^s$ .

Self-training can suffer from over-fitting, in which errors in the original model are repeated and amplified in the new model (McClosky et al., 2006). To avoid this, we remove the tag of any token that the model is uncertain of, i.e., if  $p(w_j^t|w_i^s) < S$  and  $T_j^t \neq T_i^s$  then  $T_j^t = \text{Null}$ . So, on the target side, aligned words have a tag from direct projection or no tag, and unaligned words have a tag assigned by our model.

Step 4 estimates the emission and transition probabilities as in Algorithm 1. In Step 5, emission probabilities for lexical items in the previous model, but missing from the current model, are added to the current model. Later models therefore take advantage of information from earlier models, and have wider coverage.

## 5 Experimental Results

Using parallel data from Europarl (Koehn, 2005) we apply our method to build taggers for the same eight target languages as Das and Petrov (2011) — Danish, Dutch, German, Greek, Italian, Portuguese, Spanish and Swedish — with English as the source language. Our training data (Europarl) is a subset of the training data of Das and Petrov (who also used the ODS United Nations dataset which we were unable to obtain). The evaluation metric and test data are the same as that used by Das and Petrov. Our results are comparable to theirs, although our system is penalized by having less training data. We tag the source language with the Stanford POS tagger (Toutanova et al., 2003).

|                          | Danish      | Dutch       | German      | Greek       | Italian     | Portuguese  | Spanish     | Swedish     | Average     |
|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Seed model               | 83.7        | 81.1        | 83.6        | 77.8        | 78.6        | 84.9        | 81.4        | 78.9        | 81.3        |
| Self training + revision | <b>85.6</b> | <b>84.0</b> | <b>85.4</b> | 80.4        | 81.4        | 86.3        | 83.3        | <b>81.0</b> | 83.4        |
| Das and Petrov (2011)    | 83.2        | 79.5        | 82.8        | <b>82.5</b> | <b>86.8</b> | <b>87.9</b> | <b>84.2</b> | 80.5        | <b>83.4</b> |

Table 2: Token-level POS tagging accuracy for our seed model, self training and revision, and the method of Das and Petrov (2011). The best results on each language, and on average, are shown in bold.

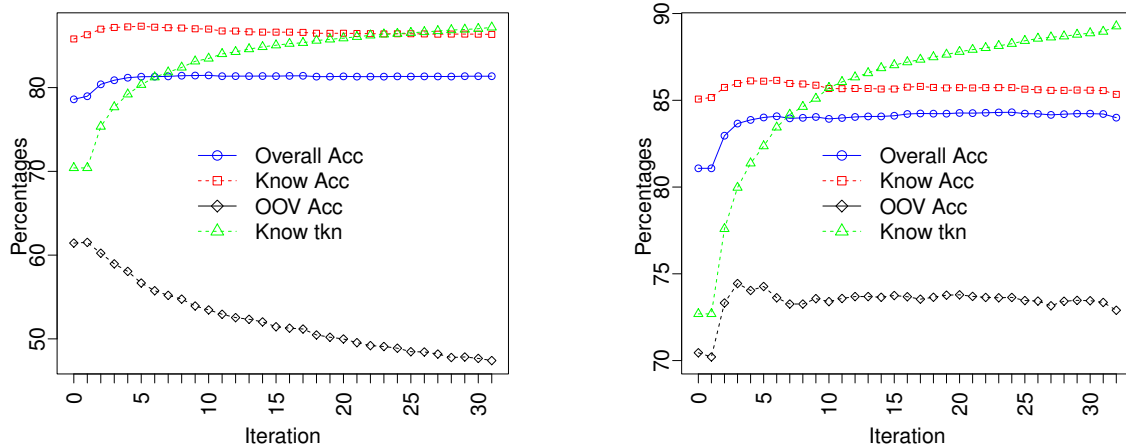


Figure 1: Overall accuracy, accuracy on known tokens, accuracy on unknown tokens, and proportion of known tokens for Italian (left) and Dutch (right).

Table 2 shows results for our seed model, self training and revision, and the results reported by Das and Petrov. Self training and revision improve the accuracy for every language over the seed model, and gives an average improvement of roughly two percentage points. The average accuracy of self training and revision is on par with that reported by Das and Petrov. On individual languages, self training and revision and the method of Das and Petrov are split — each performs better on half of the cases. Interestingly, our method achieves higher accuracies on Germanic languages — the family of our source language, English — while Das and Petrov perform better on Romance languages. This might be because our model relies on alignments, which might be more accurate for more-related languages, whereas Das and Petrov additionally rely on label propagation.

Compared to Das and Petrov, our model performs poorest on Italian, in terms of percentage point difference in accuracy. Figure 1 (left panel) shows accuracy, accuracy on known words, accuracy on unknown words, and proportion of known tokens for each iteration of our model for Italian; iteration 0 is the seed model, and iteration 31 is the final model. Our model performs poorly on unknown words as indicated by the low accuracy on unknown words, and high accuracy on known

words compared to the overall accuracy. The poor performance on unknown words is expected because we do not use any language-specific rules to handle this case. Moreover, on average for the final model, approximately 10% of the test data tokens are unknown. One way to improve the performance of our tagger might be to reduce the proportion of unknown words by using a larger training corpus, as Das and Petrov did.

We examine the impact of self-training and revision over training iterations. We find that for all languages, accuracy rises quickly in the first 5–6 iterations, and then subsequently improves only slightly. We exemplify this in Figure 1 (right panel) for Dutch. (Findings are similar for other languages.) Although accuracy does not increase much in later iterations, they may still have some benefit as the vocabulary size continues to grow.

## 6 Conclusion

We have proposed a method for unsupervised POS tagging that performs on par with the current state-of-the-art (Das and Petrov, 2011), but is substantially less-sophisticated (specifically not requiring convex optimization or a feature-based HMM). The complexity of our algorithm is  $O(n \log n)$  compared to  $O(n^2)$  for that of Das and Petrov

(2011) where  $n$  is the size of training data.<sup>3</sup> We made our code available for download.<sup>4</sup>

In future work we intend to consider using a larger training corpus to reduce the proportion of unknown tokens and improve accuracy. Given the improvements of our model over that of Das and Petrov on languages from the same family as our source language, and the observation of Snyder et al. (2008) that a better tagger can be learned from a more-closely related language, we also plan to consider strategies for selecting an appropriate source language for a given target language. Using our final model with unsupervised HMM methods might improve the final performance too, i.e. use our final model as the initial state for HMM, then experiment with different inference algorithms such as Expectation Maximization (EM), Variational Bayes (VB) or Gibbs sampling (GS).<sup>5</sup> Gao and Johnson (2008) compare EM, VB and GS for unsupervised English POS tagging. In many cases, GS outperformed other methods, thus we would like to try GS first for our model.

## 7 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). We would like to thank Prokopis Prokopidis for providing us the Greek Treebank and Antonia Marti for the Spanish CoNLL 06 dataset. Finally, we thank Siva Reddy and Spandana Gella for many discussions and suggestions.

## 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*

<sup>3</sup>We re-implemented label propagation from Das and Petrov (2011). It took over a day to complete this step on an eight core Intel Xeon 3.16GHz CPU with 32 Gb Ram, but only 15 minutes for our model.

<sup>4</sup><https://code.google.com/p/universal-tagger/>

<sup>5</sup>We in fact have tried EM, but it did not help. The overall performance dropped slightly. This might be because self-training with revision already found the local maximal point.

*the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (ACL 2011)*, pages 600–609. Portland, Oregon, USA.

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*, pages 721–736. Hong Kong, China.

Anna Feldman, Jirka Hana, and Chris Brew. 2006. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'06)*, pages 549–554. Genoa, Italy.

Jianfeng Gao and Mark Johnson. 2008. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 344–352. Association for Computational Linguistics, Stroudsburg, PA, USA.

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. AAMT, Phuket, Thailand.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. 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 152–159. New York, USA.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096. Istanbul, Turkey.

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 IJCNLP 2011 workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies (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 Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 1041–1050. Honolulu, Hawaii.

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 (NAACL '03)*, pages 173–180. Edmonton, Canada.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. 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. Pittsburgh, Pennsylvania, USA.

# Part-of-speech tagging with antagonistic adversaries

Anders Søgaard

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

## Abstract

Supervised NLP tools and on-line services are often used on data that is very different from the manually annotated data used during development. The performance loss observed in such cross-domain applications is often attributed to covariate shifts, with out-of-vocabulary effects as an important subclass. Many discriminative learning algorithms are sensitive to such shifts because highly indicative features may swamp other indicative features. Regularized and adversarial learning algorithms have been proposed to be more robust against covariate shifts. We present a new perceptron learning algorithm using antagonistic adversaries and compare it to previous proposals on 12 multilingual cross-domain part-of-speech tagging datasets. While previous approaches do not improve on our supervised baseline, our approach is better across the board with an average 4% error reduction.

## 1 Introduction

Most learning algorithms assume that training and test data are governed by identical distributions; and more specifically, in the case of part-of-speech (POS) tagging, that training and test sentences were sampled at random and that they are identically and independently distributed. Significance is usually tested across data points in standard NLP test sets. Such datasets typically contain running text rather than independently sampled sentences, thereby violating the assumption that data points are independently distributed and sampled at random. More importantly, significance across

data points only says something about the likelihood of observing the same effect on *more data sampled the same way*, but says nothing about likely performance on sentences sampled from different sources or different domains.

This paper considers the POS tagging problem, i.e. where we have training and test data consisting of sentences in which all words are assigned a label  $y$  chosen from a finite set of class labels {NOUN, VERB, DET, ...}. We assume that we are interested in *performance across data sets or domains* rather than just performance across data points, but that we do not know the target domain in advance. This is often the case when we develop NLP tools and on-line services. We will do cross-domain experiments using several target domains in order to compute significance across domains, enabling us to say something about likely performance on new domains.

Several authors have noted how POS tagging performance is sensitive to cross-domain shifts (Blitzer et al., 2006; Daume III, 2007; Jiang and Zhai, 2007), and while most authors have assumed known target distributions and pool unlabeled target data in order to automatically correct cross-domain bias (Jiang and Zhai, 2007; Foster et al., 2010), methods such as feature bagging (Sutton et al., 2006), learning with random adversaries (Globerson and Roweis, 2006) and  $L_\infty$ -regularization (Dekel and Shamir, 2008) have been proposed to improve performance on unknown target distributions. These methods explicitly or implicitly try to minimize average or worst-case expected error across a set of possible test distributions in various ways. These algorithms are related because of the intimate relationship between adversarial corruption and regularization (Ghaoui and Le Bret, 1997; Xu et al.,



2009; Hinton et al., 2012). This paper presents a new method based on learning with antagonistic adversaries.

**Outline.** Section 2 introduces previous work on robust perceptron learning, as well as the methods discussed in the paper. Section 3 motivates and introduces learning with antagonistic adversaries. Section 4 presents experiments on POS tagging and discusses how to evaluate cross-domain performance. Learning with antagonistic adversaries is superior to the other approaches across 10/12 datasets with an average error reduction of 4% over a supervised baseline.

**Motivating example.** The problem with out-of-vocabulary effects can be illustrated using a small labeled data set:  $\{\mathbf{x}_1 = \langle 1, \langle 0, 1, 0 \rangle \rangle, \mathbf{x}_2 = \langle 1, \langle 0, 1, 1 \rangle \rangle, \mathbf{x}_3 = \langle 0, \langle 0, 0, 0 \rangle \rangle, \mathbf{x}_4 = \langle 1, \langle 0, 0, 1 \rangle \rangle\}$ . Say we train our model on  $\mathbf{x}_{1-3}$  and evaluate it on the fourth data point. Most discriminate learning algorithms only update parameters when training examples are misclassified. In this example, a model initialized by zero weights would misclassify  $\mathbf{x}_1$ , update the parameter associated with feature  $x_2$  at a fixed rate  $\alpha$ , and the returned model would then classify all data points correctly. Hence the parameter associated with feature  $x_3$  would never be updated, although this feature is also correlated with class. If  $x_2$  is missing in our test data (out-of-vocabulary), we end up classifying all data points as negative. In this case, we would wrongly predict that  $\mathbf{x}_4$  is negative.

## 2 Robust perceptron learning

Our framework will be averaged perceptron learning (Freund and Schapire, 1999; Collins, 2002). We use an additive update algorithm and average parameters to prevent over-fitting. In adversarial learning, adversaries corrupt the data point by applying transformations to data points. Antagonistic adversaries choose transformations informed by the current model parameters  $\mathbf{w}$ , but random adversaries randomly select transformations from a predefined set of possible transformations, e.g. deletions of at most  $k$  features (Globerson and Roweis, 2006).

**Feature bagging.** In feature bagging (Sutton et al., 2006), the data is represented by different bags of features or different views, and the models learned using different feature bags are combined by averaging. We can reformulate feature bagging as an

adversarial learning problem. For each pass, the adversary chooses a deleting transformation corresponding to one of the feature bags. In Sutton et al. (2006), the feature bags simply divide the features into two or more representations. In an online setting feature bagging can be modelled as a game between a learner and an adversary, in which (a) the adversary can only choose between deleting transformations, (b) the adversary cannot see model parameters when choosing a transformation, and in which (c) the adversary only moves in between passes over the data.<sup>1</sup>

**Learning with random adversaries (LRA).** Globerson and Roweis (2006) let an adversary corrupt labeled data during training to learn better models of test data with missing features. They assume that missing features are randomly distributed and show that the optimization problem is a second-order cone program. LRA is an adversarial game in which the two players are unaware of the other player’s current move, and in particular, where the adversary does not see model parameters and only randomly corrupts the data points. Globerson and Roweis (2006) formulate LRA as a batch learning problem of minimizing worst case loss under deleting transformations deleting at most  $k$  features. This is related to regularization in the following way: If model parameters are chosen to minimize expected error in the absence of any  $k$  features, we explicitly prevent under-weighting more than  $n - k$  features, i.e. the model must be able to classify data well in the absence of any  $k$  features. The sparsest possible model would thus assign weights to  $k + 1$  parameters.

**$L_\infty$ -regularization** hedges its bets even more than adversarial learning by minimizing expected error with  $\max \|\mathbf{w}\| < C$ . In the online setting, this corresponds to playing against an adversary that clips any weight above a certain threshold  $C$ , whether positive or negative (Dekel and Shamir, 2008). In geometric terms the weights are projected back onto the hyper-cube  $C$ . A related approach, which is not explored in the experiments below, is to regularize linear models toward weights with low variance (Bergsma et al., 2010).

<sup>1</sup>Note that the batch version of feature bagging is an instance of group  $L_1$  regularization (Jacob et al., 2009; Schmidt and Murphy, 2010; Martins et al., 2011). Often group regularization is about finding sparser models rather than robust models. Sparse models can be obtained by grouping correlated features; non-sparse models can be obtained by using independent, exhaustive views.

```

1:  $X = \{(y_i, \mathbf{x}_i)\}_{i=1}^N, \delta$  deletion rate
2:  $\mathbf{w}^0 = 0, \mathbf{v} = 0, i = 0$ 
3: for  $k \in K$  do
4:   for  $n \in N$  do
5:      $\xi^1 \leftarrow \text{random.sample}(P(1) = 1 - \delta)$ 
6:      $\xi^2 \leftarrow \|\mathbf{w}\| < \mu_{\|\mathbf{w}\|} + \sigma_{\|\mathbf{w}\|}$ 
7:      $\xi \leftarrow (\xi^1 + \xi^2)_{(0,1)}$ 
8:     if  $\text{sign}(\mathbf{w} \cdot \mathbf{x}_n \circ \xi) \neq y_n$  then
9:        $\mathbf{w}^{i+1} \leftarrow \text{update}(\mathbf{w}^i)$ 
10:       $i \leftarrow i + 1$ 
11:     end if
12:    $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{w}^i$ 
13:   end for
14: end for
15: return  $\mathbf{w} = \mathbf{v} / (N \times K)$ 

```

Figure 1: Learning with antagonistic adversaries

### 3 Learning with antagonistic adversaries

The intuition behind learning with antagonistic adversaries is that the adversary should focus on the most predictive features. In the prediction game, this would allow the adversary to inflict more damage, corrupting data points by removing good features (rather than random ones). If the adversary focuses on the most predictive features, she is implicitly regularizing the model to obtain a more equal distribution of weights.

We draw random binary vectors with  $P(1) = 1 - \delta$  as in adversarial learning, but deletions are only effective if  $\xi_j = 0$  and the weight  $w_j$  is more than a standard deviation ( $\sigma_{\|\mathbf{w}\|}$ ) from the mean of the current absolute weight distribution ( $\mu_{\|\mathbf{w}\|}$ ). In other words, we only delete the predictive features, with predictivity being relative to the current mean weight.

The algorithm is presented in Figure 1. For each data point, we draw a random binary vector  $\xi_1$  with  $\delta$  chance of zeros.  $\xi_2$  is a vector with the  $i$ th scalar zero if and only if the absolute value of the weight  $w_i$  in  $\mathbf{w}$  is more than a standard deviation higher than the current mean. The  $i$ th scalar in  $\xi$  is only zero if the  $i$ th scalars in both  $\xi_1$  and  $\xi_2$  are zero. The corresponding features are a random subset of the predictive features.<sup>2</sup>

<sup>2</sup>The approach taken is similar in spirit to confidence-weighted learning (Dredze et al., 2008). The intuition behind confidence-weighted learning is to more aggressively update rare features or features that we are less confident about. In learning with antagonistic adversaries the adversaries delete predictive features; that is, features that we are confident about. When these features are deleted, we do not update the corresponding weights. In relative terms, we therefore update rare features more aggressively than common ones. Note also that by doing so we regularize toward weights with low variance (Bergsma et al., 2010).

## 4 Experiments

We consider part-of-speech (POS) tagging, i.e. the problem of assigning syntactic categories to word tokens in running text. POS tagging accuracy is known to be very sensitive to domain shifts. Foster et al. (2011) report a POS tagging accuracy on social media data of 84% using a tagger that achieves an accuracy of about 97% on newspaper data. In the case of social media data, many errors occur due to different spelling and capitalization conventions. The main source of error, though, is the increased out-of-vocabulary rate, i.e. the many unknown words. While POS taggers can often recover the part of speech of a previously unseen word from the context it occurs in, this is harder than for previously seen words.

We use the LXMLS toolkit<sup>3</sup> as our baseline with the default feature model, but use the PTB tagset rather than the Google tagset (Petrov et al., 2011) used by default in the LXMLS toolkit. We use four groups of datasets. The first group comes from the English Web Treebank (EWT),<sup>4</sup> also used in the Parsing the Web shared task (Petrov and McDonald, 2012). We train our tagger on Sections 2–21 of the WSJ data in the Penn-III Treebank (PTB), Ontonotes 4.0 release. The EWT contains development and test data for five domains: answers (from Yahoo!), emails (from the Enron corpus), BBC newsgroups, Amazon reviews, and weblogs. We use the emails development section for development and test on the remaining four test sets. We also do experiments with additional data from PTB. For these experiments we use the 0th even split of the biomedical section (PTB-biomedical) as development data, the 9th split and the chemistry section (PTB-chemistry) as test data, and the remaining biomedical data (splits 1–8) as training data. This data was also used for developing and testing in the CoNLL 2007 Shared Task (Nivre et al., 2007).

Our third group of datasets also comes from Ontonotes 4.0.<sup>5</sup> We use the Chinese Ontonotes (CHO) data, covering five different domains. We use newswire for training data and randomly sampled broadcasted news for development. Finally we do experiments with the Danish section of the Copenhagen Dependency Treebank (CDT). For CDT we rely on the treebank meta-data and sin-

<sup>3</sup><https://github.com/gracaninja/lxmls-toolkit>

<sup>4</sup>LDC Catalog No.: LDC2012T13.

<sup>5</sup>LDC Catalog No.: LDC2011T03.

|                       | SP    | Our          | $L_\infty$   | LRA          |
|-----------------------|-------|--------------|--------------|--------------|
| EWT-answers           | 86.04 | <b>86.06</b> | 85.90        | <b>86.06</b> |
| EWT-newsgroups        | 87.70 | <b>87.92</b> | 87.78        | 87.66        |
| EWT-reviews           | 85.96 | <b>86.10</b> | 85.80        | 86.00        |
| EWT-weblogs           | 87.59 | <b>87.89</b> | 87.60        | 87.54        |
| <i>PTB-biomedical</i> | 95.05 | 95.26        | <b>95.46</b> | 94.43        |
| PTB-chemistry         | 90.32 | <b>90.60</b> | 90.56        | 90.58        |
| CHO-broadcast         | 78.38 | <b>78.42</b> | 78.27        | 78.28        |
| CHO-magazines         | 78.50 | <b>78.57</b> | 76.80        | 78.29        |
| CHO-weblogs           | 79.64 | <b>79.76</b> | 79.24        | 79.37        |
| CDT-law               | 93.96 | <b>95.64</b> | 93.91        | 94.25        |
| CDT-literature        | 93.93 | <b>94.19</b> | 94.15        | 94.15        |
| CDT-magazines         | 94.95 | <b>95.06</b> | 94.71        | 95.04        |
| Wilcoxon $p$          |       | <0.01        |              |              |
| macro-av. err.red     |       | 4.0          | -1.2         | -0.2         |

Table 1: Results (in %).

gle out the newspaper section as training data and use held-out newspaper data for development.

We observe two characteristics about our datasets: (a) The class distributions are relatively stable across domains. For CDT, for example, we see almost identical distributions of parts of speech, except literature has more prepositions. (b) The OOV rate is significantly higher across domains than within domains. This holds even for the PTB datasets, where the OOV rate is 14.6% on the biomedical test data, but 43.3% on the chemistry test data. These two observations confirm that cross-domain data is primarily biased by covariate shifts.

All learning algorithms do the same number of passes over each training data set. The number of iterations was set optimizing baseline system performance on development data. For EWT and CHO, we do 10 passes over the data. For PTB, we do 15 passes over the data, and for CDT, we do 25 passes over the data. The deletion rate in adversarial learning was fixed to 0.1% (optimized on the EWT emails data; not optimized on PTB, CHO or CDT). In  $L_\infty$ -regularization, the parameter  $C$  was optimized the same way and set to 20. Results are averages over five runs.

#### 4.1 Results

The results are presented in Table 1. Learning with antagonistic adversaries performs significantly better than structured perceptron (SP) learning,  $L_\infty$ -regularization and LRA across the board. We follow Demsar (2006) in computing significance across datasets using a Wilcoxon signed rank test. This is a strong result given that our algorithm is as computationally efficient as SP and does not pool unlabeled data to adapt to a specific target distribution. What we see is that let-

ting an antagonistic adversary corrupt our labeled data - somewhat surprisingly, maybe - leads to better cross-domain performance.  $L_\infty$ -regularization leads to worse performance, and LRA performs very similar to SP on average. Improvements to LRA have also been explored in Trafalis and Gilbert (2007) and Dekel and Shamir (2008). We note that on the in-domain dataset (PTB-biomedical),  $L_\infty$ -regularization performs best, but our approach also performs better than the structured perceptron baseline on this dataset.

#### 4.2 Analysis

The number of zero weights or very small weights is significantly lower for learning with antagonistic adversaries than for the baseline structured perceptron. So our models become less sparse. On the other hand, we have more parameters with average weights in our models. Weights are in other words better distributed. We also observe that parameters are updated slightly more with antagonistic adversaries. In our PTB experiments, for example, the mean weight is 14.2 in structured perceptron learning, but 14.5 with antagonistic adversaries. On the other hand, weight variance is slightly lower; recall the connection to variance regularization (Bergsma et al., 2010). Note that  $L_\infty$ -regularization with  $C = 20$  corresponds to clipping all weights above 20, i.e. roughly a third of the weights in this case. To validate our intuitions about what is going on, we also tried to increase the deletion rate. If  $\delta$  is increased to 1%, the mean weight goes up to 19.2. The adversarial model is less sparse than the baseline model.

A last observation is that the structured perceptron baseline model expectedly fits the training data better than the robust models. On CDT, the structured perceptron has an accuracy of 98.26% on held-out training data, whereas our model has an accuracy of only 97.85%. The  $L_\infty$ -regularized has an accuracy of 97.82%, whereas LRA has an accuracy of 98.18%.

### 5 Conclusion

We presented a discriminative learning algorithms for cross-domain structured prediction that seems more robust to covariate shifts than previous approaches. Our approach was superior to previous approaches across 12 multilingual cross-domain POS tagging datasets, with an average error reduction of 4% over a structured perceptron baseline.

## References

- Shane Bergsma, Dekang Lin, and Dale Schuurmans. 2010. Improved natural language learning via variance-regularization support vector machines. In *CoNLL*.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.
- Michael Collins. 2002. Discriminative training methods for Hidden Markov Models. In *EMNLP*.
- Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*.
- Ofer Dekel and Ohad Shamir. 2008. Learning to classify with missing and corrupted features. In *ICML*.
- Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *ICML*.
- George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*.
- Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Josef Le Roux, Joakim Nivre, Deirde Hogan, and Josef van Genabith. 2011. From news to comments: Resources and benchmarks for parsing the language of Web 2.0. In *IJCNLP*.
- Yoav Freund and Robert Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296.
- Laurent El Ghaoui and Herve Le Bret. 1997. Robust solutions to least-squares problems with uncertain data. In *SIAM Journal of Matrix Analysis and Applications*.
- Amir Globerson and Sam Roweis. 2006. Nightmare at test time: robust learning by feature deletion. In *ICML*.
- Geoffrey Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. <http://arxiv.org/abs/1207.0580>.
- Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. 2009. Group lasso with overlap and graph lasso. In *ICML*.
- Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.
- Andre Martins, Noah Smith, Pedro Aguiar, and Mario Figueiredo. 2011. Structured sparsity in structured prediction. In *EMNLP*.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *EMNLP-CoNLL*.
- Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. CoRR abs/1104.2086.
- Mark Schmidt and Kevin Murphy. 2010. Convex structure learning in log-linear models: beyond pairwise potentials. In *AISTATS*.
- Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing weight undertraining in structured discriminative learning. In *NAACL*.
- T Trafalis and R Gilbert. 2007. Robust support vector machines for classification and computational issues. *Optimization Methods and Software*, 22:187–198.
- Huan Xu, Constantine Caramanis, and Shie Mannor. 2009. Robustness and regularization of support vector machines. In *JMLR*.

# Temporal Signals Help Label Temporal Relations

Leon Derczynski and Robert Gaizauskas

Natural Language Processing Group

Department of Computer Science

University of Sheffield

211 Portobello, S1 4DP, Sheffield, UK

{leon, robertg}@dcs.shef.ac.uk

## Abstract

Automatically determining the temporal order of events and times in a text is difficult, though humans can readily perform this task. Sometimes events and times are related through use of an explicit co-ordination which gives information about the temporal relation: expressions like “before” and “as soon as”. We investigate the rôle that these co-ordinating temporal signals have in determining the type of temporal relations in discourse. Using machine learning, we improve upon prior approaches to the problem, achieving over 80% accuracy at labelling the types of temporal relation between events and times that are related by temporal signals.

## 1 Introduction

It is important to understand time in language. The ability to express and comprehend expressions of time enables us to plan, to tell stories, and to discuss change in the world around us.

When we automatically extract temporal information, we are often concerned with events and times – referred to collectively as temporal **intervals**. We might ask, for example, “*Who is the current President of the USA?*.” In order to extract an answer to this question from a document collection, we need to identify events related to persons becoming president and the times of those events. Crucially, however, we also need to identify the temporal relations between these events and times, perhaps, for example, by recognizing a temporal relation type from a set such as that of Allen (1983). This last task, **temporal relation typing**, is challenging, and is the focus of this paper.

Temporal **signals** are words or phrases that act as discourse markers that co-ordinate a pair of events or times and explicitly state the nature of the temporal relation that holds between them. For example, in “*The parade reached the town hall before noon*”, the word *before* is a temporal signal, co-ordinating the event *reached* with the time *noon*. Intuitively, these signal

words act as discourse contain temporal ordering information that human readers can readily access, and indeed this hypothesis is borne out empirically (Bestgen and Vonk, 1999). In this paper, we present an in-depth examination into the role temporal signals can play in machine learning for temporal relation typing, within the framework of TimeML (Pustejovsky et al., 2005).

## 2 Related Work

Temporal relation typing is not a new problem. Classical work using TimeML is that of Boguraev and Ando (2005), Mani et al. (2007) and Yoshikawa et al. (2009). The TempEval challenge series features relation typing as a key task (Verhagen et al., 2009). The take-home message from all this work is that temporal relation typing is a hard problem, even using advanced techniques and extensive engineering – approaches rarely achieve over 60% on typing relations between two events or over 75% accuracy for those between an event and a time. Recent attempts to include more linguistically sophisticated features representing discourse, syntactic and semantic role information have yielded but marginal improvements, e.g. Llorens et al. (2010); Mirroshandel et al. (2011).

Although we focus solely on determining the *types* of temporal relations, one must also identify which pairs of temporal intervals should be temporally related. Previous work has covered the tasks of identifying and typing temporal relations jointly with some success (Denis and Muller, 2011; Do et al., 2012). The TempEval3 challenge addresses exactly this task (Uz-Zaman et al., 2013).

Investigations into using signals for temporal relation typing have had promising results. Lapata and Lascarides (2006) learn temporal structure according to these explicit signals, then predict temporal orderings in sentences without signals. As part of an early TempEval system, Min et al. (2007) automatically annotate signals and associate them with temporal relations. They then include the signal text as a feature for a relation type classifier. Their definition of signals varies somewhat from the traditional TimeML sig-

|                                   | Event-event relations |           |         | Event-time relations |           |         |
|-----------------------------------|-----------------------|-----------|---------|----------------------|-----------|---------|
|                                   | Non-signalled         | Signalled | Overall | Non-signalled        | Signalled | Overall |
| Baseline most-common-class        | 41.4%                 | 57.4%     | 43.0%   | 49.2%                | 51.6%     | 49.6%   |
| Maxent classifier                 | 57.7%                 | 58.6%     | 57.8%   | 81.4%                | 59.6%     | 77.3%   |
| <i>Error reduction</i>            | 27.8%                 | 2.74%     | 25.4%   | 64.5%                | 16.4%     | 55.5%   |
| Sample size (number of relations) | 3 179                 | 343       | 3 522   | 2 299                | 529       | 2 828   |

Table 1: Relation typing performance using the base feature set, for relations with and without a temporal signal.

nal definition, as they include words such as *reporting* which would otherwise be annotated as an event. The system achieves a 22% error reduction on a simplified set of temporal relation types.

Later, Derczynski and Gaizauskas (2010) saw a 50% error reduction in assignment of relation types on signalled relation instances from introducing simple features describing a temporal signal’s interaction with the events or times that it co-ordinates. The features for describing signals included the signal text itself and the signal’s position in the document relative to the intervals it co-ordinated. This led to a large increase in relation typing accuracy to 82.19% for signalled event-event relations, using a maximum entropy classifier.

Previous work has attempted to linguistically characterise temporal signals (Brée et al., 1993; Derczynski and Gaizauskas, 2011). Signal phrases typically fall into one of three categories: monosemous as temporal signals (e.g. “*during*”, “*when*”); bisemous as temporal or spatial signals (e.g. “*before*”); or polysemous with the temporal sense a minority class (e.g. “*in*”, “*following*”). Further, a signal phrase may take two arguments, though its arguments need not be in the immediate content and may be anaphoric. We leave the task of automatic signal annotation to future work, instead focusing on the impact that signals have on temporal relation typing.

Our work builds on previous work by expanding the study to include relations other than just event-event relations, by extending the feature set, by doing temporal relation labelling over a more carefully curated version of the TimeBank corpus (see below), and by providing detailed analysis of the performance of a set of labelling techniques when using temporal signals.

### 3 Experimental Setup

We only approach the relation typing task, and we use existing signal annotations – that is, we do not attempt to automatically identify temporal signals.

The corpus used is the signal-curated version of TimeBank (Pustejovsky et al., 2003). This corpus, TB-sig,<sup>1</sup> adds extra events, times and relations to TimeBank, in an effort to correct signal under-annotation in the original corpus (Derczynski and Gaizauskas, 2011). Like the original TimeBank corpus, it comprises 183 documents. In these, we are interested only in the temporal relations that use a signal. There are 851 signals annotated in the corpus, co-ordinating 886 temporal re-

lations (13.7% of all). For comparison, TimeBank has 688 signal annotations which co-ordinate 718 temporal relations (11.2%).

When evaluating classifiers, we performed 10-fold cross-validation, keeping splits at document level. There are only 14 signalled time-time relations in this corpus, which is not enough to support any generalizations, and so we disregard this interval type pairing.

As is common with statistical approaches to temporal relation typing, we also perform relation folding; that is, to reduce the number of possible classes, we sometimes invert argument order and relation type. For example, A BEFORE B and B AFTER A convey the same temporal relation, and so we can remove all AFTER-type relations by swapping their argument order and converting them to BEFORE relations. This lossless process condenses the labels that our classifier has to distinguish between, though classification remains a multi-class problem.

We adopt the base feature set of Mani et al. (2007), which consists mainly of TimeML event and time annotation surface attributes. These are, for events: class, aspect, modality, tense, polarity, part of speech; and, for times: value, type, function in document, mod, quant. To these are added same-tense and same-aspect features, as well as the string values of events/times.

The feature groups we use here are:

- **Base** – The attributes of TimeML annotations involved (includes tense, aspect, polarity and so on as above), as with previous approaches.
- **Argument Ordering** – Two features: a boolean set if both arguments are in the same sentence (as in Chambers et al. (2007)), and the text order of argument intervals (as in Hepple et al. (2007)).
- **Signal Ordering** – Textual ordering is important with temporal signals; compare “*You walk before you run*” and “*Before you walk you run*”. We add features accounting for relative textual position of signal and arguments as per Derczynski and Gaizauskas (2010). To these we add a feature reporting whether the signal occurs in first, last, or mid-sentence position, and features to indicate whether each interval is in the same sentence as the signal.
- **Syntactic** – We add syntactic features: following Bethard et al. (2007), the lowest common constituent label between each argument and

<sup>1</sup>See [http://derczynski.com/sheffield/resources/tb\\_sig.tar.bz2](http://derczynski.com/sheffield/resources/tb_sig.tar.bz2)

| Features | Classifier                  | Event-event accuracy | Event-time accuracy |
|----------|-----------------------------|----------------------|---------------------|
| N/A      | Baseline most-common-class  | 57.4%                | 51.6%               |
| Base     | Baseline maximum entropy    | 58.6%                | 59.6%               |
| DG2010   | Maximum entropy             | 72.6%                | 72.4%               |
|          | Random forest               | 76.7%                | 78.6%               |
| All      | Adaptive boosting           | 70.4%                | 73.0%               |
|          | Naïve Bayes                 | 73.8%                | 71.5%               |
|          | Maximum entropy             | 75.5%                | 78.1%               |
|          | Linear SVC / Crammer-Singer | 79.3%                | 75.6%               |
|          | Linear SVC                  | 80.7%                | 77.1%               |
|          | Random forest               | <b>80.8%</b>         | <b>80.3%</b>        |

Table 2: Results at temporal relation typing over TB-sig, for relations that use a temporal signal

the signal; following Swampillai and Stevenson (2011), the syntactic path from each argument to the signal, using a top-level `ROOT` node for cross-sentence paths; and three features indicating whether there is a temporal function tag (`-TMP` between each of the intervals or the signal to the root node). These features are generated using the Stanford parser (Klein and Manning, 2003) and a function tagger (Blaheta and Charniak, 2000).

- **Signal Text** – We add the signal’s raw string, as well as its lower-case version and its lemma.
- **DCT** – For event-time relations, whether the time expression also functions as the document’s creation timestamp.

Collectively, these feature groups comprise the **All** feature set. For comparison, the feature set we reported in previous work (Derczynski and Gaizauskas, 2010) is also included, labeled **DG2010**. This set contains the base and the signal ordering feature groups only, plus a single signal feature for the signal raw string.

Using these feature representations we trained multinomial naïve Bayes (Rennie et al., 2003), maximum entropy (Daumé III, 2008), adaptive boosting (Freund and Schapire, 1997; Zhu et al., 2009), multi-class SVM (Crammer and Singer, 2002; Chang and Lin, 2011) and random forest<sup>2</sup> (Breiman, 2001) classifiers via Scikit-learn (Pedregosa et al., 2011).

We use two baselines: most-common-class and a model trained with no signal features. We also introduce two measures replicating earlier work: one using the DG2010 features and the classifier used in that work (maximum entropy), and another using the DG2010 features with the best-performing classifier under our All feature set, in order to see if performance changes are due to features or classifier.

Classifiers were evaluated by determining if the class they output matched the relation type in TB-sig. Results are given in Table 2. For comparison with the general case, i.e. for both signalled and non-signalled temporal relation instances, we list performance with a maximum entropy classifier and the base feature set

<sup>2</sup>With  $n_{estimators} = 200$ , a minimum of one sample per node, and no maximum depth.

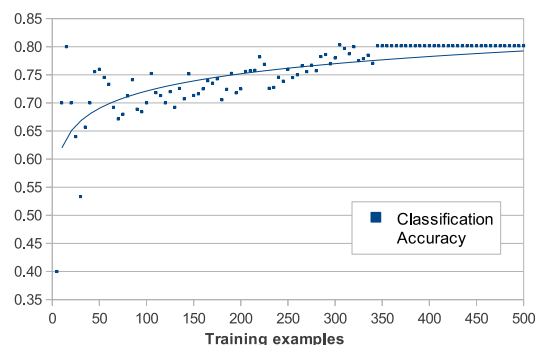


Figure 1: Effect of training data size on relation typing performance.

on TB-sig’s temporal relations. Results are in Table 1. These are split into those that use a signal and those that do not, though no features relating signal information are included.

In order to assess the adequacy of the dataset in terms of size, we also examined performance using a maximum entropy classifier learned from varying sub-proportions of the training data. This was measured over event-event relations, using all features. Results are given in Figure 1. That performance appears to stabilise and level off indicates that the training set is of sufficient size for these experiments.

## 4 Analysis

The results in Table 2 echo earlier findings and intuition: temporal signals are useful in temporal relation typing. Results support that signals are not only helpful in event-event relation typing but also event-time typing. For comparison, inter-annotator agreement across all temporal relation labels, i.e. signalled and non-signalled relations, in TimeBank is 77%.

Using the maximum entropy classifier, our approach gives a 2.9% absolute performance increase over the DG2010 feature set for event-event relations (10.6% error reduction) and a 5.7% absolute increase for event-time relations (20.7% error reduction). Random forests

| Feature sets                  | Evt-evt      | Evt-time     |
|-------------------------------|--------------|--------------|
| All                           | 80.8%        | 80.3%        |
| All-argument order            | 80.8%        | 78.3%        |
| All-signal order              | 79.0%        | 77.5%        |
| All-syntax                    | 79.2%        | 79.6%        |
| All-signal text               | <b>70.8%</b> | <b>72.7%</b> |
| All-DCT                       | 79.9%        | 79.4%        |
| Base                          | 54.2%        | 53.9%        |
| Base+argument order           | 56.8%        | 60.1%        |
| Base+signal order             | 59.7%        | 65.0%        |
| Base+syntax                   | 70.0%        | <b>71.0%</b> |
| Base+signal text              | <b>75.5%</b> | 66.3%        |
| Base+DCT                      | 54.2%        | 53.9%        |
| Base+signal text+signal order | 80.4%        | 76.9%        |
| Base+signal text+syntax       | 79.0%        | 74.1%        |
| Base+arg order+signal order   | 77.8%        | 75.2%        |

Table 3: Relation typing accuracy based on various feature combinations, using random forests. Bold figures indicate the largest performance change.

offer better performance under both feature sets, with the extended features achieving notable error reduction over DG2010 – 17.6% for event-event, 7.9% for event-time relations. Linear support vector classification provided rapid labelling and comparable performance for event-event relations but accuracy was not as good as random forests for event-time relation labelling.

Note, figures reported earlier in Derczynski and Gaizauskas (2010) are not directly comparable to the DG2010 figures reported here, as here we are using the better-annotated TB-sig corpus, which contains a larger and more varied set of temporal signal annotations.

Although we are only examining the 13.7% of temporal relations that are co-ordinated with a signal, it is important to note the performance of conventional classification approaches on this subset of temporal relations. Specifically, the error reduction relative to the baseline that is achieved without signal features is much lower on relations that use signals than on non-signalled relations (Table 1). Thus, temporal relations that use a signal appear to be more difficult to classify than other relations, unless signal information is present in the features. This may be due to differences in how signals are used by authors. One explanation is that signals may be used in the stead of temporal ordering information in surrounding discourse, such as modulations of dominant tense or aspect (Derczynski and Gaizauskas, 2013).

Unlike earlier work using maxent, we experiment with a variety of classifiers, and find a consistent improvement in temporal relation typing using signal features. With the notable exception of adaptive boosting, classifiers with preference bias (Liu et al., 2002) – AdaBoost, random trees and SVC – performed best in this task. Conversely, those tending toward the independence assumption (naïve Bayes and maxent) did not capitalise as effectively on the training data.

| Features                       | Evt-evt      | Evt-time     |
|--------------------------------|--------------|--------------|
| All                            | 80.8%        | 80.3%        |
| All-signal text                | 70.8%        | 72.7%        |
| All-signal text-argument order | 70.7%        | 72.2%        |
| All-signal text-signal order   | 69.5%        | 71.2%        |
| All-signal text-syntax         | <b>59.5%</b> | <b>69.0%</b> |
| All-signal text-DCT            | 70.8%        | 72.8%        |

Table 4: Feature ablation without signal text features. Bold figures indicate largest performance change.

We also investigated the impact of each feature group on the best-performing classifier (random forests with  $n = 200$ ) through feature ablation. Results are given in Table 3. Ablation suggested that the signal text features (signal string, lower case string, head word and lemma) had most impact in event-event relation typing, though were second to syntax features in event-time relations. Removing other feature groups gave only minor performance decreases.

We also experimented with adding feature groups to the base set one-by-one. All but DCT features gave above-baseline improvement, though argument ordering features were not very helpful for event-event relation typing. Signal text features gave the strongest improvement over baseline for event-event relations, but syntax gave a larger improvement for event-time relations. Accordingly, it may be useful to distinguish between event-event and event-time relations when extracting temporal information using syntax (c.f. the approach of Wang et al. (2010)).

A strong above-baseline performance was still obtained even when signal text features were removed, which included the signal text itself. This was interesting, as signal phrases can indicate quite different temporal orderings (e.g. “Open the box *while* it rains” vs. “Open the box *before* it rains”, and the words used are typically critical to correct interpretation of the temporal relation. Further, the model is able to generalise beyond particular signal phrase choices. To investigate further, we examined the performance impact of each group sans “signal text” features (Table 4). In this case, removing the syntactic features had the greatest (negative) impact on performance, though the absolute impact on event-event relations (a drop of 11.3%) was far lower than that on event-time relations (3.7%).

To examine helpful features, we trained a maxent classifier on the entire dataset and collected feature:value pairs. These were then ranked by their weight. The ten largest-weighted pairings for event-event relations (the hardest problem in overall temporal relation typing) are given in Table 5. Prefixes of 1- and 2- correspond to the two interval arguments (events). Negative values are those where the presence of a particular feature:value pair suggests the mentioned class is not applicable.



| Weight | Feature         | Value    | Class    |
|--------|-----------------|----------|----------|
| 9.346  | 2-polarity      | POS      | ENDS     |
| -8.713 | 1-2-same-sent   | True     | BEGINS   |
| -7.861 | 2-aspect        | NONE     | BEGINS   |
| -7.256 | 1-aspect        | NONE     | INCLUDES |
| 6.564  | 2-sig-synt-path | NN-NP-IN | INCLUDES |
| 6.519  | signal-lower    | before   | ENDS     |
| -6.294 | 2-tense         | NONE     | BEGINS   |
| -5.908 | 2-modality      | None     | ENDS     |
| 5.643  | 2-text          | took     | BEGINS   |
| -5.580 | 1-modality      | None     | ENDS     |

Table 5: Top ten largest-weighted feature:value pairs.

It can be seen that BEGINS and INCLUDES relationships are not indicated if the arguments have no TimeML aspect assigned; this is what one might expect, given how aspect is used in English, with these temporal relation types corresponding to event starts and the progressive. Also, notice how a particular syntactic path, connecting adjacent nominalised event and the word *in* acting as a signal, indicate a temporal inclusion relationship. Temporal polysemy, where a word has more than one possible temporal interpretation, is also observable here (Derczynski and Gaizauskas (2011) examine this polysemy in depth). This is visible in how the temporal signal phrase “before” is not, as one might expect, a strong indicator of a BEFORE or even AFTER relation, but of an ENDS relationship.

## 5 Conclusion

This paper set out to investigate the rôle of temporal signals in predicting the type of temporal relation between two intervals. The paper demonstrated the utility of temporal signals in this task, and identified approaches for using the information these signals contain, which performed consistently better than the state-of-the-art across a range of machine learning classifiers. Further, it identified the impact that signal text, signal order and syntax features had in temporal relation typing of signalled relations.

Two directions of future work are indicated. Firstly, the utility of signals prompts investigation into detecting which words in a given text occur as temporal signals. Secondly, it is intuitive that temporal signals explicitly indicate related pairs of intervals (i.e. events or times). So, the task of deciding which interval pair(s) a temporal signal co-ordinates must be approached.

Although we have found a method for achieving good temporal relation typing performance on a subset of temporal relations, the greater problem of general temporal relation typing remains. A better understanding of the semantics of events, times, signals and how they are related together through syntax may provide further insights into the temporal relation typing task.

Finally, Bethard et al. (2007) reached high temporal relation typing performance on one a subset of relations

(events and times in the same sentence); we reach high temporal relation typing performance on another subset of relations – those using a temporal signal. Identifying further explicit sources of temporal information applicable to new sets of relations may reveal promising paths for investigation.

## Acknowledgements

The first author was supported by UK EPSRC grant EP/K017896/1, uComp (<http://www.ucomp.eu/>).

## References

- J. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Y. Bestgen and W. Vonk. 1999. Temporal adverbials as segmentation markers in discourse comprehension. *Journal of Memory and Language*, 42(1):74–87.
- S. Bethard, J. Martin, and S. Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *Proceedings of the International Conference on Semantic Computing*, pages 11–18.
- D. Blaheta and E. Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of the meeting of the North American chapter of the Association for Computational Linguistics*, pages 234–240. ACL.
- B. Boguraev and R. K. Ando. 2005. TimeBank-Driven TimeML Analysis. In G. Katz, J. Pustejovsky, and F. Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, number 05151 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- D. Brée, A. Feddag, and I. Pratt. 1993. Towards a formalization of the semantics of some temporal prepositions. *Time & Society*, 2(2):219.
- L. Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.
- N. Chambers, S. Wang, and D. Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th meeting of the Association for Computational Linguistics*, pages 173–176. ACL.
- C.-C. Chang and C.-J. Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27.
- K. Crammer and Y. Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292.
- H. Daumé III. 2008. MegaM: Maximum entropy model optimization package. *ACL Data and Code Repository*, ADCLR2008C003, 50.

- P. Denis and P. Muller. 2011. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1788–1793. AAAI Press.
- L. Derczynski and R. Gaizauskas. 2010. Using Signals to Improve Automatic Classification of Temporal Relations. In *Proceedings of 15th Student Session of the European Summer School for Logic, Language and Information*, pages 224–231. FoLLI.
- L. Derczynski and R. Gaizauskas. 2011. A Corpus-based Study of Temporal Signals. In *Proceedings of the Corpus Linguistics Conference*.
- L. Derczynski and R. Gaizauskas. 2013. Empirical Validation of Reichenbach’s Tense Framework. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 71–82. ACL.
- Q. X. Do, W. Lu, and D. Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 677–687. ACL.
- Y. Freund and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.
- M. Hepple, A. Setzer, and R. Gaizauskas. 2007. USFD: preliminary exploration of features and classifiers for the TempEval-2007 tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 438–441. ACL.
- D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st meeting of the Association for Computational Linguistics*, pages 423–430. ACL.
- M. Lapata and A. Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27(1):85–117.
- Y. Liu, Y. Yang, and J. Carbonell. 2002. Boosting to correct inductive bias in text classification. In *Proceedings of the 11th international Conference on Information and Knowledge Management*, pages 348–355. ACM.
- H. Llorens, E. Saquete, and B. Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of SemEval-2010*. ACL.
- I. Mani, B. Wellner, M. Verhagen, and J. Pustejovsky. 2007. Three approaches to learning TLINKS in TimeML. Technical report, CS-07-268, Brandeis University.
- C. Min, M. Srikanth, and A. Fowler. 2007. LCC-TE: A hybrid approach to temporal relation identification in news text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 219–222. ACL.
- S. A. Mirroshandel, G. Ghassem-Sani, and M. Khayyamian. 2011. Using syntactic-based kernels for classifying temporal relations. *Journal of Computer Science and Technology*, 26(1):68–80.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.
- J. Pustejovsky, R. Sauri, R. Gaizauskas, A. Setzer, L. Ferro, et al. 2003. The TimeBank Corpus. In *Proceedings of the Corpus Linguistics Conference*, pages 647–656.
- J. Pustejovsky, J. Castano, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. 2005. TimeML: Robust specification of event and temporal expressions in text. In I. Mani, J. Pustejovsky, and R. Gaizauskas, editors, *The language of time: a reader*. Oxford University Press.
- J. D. Rennie, L. Shih, J. Teevan, and D. Karger. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the International Conference on Machine Learning*. AAAI Press.
- K. Swampillai and M. Stevenson. 2011. Extracting relations within and across sentences. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 25–32. ACL.
- N. UzZaman, H. Llorens, L. Derczynski, M. Verhagen, J. F. Allen, and J. Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the 7th International Workshop on Semantic Evaluations*.
- M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, J. Moszkowicz, and J. Pustejovsky. 2009. The TempEval challenge: identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179.
- W. Wang, J. Su, and C. L. Tan. 2010. Kernel based discourse relation recognition with temporal ordering information. In *Proceedings of the 48th meeting of the Association for Computational Linguistics*, pages 710–719. ACL.
- K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto. 2009. Jointly identifying temporal relations with Markov logic. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 405–413. ACL.
- J. Zhu, H. Zou, S. Rosset, and T. Hastie. 2009. Multi-class AdaBoost. *Statistics and Its Interface*, 2:349–360.

# Diverse Keyword Extraction from Conversations

**Maryam Habibi**

Idiap Research Institute and EPFL  
Rue Marconi 19, CP 592  
1920 Martigny, Switzerland  
maryam.habibi@idiap.ch

**Andrei Popescu-Belis**

Idiap Research Institute  
Rue Marconi 19, CP 592  
1920 Martigny, Switzerland  
andrei.popescu-belis@idiap.ch

## Abstract

A new method for keyword extraction from conversations is introduced, which preserves the diversity of topics that are mentioned. Inspired from summarization, the method maximizes the coverage of topics that are recognized automatically in transcripts of conversation fragments. The method is evaluated on excerpts of the Fisher and AMI corpora, using a crowd-sourcing platform to elicit comparative relevance judgments. The results demonstrate that the method outperforms two competitive baselines.

## 1 Introduction

The goal of keyword extraction from texts is to provide a set of words that are representative of the semantic content of the texts. In the application intended here, keywords are automatically extracted from transcripts of conversation fragments, and are used to formulate queries to a just-in-time document recommender system. It is thus important that the keyword set preserves the diversity of topics from the conversation. While the first keyword extraction methods ignored topicality as they were based on word frequencies, more recent methods have considered topic modeling factors for keyword extraction, but without specifically setting a topic diversity constraint, which is important for naturally-occurring conversations.

In this paper, we propose a new method for keyword extraction that rewards both word similarity, to extract the most representative words, and word diversity, to cover several topics if necessary. The paper is organized as follows. In Section 2 we review existing methods for keyword extraction. In Section 3 we describe our proposal, which relies on topic modeling and a novel topic-aware diverse keyword extraction algorithm. Section 4 presents

the data and tasks for comparing sets of keywords. In Section 5 we show that our method outperforms two existing ones.

## 2 State of the Art in Keyword Extraction

Numerous studies have been conducted to automatically extract keywords from a text or a transcribed conversation. The earliest techniques have used word frequencies (Luhn, 1957), TFIDF values (Salton et al., 1975; Salton and Buckley, 1988), and pairwise word co-occurrence frequencies (Matsuo and Ishizuka, 2004) to rank words for extraction. These approaches do not consider word meaning, so they may ignore low-frequency words which together indicate a highly-salient topic (Nenkova and McKeown, 2012).

To improve over frequency-based methods, several ways to use lexical semantic information have been proposed. Semantic relations between words can be obtained from a manually-constructed thesaurus such as WordNet, or from Wikipedia, or from an automatically-built thesaurus using latent topic modeling techniques. Ye et al. (2007) used the frequency of all words belonging to the same WordNet concept set, while the Wikifier system (Csomai and Mihalcea, 2007) relied on Wikipedia links to compute a substitute to word frequency. Harwath and Hazen (2012) used topic modeling with PLSA to build a thesaurus, which they used to rank words based on topical similarity to the topics of a transcribed conversation. To consider dependencies among selected words, word co-occurrence has been combined with PageRank by Mihalcea and Tarau (2004), and additionally with WordNet by Wang et al. (2007), or with topical information by Z. Liu et al. (2010). However, as shown empirically by Mihalcea and Tarau (2004) and by Z. Liu et al. (2010) with various co-occurrence windows, such approaches have difficulties modeling long-range dependencies between words related to the same

topic. Z. Liu et al. (2009b) used part-of-speech information and word clustering techniques, while F. Liu et al. (2009a) added this information to the TFIDF method so as to consider both word dependency and semantic information. However, although they considered topical similarity, the above methods did not explicitly reward diversity and might miss secondary topics.

Supervised methods have been used to learn a model for extracting keywords with various learning algorithms (Turney, 1999; Frank et al., 1999; Hulth, 2003). These approaches, however, rely on the availability of in-domain training data, and the objective functions they use for learning do not consider yet the diversity of keywords.

### 3 Diverse Keyword Extraction

We propose to build a topical representation of a conversation fragment, and then to select keywords using topical similarity while also rewarding the diversity of topic coverage, inspired by recent summarization methods (Lin and Bilmes, 2011; Li et al., 2012).

#### 3.1 Representing Topic Information

Topic models such as Probabilistic Latent Semantic Analysis (PLSA) or Latent Dirichlet Allocation (LDA) can be used to determine the distribution over the topic  $z$  of a word  $w$ , noted  $p(z|w)$ , from a large amount of training documents. LDA implemented in the Mallet toolkit (McCallum, 2002) is used in this paper because it does not suffer from the overfitting issue of PLSA (Blei et al., 2003).

The distribution of each topic  $z$  in a given conversation fragment  $t$ , noted  $p(z|t)$ , can be computed by summing over all probabilities  $p(z|w)$  of the  $N$  words  $w$  spoken in the fragment:

$$p(z|t) = \frac{1}{N} \sum_{w \in t} p(z|w).$$

#### 3.2 Selecting Keywords

The problem of keyword extraction with maximal topic coverage is formulated as follows. If a conversation fragment  $t$  mentions a set of topics  $Z$ , and each word  $w$  from the fragment  $t$  can evoke a subset of the topics in  $Z$ , then the goal is to find a subset of unique words  $S \subseteq t$ , with  $|S| \leq k$ , which maximizes the number of covered topics for each number of keywords  $k$ .

This problem is an instance of the maximum coverage problem, which is  $NP$ -hard. Nemhauser

et al. (1978) showed that a greedy algorithm can find an approximate solution guaranteed to be within  $(1 - \frac{1}{e}) \simeq 0.63$  of the optimal solution if the coverage function is submodular and monotone nondecreasing<sup>1</sup>.

To find a monotone submodular function for keyword extraction, we used inspiration from recent work on extractive summarization methods (Lin and Bilmes, 2011; Li et al., 2012), which proposed a square root function for diverse selection of sentences to cover the maximum number of key concepts of a given document. The function rewards diversity by increasing the gain of selecting a sentence including a concept that was not yet covered by a previously selected sentence. This must be adapted for keyword extraction by defining an appropriate reward function.

We first introduce  $r_{S,z}$ , the topical similarity with respect to topic  $z$  of the keyword set  $S$  selected from the fragment  $t$ , defined as follows:

$$r_{S,z} = \sum_{w \in S} p(z|w) \cdot p(z|t).$$

We then propose the following reward function for each topic, where  $p(z|t)$  is the importance of the topic and  $\lambda$  is a parameter between 0 and 1:

$$f : r_{S,z} \rightarrow p(z|t) \cdot r_{S,z}^\lambda.$$

This is clearly a submodular function with diminishing returns as  $r_{S,z}$  increases.

Finally, the keywords  $S \subseteq t$ , with  $|S| \leq k$ , are chosen by maximizing the cumulative reward function over all the topics, formulated as follows:

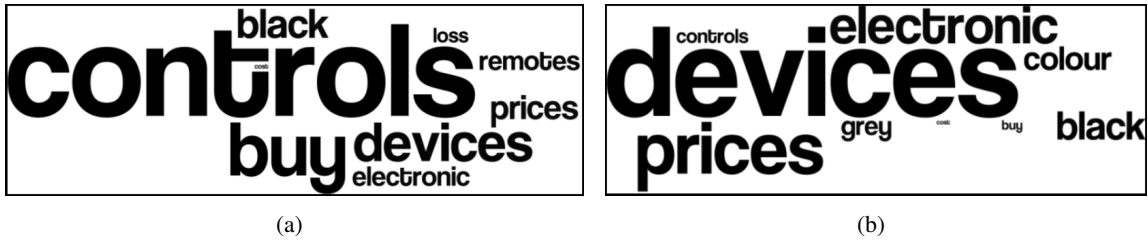
$$R(S) = \sum_{z \in Z} p(z|t) \cdot r_{S,z}^\lambda.$$

Since  $R(S)$  is submodular, the greedy algorithm for maximizing  $R(S)$  is shown as Algorithm 1 on the next page, with  $r_{\{w\},z}$  being similar to  $r_{S,z}$  with  $S = \{w\}$ . If  $\lambda = 1$ , the reward function is linear and only measures the topical similarity of words with the main topics of  $t$ . However, when  $0 < \lambda < 1$ , as soon as a word is selected from a topic, other words from the same topic start having diminishing gains.

## 4 Data and Evaluation Method

The proposed keyword extraction method was tested on two conversational corpora, the Fisher

<sup>1</sup>A function  $F$  is *submodular* if  $\forall A \subseteq B \subseteq T \setminus t, F(A+t) - F(A) \geq F(B+t) - F(B)$  (diminishing returns) and is *monotone nondecreasing* if  $\forall A \subseteq B, F(A) \leq F(B)$ .



- Please select one of the following options:
1. Image (a) represents the conversation fragment better than (b).
  2. Image (b) represents the conversation fragment better than (a).
  3. Both (a) and (b) offer a good representation of the conversation.
  4. None of (a) and (b) offer a good representation of the conversation.

Figure 1: Example of a HIT based on an AMI discussion about the impact on sales of some features of remote controls (the conversation transcript is given in the Appendix). The word cloud was generated using Wordle™ from the list produced by the diverse keyword extraction method with  $\lambda = 0.75$  (noted D(.75)) for image (a) and by a topic similarity method (TS) for image (b). TS over-represents the topic “color” by selecting three words related to it, but misses other topics such as “remote control”, “losing a device” and “buying a device” which are also representative of the fragment.

**Input** : a given text  $t$ , a set of topics  $Z$ , the number of keywords  $k$   
**Output**: a set of keywords  $S$   
 $S \leftarrow \emptyset$ ;  
**while**  $|S| \leq k$  **do**  
     $S \leftarrow S \cup \{ \text{argmax}_{w \in t \setminus S} (h(w)) \text{ where } h(w) = \sum_{z \in Z} p(z|t)[r_{\{w\},z} + r_{S,z}]^\lambda \}$ ;  
**end**  
**return**  $S$ ;

**Algorithm 1:** Diverse keyword extraction.

Corpus (Cieri et al., 2004), and the AMI Meeting Corpus (Carletta, 2007). The former corpus contains about 11,000 topic-labeled telephone conversations, on 40 pre-selected topics (one per conversation). We created a topic model using Mallet over two thirds of the Fisher Corpus, given its large number of single-topic documents, with 40 topics. The remaining data is used to build 11 artificial “conversations” (1-2 minutes long) for testing, by concatenating 11 times three fragments about three different topics.

The AMI Corpus contains 171 half-hour meetings about remote control design, which include several topics each – so they cannot be directly used for learning topic models. While selecting for testing 8 conversation fragments of 2-3 minutes each, we trained topic models on a subset of the English Wikipedia (10% or 124,684 articles). Following several previous studies, the number of

topics was set to 100 (Boyd-Graber et al., 2009; Hoffman et al., 2010).

To evaluate the relevance (or representativeness) of extracted keywords with respect to a conversation fragment, we designed comparison tasks. In each task, a fragment is shown, followed by three control questions about its content, and then by two lists of nine keywords each, from two different extraction methods. To improve readability, the keyword lists are presented to the judges using a word cloud representation generated by Wordle™ (<http://www.wordle.net>), in which the words ranked higher are emphasized in the word cloud (see example in Figure 1). The judges had to read the conversation transcript, answer the control questions, and then decide which word cloud better represents the content of the conversation.

The tasks were crowdsourced via Amazon’s Mechanical Turk (AMT) as “human intelligence tasks” (HITs). One of them is exemplified in Figure 1, without the control questions, and the respective conversation transcript is given in the Appendix. Ten workers were recruited for each corpus. An example of judgment counts for each of the 8 AMI HITs comparing two methods is shown in Table 1. After collecting judgments, the comparative relevance values were computed by first applying a qualification control factor to the human judgments, and then averaging results over all judgments (Habibi and Popescu-Belis, 2012).

Moreover, to verify the diversity of the key-

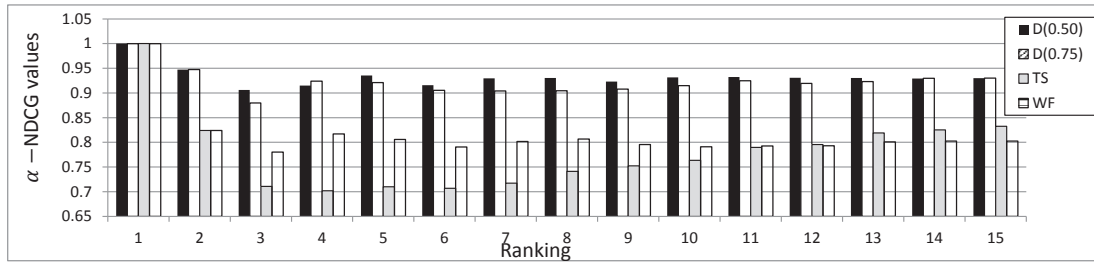


Figure 2: Average  $\alpha$ -NDCG over the 11 conversations from the Fisher Corpus, for 1 to 15 extracted keywords.

word set, we use the  $\alpha$ -NDCG measure (Clarke et al., 2008) proposed for information retrieval, which rewards a mixture of relevance and diversity – with equal weights when  $\alpha = .5$  as set here. We only apply  $\alpha$ -NDCG to the three-topic conversation fragments from the Fisher Corpus, relevance of a keyword being set to 1 when it belongs to the fragment corresponding to the topic. A higher value indicates that keywords are more uniformly distributed across the three topics.

## 5 Experimental Results

We have compared several versions of the diverse keyword extraction method, noted  $D(\lambda)$ , for  $\lambda \in \{.5, .75, 1\}$ , with two other methods. The first one uses only word frequency (not including stopwords) and is noted WF. We did not use TFIDF because it sets low weights on keywords that are repeated in many fragments but which are nevertheless important to extract. The second method is based on topical similarity (noted TS) but does not specifically enforce diversity (Harwath and Hazen, 2012). In fact TS coincides with  $D(1)$ , so it is noted TS. As the relevance of keywords for  $D(.5)$  was already quite low, we did not test lower values of  $\lambda$ . Similarly, we did not test additional values of  $\lambda$  above  $.5$  because the resulting word lists were very similar to tested values.

First of all, we compared the four methods with respect to the diversity constraint over the con-

| HIT                | A | B | C | D | E | F | G | H |
|--------------------|---|---|---|---|---|---|---|---|
| TS more relevant   | 4 | 1 | 1 | 1 | 2 | 2 | 1 | 1 |
| $D(.75)$ more rel. | 4 | 1 | 8 | 9 | 6 | 6 | 6 | 8 |
| Both relevant      | 2 | 5 | 1 | 0 | 2 | 2 | 3 | 1 |
| Both irrelevant    | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 1: Number of answers for each of the four options of the comparative evaluation task, from ten human judges. The 8 HITs compare the  $D(.75)$  and TS methods on 8 AMI HITs.

| Corpus | Compared methods ( $m_1$ vs. $m_2$ ) | Relevance (%) |       |
|--------|--------------------------------------|---------------|-------|
|        |                                      | $m_1$         | $m_2$ |
| Fisher | $D(.75)$ vs. TS                      | 68            | 32    |
|        | TS vs. WF                            | 82            | 18    |
|        | WF vs. $D(.5)$                       | 95            | 5     |
| AMI    | $D(.75)$ vs. TS                      | 78            | 22    |
|        | TS vs. WF                            | 60            | 40    |
|        | WF vs. $D(.5)$                       | 78            | 22    |

Table 2: Comparative relevance scores of keyword extraction methods based on human judgments.

catenated fragments of the Fisher Corpus, by using  $\alpha$ -NDCG to measure how evenly the extracted keywords were distributed across the three topics. Figure 2 shows results averaged over 11 conversations for various sizes of the keyword set (1–15). The average  $\alpha$ -NDCG values for  $D(.75)$  and  $D(.5)$  are similar, and clearly higher than WF and TS for all ranks (except, of course, for a single keyword). The values for TS are quite low, and only increase for a large number of keywords, demonstrating that TS does not cope well with topic diversity, but on the contrary first selects keywords from the dominant topic. The values for WF are more uniform as it does not consider topics at all.

To measure the overall representativeness of keywords, we performed binary comparisons between the outputs of each method, using crowdsourcing, over 11 fragments from the Fisher Corpus and 8 fragments from AMI. The goal is to rank the methods, so we only report here on the comparisons required for complete ordering. AMT workers compared two lists of nine keywords each, with four options:  $X$  more representative or relevant than  $Y$ , or vice-versa, or both relevant, or both irrelevant. Table 1 shows the judgments collected when comparing the output of  $D(.75)$  with TS on the AMI Corpus. Workers disagreed for the first two HITs, but then found that the keywords extracted by  $D(.75)$  were more representative compared to TS. The consolidated rel-

evance (Habibi and Popescu-Belis, 2012) is 78% for D(.75) vs. 22% for TS.

The averaged relevance values for all comparisons needed to rank the four methods are shown in Table 2 separately for the Fisher and AMI Corpora. Although the exact differences vary, the human judgments over the two corpora both indicate the following ranking:  $D(.75) > TS > WF > D(.5)$ . The optimal value of  $\lambda$  is thus around .75, and with this value, our diversity-aware method extracts more representative keyword sets than TS and WF. The differences between methods are larger for the Fisher Corpus, due to the artificial fragments that concatenate three topics, but they are still visible on the natural fragments of the AMI Corpus. The low scores of D(.5) are found to be due, upon inspection, to the low relevance of keywords. In particular, the comparative relevance of D(.75) vs. D(.5) on the Fisher Corpus is very large (96% vs. 4%).

## 6 Conclusion

The diverse keyword extraction method with  $\lambda = .75$  provides the keyword sets that are judged most representative of the conversation fragments (two conversational datasets) by a large number of human judges recruited via AMT, and has the highest  $\alpha$ -NDCG value. Therefore, enforcing both relevance and diversity brings an effective improvement to keyword extraction.

Setting  $\lambda$  for a new dataset remains an issue, and requires a small development data set. However, preliminary experiments with a third dataset showed that  $\lambda = .75$  remains a good value.

In the future, we will use keywords to retrieve documents from a repository and recommend them to conversation participants by formulating topically-separate queries.

### Appendix: Conversation transcript of AMI ES2005a meeting (00:00:5-00:01:52)

The following transcript of a four-party conversations (speakers noted A through D) was submitted to our keyword extraction method and a baseline one, generating respectively the two word clouds shown in Figure 1.

A: The only the only remote controls I've used usually come with the television, and they're fairly basic. So uh

D: Yeah. Yeah.

C: Mm-hmm.

D: Yeah, I was thinking that as well, I think the the only ones that I've seen that you buy are the sort of one for all type things where they're, yeah. So presumably that might be an idea to

C: Yeah the universal ones. Yeah.

A: Mm. But but to sell it for twenty five you need a lot of neat features. For sure.

D: put into.

C: Yeah.

D: Yeah, yeah. Uh 'cause I mean, what uh twenty five Euros, that's about I dunno, fifteen Pounds or so?

C: Mm-hmm, it's about that.

D: And that's quite a lot for a remote control.

A: Yeah, yeah.

C: Mm. Um well my first thoughts would be most remote controls are grey or black. As you said they come with the TV so it's normally just your basic grey black remote control functions, so maybe we could think about colour? Make that might make it a bit different from the rest at least. Um, and as you say, we need to have some kind of gimmick, so um I thought maybe something like if you lose it and you can whistle, you know those things?

D: Uh-huh. Mm-hmm. Okay. The the keyrings, yeah yeah. Okay, that's cool.

C: Because we always lose our remote control.

B: Uh yeah uh, being as a Marketing Expert I will like to say like before deciding the cost of this remote control or any other things we must see the market potential for this product like what is the competition in the market? What are the available prices of the other remote controls in the prices? What speciality other remote controls are having and how complicated it is to use these remote controls as compared to other remote controls available in the market.

D: Okay.

B: So before deciding or before finalising this project, we must discuss all these things, like and apart from this, it should be having a good look also, because people really uh like to play with it when they are watching movies or playing with or playing with their CD player, MP three player like any electronic devices. They really want to have something good, having a good design in their hands, so, yes, all this.

## Acknowledgments

The authors are grateful to the Swiss National Science Foundation for its financial support through the IM2 NCCR on Interactive Multimodal Information Management (see [www.im2.ch](http://www.im2.ch)).

## References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Jonathan Boyd-Graber, Jordan Chang, Sean Gerrish, Chong Wang, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*.
- Jean Carletta. 2007. Unleashing the killer corpus: Experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation Journal*, 41(2):181–190.
- Christopher Cieri, David Miller, and Kevin Walker. 2004. The Fisher Corpus: a resource for the next generations of speech-to-text. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)*, pages 69–71.
- Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666.
- Andras Csomai and Rada Mihalcea. 2007. Linking educational materials to encyclopedic knowledge. *Frontiers in Artificial Intelligence and Applications*, 158:557.
- Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 1999)*, pages 668–673, Stockholm, Sweden.
- Maryam Habibi and Andrei Popescu-Belis. 2012. Using crowdsourcing to compare document recommendation strategies for conversations. In *Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2011)*, page 15.
- David Harwath and Timothy J. Hazen. 2012. Topic identification based extrinsic evaluation of summarization techniques applied to conversational speech. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5073–5076. IEEE.
- Matthew D. Hoffman, David M. Blei, and Francis Bach. 2010. Online learning for Latent Dirichlet Allocation. *Proceedings of 24th Annual Conference on Neural Information Processing Systems*, 23:856–864.
- Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 216–223, Sapporo, Japan.
- Jingxuan Li, Lei Li, and Tao Li. 2012. Multi-document summarization via submodularity. *Applied Intelligence*, 37(3):420–430.
- Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the ACL*.
- Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009a. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL (HLT-NAACL)*, pages 620–628.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 257–266.
- 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 (EMNLP 2010)*, pages 366–376.
- Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317.
- Yutaka Matsuo and Mitsuru Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169.
- Andrew K. McCallum. 2002. MALLET: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 404–411, Barcelona.
- George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming Journal*, 14(1):265–294.
- Ani Nenkova and Kathleen McKeown. 2012. *A Survey of Text Summarization Techniques*, chapter 3, pages 43–76. Springer.
- Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management Journal*, 24(5):513–523.



- Gerard Salton, Chung-Shu Yang, and Clement T. Yu. 1975. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44.
- Peter Turney. 1999. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council Canada (NRC).
- Jinghua Wang, Jianyi Liu, and Cong Wang. 2007. Keyword extraction based on PageRank. In *Advances in Knowledge Discovery and Data Mining (Proceedings of PAKDD 2007)*, LNAI 4426, pages 857–864. Springer-Verlag, Berlin.
- Shiren Ye, Tat-Seng Chua, Min-Yen Kan, and Long Qiu. 2007. Document concept lattice for text understanding and summarization. *Information Processing and Management*, 43(6):1643–1662.

# Understanding Tables in Context Using Standard NLP Toolkits

Vidhya Govindaraju    Ce Zhang    Christopher Ré

University of Wisconsin-Madison  
{vidhya, czhang, chrisre}@cs.wisc.edu

## Abstract

Tabular information in text documents contains a wealth of information, and so tables are a natural candidate for information extraction. There are many cues buried in both a table and its surrounding text that allow us to understand the meaning of the data in a table. We study how natural-language tools, such as part-of-speech tagging, dependency paths, and named-entity recognition, can be used to improve the quality of relation extraction from tables. In three domains we show that (1) a model that performs joint probabilistic inference across tabular and natural language features achieves an F1 score that is twice as high as either a pure-table or pure-text system, and (2) using only shallower features or non-joint inference results in lower quality.

## 1 Introduction

Tabular data is ubiquitous and often contains high-quality, structured relational data. Recent studies found billions of high-quality relations on the web in HTML (Cafarella et al., 2008). In financial applications, a huge amount of data is buried in the tables of corporate filings and earnings reports; in science, millions of journal articles contain billions of scientific facts in tables. Although tables describe precise, structured relations, tables are rarely written in a way that is self-describing, e.g., tables may contain abbreviations or only informal schema information; in turn, the contents of tables are often ambiguously specified, which makes extracting the relations implicit in tabular data difficult.

Tables are, however, not written in isolation. The text surrounding a table in a jour-

nal article explains its contents to its intended audience, a human reader. For example, in a simple study, we demonstrate that humans can achieve more than 60% higher recall by jointly reading the text and tables in a journal article than by only looking at the tables. The conclusion of this experiment is not surprising, but it raises a question: *How should a system combine tabular and natural-language features to understand tables in text?*

The literature provides a broad spectrum of answers to this question. Most previous approaches use textual or tabular features separately, e.g., tabular approaches that do not use text features (Dalvi et al., 2012; Wu and Lee, 2006; Pinto et al., 2003) or textual approaches that do not use tabular features (Mintz et al., 2009; Wu and Weld, 2010; Poon and Domingos, 2007). In a prescient study, Liu et al. (2007) proposed to learn the target relation independently from both table and surface textual features, and then combine the result using a linear combination of the predictions.

In a similar spirit, we propose to use both types of features in our approach of relation extraction. Our proposed approach differs from prior approaches in two ways: (1) We use deeper—but standard—NLP features than prior approaches for table extraction. In contrast to the shallow, lexical features that prior approaches have used, we use standard NLP features, such as dependency paths, parts of speech, etc. Our hypothesis is that a deeper understanding of the text in which a table is embedded will lead to higher quality table extraction. (2) Our probabilistic model jointly uses both tabular and textual features. One advantage of a joint approach is that one can predict portions of the complicated predicate that is buried in a table. For example, in a geology journal article, we may read a measure-

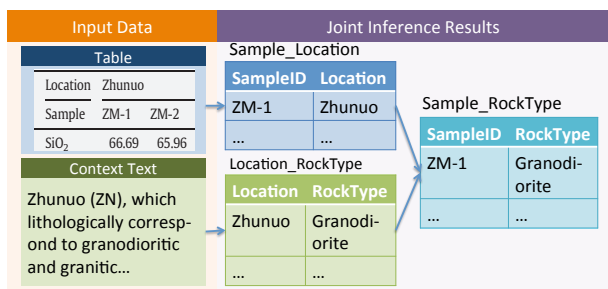


Figure 1: An example of joint inference between a table and its context.

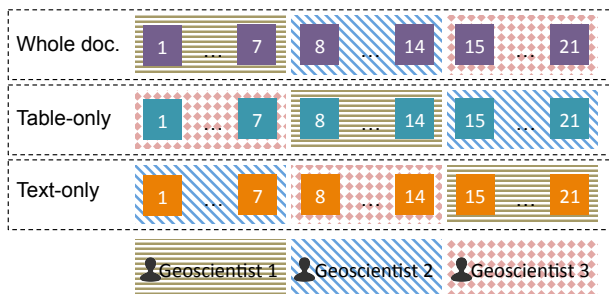


Figure 2: Job assignments for the human study.

ment in a table that tells us the type of rock and its weight—but data such as the location where this rock was unearthed and in what geological time interval this rock appeared may not be specified in the table.

We consider tasks in three domains: PETROLOGY, FINANCE, and GEOLOGY. For each domain, we build a system to extract relations from text, tables, or both. We found that a joint inference system that uses non-shallow, but standard NLP features can significantly improve the quality of the extracted relations, and *that this result holds consistently across all three domains*. For example, in our Petrology application to extract a knowledge base, called PETDB<sup>1</sup>, by using information extracted from both text and tables, we can achieve twice as high F1 compared to either a pure-table or pure-text system.

## 2 Motivating Human Study

We describe a simple human study that motivated our approach to jointly combine both tabular features and natural language features to extract relations from tables. The hypoth-

<sup>1</sup><http://www.earthchem.org/petdb>

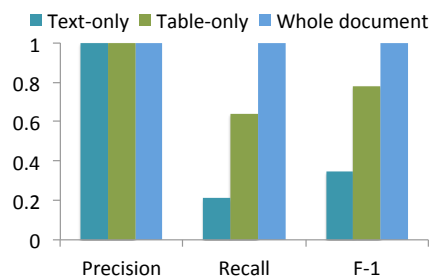


Figure 3: Human quality to extract SAMPLE-ROCKTYPE relations in PETDB.

| Task | TEXT                          | TABLE                         | JOINT  |
|------|-------------------------------|-------------------------------|--|
| NER  | POS tags                      | pdftotable                    | Whether a mention in table also appears in the text. |
|      | Stanford NER                  | NER of neighbor cells         |  |
| EL   | Regular Expression Dictionary | Regular expression Dictionary | Subjective mentions in the sentence near a table     |
|      | Freebase                      | Freebase                      |  |
| RE   | Stanford Parser               | Freebase                      | Join between relations (See Figure 1 for an example) |
|      | Dependency path               | Table headers                 |  |
|      | Term proximity                | Table subheaders              |  |
|      | Word sequence                 | RE of neighbor rows           |  |

Figure 4: List of features we used in TEXT, TABLE, and JOINT approaches. **NER**, **EL**, and **RE** refer to named-entity recognition, entity linking, and relation extraction, respectively.

esis that we want to validate is that the text surrounding a table could provide valuable information even for a human reader, and therefore, an ideal machine reading system should also try to capture similar information.

We asked three geoscientists to manually read journal articles and extract relations for the PETROLOGY domain. We report our results for the target relation, SAMPLE-ROCKTYPE, which associates a rock type with a rock sample (see Figure 1 for an example). We randomly sampled 21 journal articles. For each journal article, we produced three variants: (1) the original document; (2) *table-only*, which is the set of tables in the document (without the text); (3) *text-only*, which is the text of the document with the tables removed from the document. Each geoscientist was asked to read and extract the relations from one of the three variants. We then judged the precision and recall of their extraction, as shown in Figure 2.

As shown in Figure 3, human readers not surprisingly achieve perfect precision on each of the variants, but lower recall on both the table-only and text-only variants. However, summing the recall of table-only (60%) and text-only (20%) variants together would achieve only 80% recall; this implies that in the best case more than 20% of the extractions require that the human reader read the table *and* its surrounding text *jointly*. Figure 1 shows one representative example.

This motivates our approach, which uses a *joint* inference system to model features from a table and its surrounding text. We also propose to use deep linguistic features instead of shallower features to get as close as possible to the ability of human readers in understanding the surrounding text of a table.

### 3 Empirical Study & Experiments

We describe our experiments to test the hypothesis that (1) deeper linguistic features can help to extract higher quality relations from tables, and (2) joint inference across tables and text improves extraction quality compared to approaches that use pure-table, pure-text, and non-joint ways of combining these two. We briefly describe some experiments for a dataset that we call GEOLOGY (Zhang et al., 2013). The detailed experimental results in all three domains are in the technical report version of this paper.

#### 3.1 Experimental Setup

We consider the task of constructing a geology knowledge base. Specifically, our goal is to extract a ROCK-TOTALORGANICCARBON relation that maps rock formations (e.g., “Barnett Formation”) to their total organic carbon (e.g., “6%”). Such data is important for estimating stored energy and for global climate research.

**Dataset.** We selected 100 geology journal articles.<sup>2</sup> We asked three geoscientists to annotate these journal articles manually to extract the ROCK-TOTALORGANICCARBON relation (1.5K tuples). We processed each document using Stanford CoreNLP (de Marneffe et al., 2006; Toutanova and Manning, 2000),

<sup>2</sup>We choose a set of documents that (1) are in English, and (2) contain at least one table.

PDFtoHTML<sup>3</sup>, and pdf2table (Yildiz, 2004). We then extracted features following state-of-the-art practices (see Figure 4).

**Approaches.** To validate our hypothesis, we implement four systems, each of which has access to different types of data:

(1) TABLE. This approach follows Pinto et al. (2003) and Dalvi et al. (2012) and only uses the tables in a document.

(2) TEXT. This approach only has access to the text in a document and contains all the features mentioned in Wu and Weld (2010) and Mintz et al. (2009).

The features used in (1) and (2) are shown in Figure 4. In both TABLE and TEXT, we use a conditional random field (Lafferty et al., 2001) model for the ROCK-TOTALORGANICCARBON relation.

(3) MERGE. Using TABLE and TEXT, we extract all facts and their associated probability. Following Duin (2002), we combine these two probabilities using a linear combination. MERGE is a baseline approach that uses information from both tables and text.

(4) JOINT. We build a joint approach that uses information from both tables and text. This approach is a large factor graph in which we embed the CRFs developed in TABLE and TEXT. Additionally, we allow JOINT to predict projections of each relation, as shown in Figure 4. Recall that a key advantage of a joint approach is that we do not need to predict all arguments of the relation (if such a prediction is unwarranted from the data). The inference is done by Gibbs sampling using our inference engine ELEMENTARY (Zhang and Ré, 2013). We describe the JOINT system in more detail in the technical report version of this paper.

#### 3.2 End-to-End Quality

We were able to validate that JOINT achieves higher quality than the other three approaches we considered. Figure 5 shows the P/R curve of different approaches on three domains. We analyzed the domain GEOLOGY.

JOINT dominates all other approaches. At a recall of 10%, JOINT achieves 3x higher precision than all other approaches. In our error analysis, we saw that tables in geology articles often contain ambiguous words; for example,

<sup>3</sup><http://pdf-tohtml.sourceforge.net/>

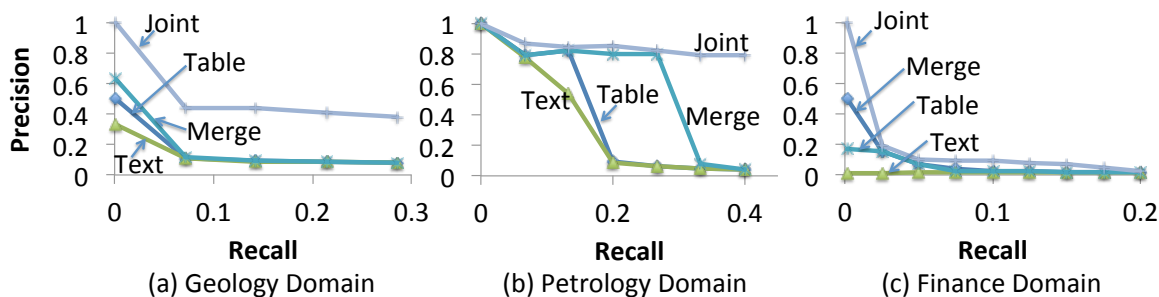


Figure 5: End-to-end extraction quality on PETROLOGY, FINANCE, and GEODEEPDIVE. The recall is limited by the quality of state-of-the-art table recognition software on PDFs.

the word “Barnett” in a table may refer to either a location or a rock formation. By using features extracted from text, JOINT achieves higher precision. For recall in the range of 0–10%, MERGE outperforms both TEXT and TABLE, with 3%–90% improvement in precision.

In GEOLOGY, MERGE has precision that is similar to TEXT and TABLE for the higher recall range (>10%). In this domain, we found that relations that appeared in the text often repeated relations described in the table. In other domains, such as PETROLOGY, where the relations in text and tables have lower degrees of overlap, MERGE significantly improves over TEXT and TABLE (Figure 5(b)).

We conducted a statistical significance test to check whether the improvement of JOINT over the three other approaches is statistically significant. For each of the three probability thresholds,  $t \in \{.99, .90, .50\}$ , we created the set of predictions that JOINT assigns probability greater than  $t$ . Figure 6 shows the results of the statistical significance test in which the null hypothesis is that the F1 scores of two approaches are the same. With  $p = 0.01$ , JOINT has statistically significant improvement of F1 score over all three other approaches with each probability threshold.

### 3.3 Shallow vs. Linguistic Features

We validate the hypothesis that using linguistic features, e.g., part-of-speech tags (Toutanova and Manning, 2000), named-entity tags (Finkel et al., 2005), and dependency trees (de Marneffe et al., 2006), helps improve the quality of our approach, called JOINT. There are different ways to use shallow and linguistic features; we select

| Approaches \ Prob. | .99 | .90 | .50 |
|--------------------|-----|-----|-----|
| TEXT               | +   | +   | +   |
| TABLE              | +   | +   | +   |
| MERGE              | +   | +   | +   |

Figure 6: Approximate randomization test from Chinchor (1992) of F1 score with  $p = 0.01$  on the impact of joint inference compared with pure-table or pure-text approaches for different probability thresholds. A + sign indicates that the F1 score of joint approach increased significantly.

| Type       | Features   |
|------------|--|
| Shallow    | Regular Expressions (Dalvi et al., 2012)<br>Term proximity (Matsuo et al., 2003)<br>Dictionary and Freebase (Mintz et al., 2009) |
| Linguistic | POS tags (Wu et al., 2010)<br>Stanford NER tags (Mintz et al., 2009)<br>Dependence trees (Mintz et al., 2009)                    |

Figure 7: Types of Features.

state-of-the-art approaches from the literature (see Figure 7).

We created the following variants of JOINT.  $\text{JOINT}^{(-\text{PARSE})}$  removes features generated by the dependency parser and syntax parser. Similarly,  $\text{JOINT}^{(-\text{NER})}$  ( $\text{JOINT}^{(-\text{POS})}$ ) removes all features related to NER (resp. POS).  $\text{JOINT}^{(-\text{POS})}$  also removes NER and parser features because the latter two are dependent on POS features.

Figure 8 shows the P/R curve for all these variants on GEOLOGY, and Figure 9 shows the results of statistical significance test. For probability threshold .90, JOINT outperforms  $\text{JOINT}^{(-\text{POS})}$  significantly. The difference between JOINT,  $\text{JOINT}^{(-\text{PARSE})}$ ,

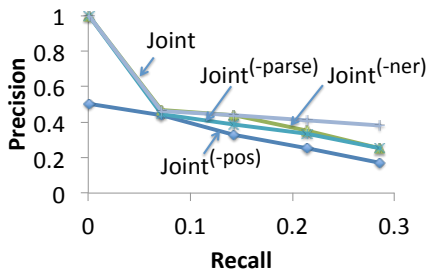


Figure 8: Lesion study of different features for GEOLOGY.

| Features \ Prob.                  | .90 | .50 |
|-----------------------------------|-----|-----|
| JOINT <sup>(-PARSE)</sup> → JOINT | 0   | +   |
| JOINT <sup>(-NER)</sup> → JOINT   | 0   | +   |
| JOINT <sup>(-POS)</sup> → JOINT   | +   | +   |

Figure 9: Approximate randomization test of F1 score with  $p = 0.01$  on the impact of linguistic features. For  $x \rightarrow y$ , a + indicates that the F1 score of  $y$  is significantly higher than  $x$ . 0 indicates that the F1 score does not change significantly.

and JOINT<sup>(-NER)</sup> is not significant because there are “easy-to-extract” facts in the high-probability range. For probability threshold .50, JOINT outperforms all three other variants significantly.

## 4 Related Work

The intuition that context features might help table-related tasks has existed for decades. For example, Hurst and Nasukawa (2000) mentioned (as future work) that context features could be used to further improve their relation extraction approaches from tables. Lin et al. (2010) use bag-of-words features and hyperlinks to recommend new columns for web tables. Liu et al. (2007) extract features, including font size and title, from PDF documents in which a table appears to help the table ranking task. They find that these features only contribute less than 2% to precision. In contrast, in our approach linguistic features are quite useful. The above approaches use context features that can be extracted without POS tagging or linguistic parsing. One aspect of our work is to demonstrate that traditional NLP tools can enhance the quality of table extraction.

Extracting information from tables has been discussed by different communities in the last decade, including NLP (Wu and Lee, 2006; Tengli et al., 2004; Chen et al., 2000), artificial intelligence (Fang et al., 2012; Pivk, 2006), information retrieval (Wei et al., 2006; Pinto et al., 2003), database (Cafarella et al., 2008), and the web (Dalvi et al., 2012). This body of work considers only features derived from tables and does not examine richer NLP features as we do.

While joint inference is popular, it is not clear when a joint inference system outperforms a more traditional NLP pipeline. Recent studies have reached a variety of conclusions: in some, joint inference helps extraction quality (McCallum, 2009; Poon and Domingos, 2007; Singh et al., 2009); and in some, joint inference hurts extraction quality (Poon and Domingos, 2007; Eisner, 2009). Our intuition is that joint inference is helpful in this application because our joint inference approach combines non-redundant signals (textual versus tabular).

## 5 Conclusion

To improve the quality of extractions of tabular data, we use standard NLP techniques to more deeply understand the text in which a table is embedded. We validate that deeper NLP features combined with a joint probabilistic model has a statistically significant impact on quality, i.e., recall and precision. Our ongoing work is to apply these ideas to a much larger corpus from each of the three domains.

## 6 Acknowledgments

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) DEFT Program under Air Force Research Laboratory (AFRL) prime contract No. FA8750-13-2-0039, the National Science Foundation EAGER Award under No. EAR-1242902 and CAREER Award under No. IIS-1054009, and the Sloan Research Fellowship. 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, AFRL, NSF, or the US government. We are also grateful to Jude W. Shavlik for his insightful comments.

## References

- Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. WebTables: Exploring the power of tables on the web. *Proceedings of VLDB Endowment*, 1(1).
- Hsin-Hsi Chen, Shih-Chung Tsai, and Jin-He Tsai. 2000. Mining tables from large scale HTML texts. In *Proceedings of the 18th Conference on Computational Linguistics*, COLING '00.
- Nancy Chinchor. 1992. The statistical significance of the MUC-4 results. In *Proceedings of the 4th Conference on Message Understanding*, MUC4 '92.
- Bhavana Bharat Dalvi, William Cohen, and Jamie Callan. 2012. WebSets: Extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM '12.
- 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*.
- Robert Duin. 2002. The combining classifier: to train or not to train? In *16th International Conference on Pattern Recognition*.
- Jason Eisner. 2009. Joint models with missing data for semi-supervised learning. In *NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*.
- Jing Fang, Prasenjit Mitra, Zhi Tang, and C. Lee Giles. 2012. Table header detection and classification. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, AAAI '12.
- Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05.
- Matthew Hurst and Tetsuya Nasukawa. 2000. Layout and language: Integrating spatial and linguistic knowledge for layout understanding tasks. In *Proceedings of the 18th Conference on Computational Linguistics*, COLING '00.
- 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.
- Cindy Xide Lin, Bo Zhao, Tim Wenginger, Jiawei Han, and Bing Liu. 2010. Entity relation discovery from web tables and links. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10.
- Ying Liu, Kun Bai, Prasenjit Mitra, and C. Lee Giles. 2007. TableSeer: Automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '07.
- Andrew McCallum. 2009. Joint inference for natural language processing. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, CoNLL '09.
- 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*, ACL '09.
- David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. 2003. Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '03.
- Aleksander Pivk. 2006. Automatic ontology generation from web tabular structures. *AI Communication*, 19(1).
- Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, AAAI'07.
- Sameer Singh, Karl Schultz, and Andrew McCallum. 2009. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD '09.
- Ashwin Tengli, Yiming Yang, and Nian Li Ma. 2004. Learning table extraction from examples. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04.
- Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing*, EMNLP '00.
- Xing Wei, Bruce Croft, and Andrew McCallum. 2006. Table extraction for answer retrieval. *Information Retrieval*, 9(5).

- Dekai Wu and Ken Wing Kuen Lee. 2006. A grammatical approach to understanding textual tables using two-dimensional scfgs. In *Proceedings of the COLING/ACL, COLING-ACL '06*.
- Fei Wu and Daniel S. Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*.
- Burcu Yildiz. 2004. Information extraction – utilizing table patterns. Master’s thesis, Institut für Softwaretechnik und Interaktive Systeme.
- Ce Zhang and Christopher Ré. 2013. Towards high-throughput Gibbs sampling at scale: A study across storage managers. SIGMOD '13.
- Ce Zhang, Vidhya Govindaraju, Jackson Borhardt, Tim Foltz, Christopher Ré, and Shanan Peters. 2013. GeoDeepDive: Statistical inference using familiar data-processing languages. SIGMOD '13.



# Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction

Wei Xu<sup>+</sup> Raphael Hoffmann<sup>^</sup> Le Zhao<sup>#,\*</sup> Ralph Grishman<sup>+</sup>

<sup>+</sup>New York University, New York, NY, USA

{xuwei, grishman}@cs.nyu.edu

<sup>^</sup>University of Washington, Seattle, WA, USA

raphaelh@cs.washington.edu

<sup>#</sup>Google Inc., Mountain View, CA, USA

lezhao@google.com

## Abstract

Distant supervision has attracted recent interest for training information extraction systems because it does not require any human annotation but rather employs existing knowledge bases to heuristically label a training corpus. However, previous work has failed to address the problem of false negative training examples mislabeled due to the incompleteness of knowledge bases. To tackle this problem, we propose a simple yet novel framework that combines a passage retrieval model using coarse features into a state-of-the-art relation extractor using multi-instance learning with fine features. We adapt the information retrieval technique of pseudo-relevance feedback to expand knowledge bases, assuming entity pairs in top-ranked passages are more likely to express a relation. Our proposed technique significantly improves the quality of distantly supervised relation extraction, boosting recall from 47.7% to 61.2% with a consistently high level of precision of around 93% in the experiments.

## 1 Introduction

A recent approach for training information extraction systems is distant supervision, which exploits existing knowledge bases instead of annotated texts as the source of supervision (Craven and Kumlien, 1999; Mintz et al., 2009; Nguyen and Moschitti, 2011). To combat the noisy training data produced by heuristic labeling in distant supervision, researchers (Bunescu and Mooney, 2007; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012) exploited multi-instance

\*This work was done while Le Zhao was at Carnegie Mellon University.

learning models. Only a few studies have directly examined the influence of the quality of the training data and attempted to enhance it (Sun et al., 2011; Wang et al., 2011; Takamatsu et al., 2012). However, their methods are handicapped by the built-in assumption that a sentence does not express a relation unless it mentions two entities which participate in the relation in the knowledge base, leading to false negatives.

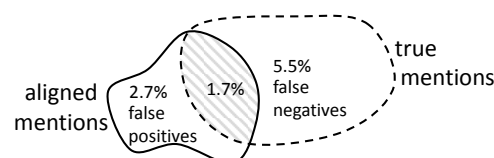


Figure 1: Noisy training data in distant supervision

In reality, knowledge bases are often incomplete, giving rise to numerous false negatives in the training data. We sampled 1834 sentences that contain two entities in the New York Times 2006 corpus and manually evaluated whether they express any of a set of 50 common Freebase<sup>1</sup> relations. As shown in Figure 1, of the 133 (7.3%) sentences that truly express one of these relations, only 32 (1.7%) are covered by Freebase, leaving 101 (5.5%) false negatives. Even for one of the most complete relations in Freebase, *Employee-of* (with more than 100,000 entity pairs), 6 out of 27 sentences with the pattern ‘PERSON executive of ORGANIZATION’ contain a fact that is not included in Freebase and are thus mislabeled as negative. These mislabelings dilute the discriminative capability of useful features and confuse the models. In this paper, we will show how reducing this source of noise can significantly improve the performance of distant supervision. In fact, our system corrects the relation labels of the above 6 sentences before training the relation extractor.

<sup>1</sup><http://www.freebase.com>

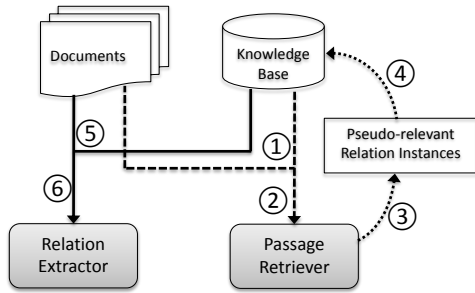


Figure 2: Overall system architecture: The system (1) matches relation instances to sentences and (2) learns a passage retrieval model to (3) provide relevance feedback on sentences; Relevant sentences (4) yield new relation instances which are added to the knowledge base; Finally, instances are again (5) matched to sentences to (6) create training data for relation extraction.

Encouraged by the recent success of simple methods for coreference resolution (Raghunathan et al., 2010) and inspired by pseudo-relevance feedback (Xu and Croft, 1996; Lavrenko and Croft, 2001; Matveeva et al., 2006; Cao et al., 2008) in the field of information retrieval, which expands or reformulates query terms based on the highest ranked documents of an initial query, we propose to increase the quality and quantity of training data generated by distant supervision for information extraction task using pseudo feedback. As shown in Figure 2, we expand an original knowledge base with possibly missing relation instances with information from the highest ranked sentences returned by a passage retrieval model (Xu et al., 2011) trained on the same data. We use coarse features for our passage retrieval model to aggressively expand the knowledge base for maximum recall; at the same time, we exploit a multi-instance learning model with fine features for relation extraction to handle the newly introduced false positives and maintain high precision.

Similar to iterative bootstrapping techniques (Yangarber, 2001), this mechanism uses the outputs of the first trained model to expand training data for the second model, but unlike bootstrapping it does not require iteration and avoids the problem of semantic drift. We further note that iterative bootstrapping over a single distant supervision system is difficult, because state-of-the-art systems (Surdeanu et al., 2012; Hoffmann et al., 2011; Riedel et al., 2010; Mintz et al., 2009), detect only few false negatives in the

training data due to their high-precision low-recall features, which were originally proposed by Mintz et al. (2009). We present a reliable and novel way to address these issues and achieve significant improvement over the MULTIR system (Hoffmann et al., 2011), increasing recall from 47.7% to 61.2% at comparable precision. The key to this success is the combination of two different views as in co-training (Blum and Mitchell, 1998): an information extraction technique with fine features for high precision and an information retrieval technique with coarse features for high recall. Our work is developed in parallel with Min et al. (2013), who take a very different approach by adding additional latent variables to a multi-instance multi-label model (Surdeanu et al., 2012) to solve this same problem.

## 2 System Details

In this section, we first introduce some formal notations then describe in detail each component of the proposed system in Figure 2.

### 2.1 Definitions

A *relation instance* is an expression  $r(e1, e2)$  where  $r$  is a binary *relation*, and  $e1$  and  $e2$  are two entities having such a relation, for example *CEO-of(Tim Cook, Apple)*. The knowledge-based distant supervised learning problem takes as input (1)  $\Sigma$ , a training corpus, (2)  $E$ , a set of entities mentioned in that corpus, (3)  $R$ , a set of relation names, and (4)  $\Delta$ , a set of ground facts of relations in  $R$ . To generate our training data, we further assume (5)  $T$ , a set of entity types, as well as type signature  $r(E1, E2)$  for relations.

We define the positive data set  $POS(r)$  to be the set of sentences in which any related pair of entities of relation  $r$  (according to the knowledge base) is mentioned. The negative data set  $RAW(r)$  is the rest of the training data, which contain two entities of the required types in the knowledge base, e.g. one person and one organization for the *CEO-of* relation in Freebase. Another negative data set with more conservative sense  $NEG(r)$  is defined as the set of sentences which contain the primary entity  $e1$  (e.g. person in any *CEO-of* relation in the knowledge base) and any secondary entity  $e2$  of required type (e.g. organization for the *CEO-of* relation) but the relation does not hold for this pair of entities in the knowledge base.

## 2.2 Distantly Supervised Passage Retrieval

We extend the learning-to-rank techniques (Liu, 2011) to distant supervision setting (Xu et al., 2011) to create a robust passage retrieval system. While relation extraction systems exploit rich and complex features that are necessary to extract the exact relation (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011), passage retrieval components use coarse features in order to provide different and complementary feedback to information extraction models.

We exploit two types of lexical features: Bag-Of-Words and Word-Position. The two types of simple binary features are shown in the following example:

**Sentence:** *Apple founder Steve Jobs died.*

**Target (Primary) entity:** *Steve Jobs*

**Bag-Of-Word features:** *'apple' 'founder' 'died' '*

**Word-Position features:** *'apple:-2' 'founder:-1' 'died:+1' '::+2'*

For each relation  $r$ , we assume each sentence has a binary relevance label to form distantly supervised training data: sentences in  $POS(r)$  are *relevant* and sentences in  $NEG(r)$  are *irrelevant*. As a pointwise learning-to-rank approach (Nallapati, 2004), the probabilities of relevance estimated by SVMs (Platt and others, 1999) are used for ranking all the sentences in the original training corpus for each relation respectively. We use LibSVM<sup>2</sup> (Chang and Lin, 2011) in our implementation.

## 2.3 Pseudo-relevance Relation Feedback

In the field of information retrieval, pseudo-relevance feedback assumes that the top-ranked documents from an initial retrieval are likely relevant, and extracts relevant terms to expand the original query (Xu and Croft, 1996; Lavrenko and Croft, 2001; Cao et al., 2008). Analogously, our assumption is that entity pairs that appear in more relevant and more sentences are more likely to express the relation, and can be used to expand knowledge base and reduce false negative noise in the training data for information extraction. We identify the most likely relevant entity pairs as follows:

<sup>2</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm>

**initialize**  $\Delta' \leftarrow \Delta$

**for** each relation type  $r \in R$  **do**

**learn** a passage (sentence) retrieval model  $L(r)$  using coarse features and  $POS(r) \cup NEG(r)$  as training data

**score** the sentences in the  $RAW(r)$  by  $L(r)$

**score** the entity pairs according to the scores of sentences they are involved in

**select** the top ranked pairs of entities, then add the relation  $r$  to their label in  $\Delta'$

**end for**

We select the entity pairs whose average score of the sentences they are involved in is greater than  $p$ , where  $p$  is a parameter tuned on development data.<sup>3</sup> The relation extraction model is then trained using  $(\Sigma, E, R, \Delta')$  with a more complete database than the original knowledge base  $\Delta$ .

## 2.4 Distantly Supervised Relation Extraction

We use a state-of-the-art open-source system, MULTIR (Hoffmann et al., 2011), as the relation extraction component. MULTIR is based on multi-instance learning, which assumes that at least one sentence of those matching a given entity-pair contains the relation of interest (Riedel et al., 2010) in the given knowledge base to tolerate false positive noise in the training data and superior than previous models (Riedel et al., 2010; Mintz et al., 2009) by allowing overlapping relations. MULTIR uses features which are based on Mintz et al. (2009) and consist of conjunctions of named entity tags, syntactic dependency paths between arguments, and lexical information.

## 3 Experiments

For evaluating extraction accuracy, we follow the experimental setup of Hoffmann et al. (2011), and use their implementation of MULTIR<sup>4</sup> with 50 training iterations as our baseline. Our complete system, which we call IRMIE, combines our passage retrieval component with MULTIR. We use the same datasets as in Hoffmann et al. (2011) and Riedel et al. (2010), which include 3-years of New York Times articles aligned with Freebase. The **sentential extraction** evaluation is performed on a small amount of manually annotated sentences, sampled from the union of matched sentences and

<sup>3</sup>We found  $p = 0.5$  to work well in practice.

<sup>4</sup><http://homes.cs.washington.edu/~raphaelh/mr/>

| Test Data Set | Original Test Set |             |             |                   | Corrected Test Set |             |             |                   |
|---------------|-------------------|-------------|-------------|-------------------|--------------------|-------------|-------------|-------------------|
|               | $\tilde{P}$       | $\tilde{R}$ | $\tilde{F}$ | $\Delta\tilde{F}$ | $\tilde{P}$        | $\tilde{R}$ | $\tilde{F}$ | $\Delta\tilde{F}$ |
| MULTIR        | 80.0              | 44.6        | 62.3        |                   | 92.7               | 47.7        | 70.2        |                   |
| IRMIE         | 84.6              | 56.1        | 70.3        | +8.0              | 92.6               | 61.2        | 76.9        | +6.7              |
| MULTIRLEX     | 91.8              | 43.0        | 67.4        |                   | 79.6               | 57.0        | 68.3        |                   |
| IRMIELEX      | 89.2              | 52.5        | 70.9        | +3.5              | 78.0               | 69.2        | 73.6        | +5.3              |

Table 1: Overall sentential extraction performance evaluated on the original test set of Hoffmann et al. (2011) and our corrected test set: Our proposed relevance feedback technique yields a substantial increase in recall.

system predictions. We define  $S^e$  as the sentences where some system extracted a relation and  $S^F$  as the sentences that match the arguments of a fact in  $\Delta$ . The sentential precision and recall is computed on a randomly sampled set of sentences from  $S^e \cup S^F$ , in which each sentence is manually labeled whether it expresses any relation in  $R$ .

Figure 3 shows the precision/recall curves for MULTIR with and without pseudo-relevance feedback computed on the test dataset of 1000 sentence used by Hoffmann et al. (2011). With the pseudo-relevance feedback from passage retrieval, IRMIE achieves significantly higher recall at a consistently high level of precision. At the highest recall point, IRMIE reaches 78.5% precision and 59.2% recall, for an F1 score of 68.9%.

Because the two types of lexical features used in our passage retrieval models are not used in MULTIR, we created another baseline MULTIRLEX by adding these features into MULTIR in order to rule out the improvement from additional information. Note that the sentences are sampled from the union of Freebase matches and sentences from which some systems in Hoffmann et al. (2011) extracted a relation. It underestimates the improvements of the newly developed systems in this paper. We therefore also created a new test set of 1000 sentences by sampling from the union of Freebase matches and sentences where MULTIRLEX or IRMIELEX extracted a relation. Table 1 shows the overall precision and recall computed against these two test datasets, with and without adding lexical features into multi-instance learning models. The performance improvement by using pseudo-feedback is significant ( $p < 0.05$ ) in McNemar’s test for both datasets.

#### 4 Conclusion and Perspectives

This paper proposes a novel approach to address an overlooked problem in distant supervision: the knowledge base is often incomplete causing nu-

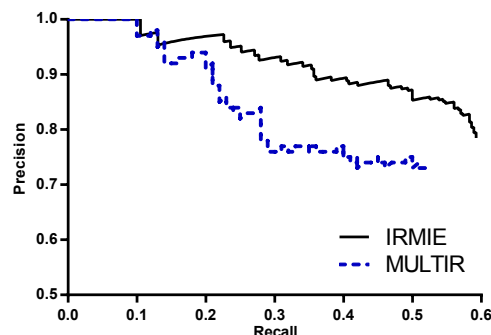


Figure 3: Sentential extraction: precision/recall curves using exact same training and test data, features and system settings as in Hoffmann et al. (2011).

merous false negatives in the training data. It greatly improves a state-of-the-art multi-instance learning model by correcting the most likely false negatives in the training data based on the ranking of a passage retrieval model.

In the future, we would like to more tightly integrate a coarser featured estimator of sentential relevance and a finer featured relation extractor, such that a single joint-model can be learned.

#### Acknowledgments

Supported in part by NSF grant IIS-1018317, the Air Force Research Laboratory (AFRL) under prime contract number FA8750-09-C-0181 and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20154. 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 AFRL, IARPA, DoI/NBC, or the U.S. Government.

## References

- Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100.
- Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 243–250.
- Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27.
- Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 77–86.
- Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 541–550.
- Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 120–127.
- Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer-Verlag Berlin Heidelberg.
- Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 437–444.
- Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013)*.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL)*, pages 1003–1011.
- Ramesh Nallapati. 2004. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 64–71.
- Truc Vien T Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 277–282.
- 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.
- Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 148–163.
- Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New york university 2011 system for kbp slot filling. In *Text Analysis Conference 2011 Workshop*.
- Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 455–465.
- Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 721–729.
- Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. 2011. Relation extraction with relation topics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1426–1436.
- Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson, editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 4–11. ACM.

Wei Xu, Ralph Grishman, and Le Zhao. 2011. Passage retrieval for information extraction using distant supervision. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1046–1054.

Roman Yangarber. 2001. *Scenario customization for information extraction*. Ph.D. thesis, Department of Computer Science, Graduate School of Arts and Science, New York University.

# Joint Apposition Extraction with Syntactic and Semantic Constraints

Will Radford and James R. Curran  
a-lab, School of Information Technologies  
University of Sydney  
NSW, 2006, Australia  
{wradford, james}@it.usyd.edu.au

## Abstract

Appositions are adjacent NPs used to add information to a discourse. We propose systems exploiting syntactic and semantic constraints to extract appositions from OntoNotes. Our joint log-linear model outperforms the state-of-the-art Favre and Hakkani-Tür (2009) model by  $\sim 10\%$  on Broadcast News, and achieves 54.3% F-score on multiple genres.

## 1 Introduction

Appositions are typically adjacent coreferent noun phrases (NP) that often add information about named entities (NEs). The apposition in Figure 1 consists of three comma-separated NPs – the first NP (HEAD) names an entity and the others (ATTRS) supply age and profession attributes. Attributes can be difficult to identify despite characteristic punctuation cues, as punctuation plays many roles and attributes may have rich substructure.

While linguists have studied apposition in detail (Quirk et al., 1985; Meyer, 1992), most apposition extraction has been within other tasks, such as coreference resolution (Luo and Zitouni, 2005; Culotta et al., 2007) and textual entailment (Roth and Sammons, 2007). Extraction has rarely been intrinsically evaluated, with Favre and Hakkani-Tür’s work a notable exception.

We analyze apposition distribution in OntoNotes 4 (Pradhan et al., 2007) and compare rule-based, classification and parsing extraction systems. Our best system uses a joint model to classify pairs of NPs with features that faithfully encode syntactic and semantic restrictions on appositions, using parse trees and WordNet synsets.

$\{John\ Ake\}_h$ ,  $\{48\}_a$ ,  $\{a\ \text{former}\ \text{vice-president}\ \text{in}\ \text{charge}\ \text{of}\ \text{legal}\ \text{compliance}\ \text{at}\ \text{American}\ \text{Capital}\ \text{Management}\ \&\ \text{Research}\ \text{Inc.},\ \text{in}\ \text{Houston},\}_a, \dots$

Figure 1: Example apposition from OntoNotes 4

Our approach substantially outperforms Favre and Hakkani-Tür on Broadcast News (BN) at 54.9% F-score and has state-of-the-art performance 54.3% F-score across multiple genres. Our results will immediately help the many systems that already use apposition extraction components, such as coreference resolution and IE.

## 2 Background

Apposition is widely studied, but “grammarians vary in the freedom with which they apply the term ‘apposition’” (Quirk et al., 1985). They are usually composed of two or more adjacent NPs, hierarchically structured, so one is the *head* NP (HEAD) and the rest *attributes* (ATTRS). They are often flagged using punctuation in text and pauses in speech. Pragmatically, they allow an author to introduce new information and build a shared context (Meyer, 1992).

Quirk et al. propose three tests for apposition: i) each phrase can be omitted without affecting sentence acceptability, ii) each fulfils the same syntactic function in the resultant sentences, iii) extralinguistic reference is unchanged. Strict interpretations may exclude other information-bearing cases like *pseudo-titles* (e.g.  $\{\text{President}\}_a \{\text{Bush}\}_h$ ), but include some adverbial phrases (e.g.  $\{\text{John}\ \text{Smith}\}_h$ ,  $\{\text{formerly}\ \text{(the}\ \text{president})_{AP}\}_a$ ). We adopt the OntoNotes guidelines’ relatively strict interpretation: “a noun phrase that modifies an immediately-adjacent noun phrase (these may be separated by only a comma, colon, or parenthesis).” (BBN, 2004–2007).

| Unit   | TRAIN <sub>F</sub> | DEV <sub>F</sub> | TEST <sub>F</sub> | TRAIN  | DEV   | TEST  |
|--------|--------------------|------------------|-------------------|--------|-------|-------|
| Sents. | 9,595              | 976              | 1,098             | 48,762 | 6,894 | 6,896 |
| Appos. | 590                | 64               | 68                | 3,877  | 502   | 490   |

Table 1: Sentence and apposition distribution

Apposition extraction is a common component in many NLP tasks: coreference resolution (Luo and Zitouni, 2005; Culotta et al., 2007; Bengtson and Roth, 2008; Poon and Domingos, 2008), textual entailment (Roth and Sammons, 2007; Cabrio and Magnini, 2010), sentence simplification (Miwa et al., 2010; Candido et al., 2009; Siddharthan, 2002) and summarization (Nenkova et al., 2005). Comma ambiguity has been studied in the RTE (Srikumar et al., 2008) and generation domains (White and Rajkumar, 2008).

Despite this, few papers to our knowledge explicitly evaluate apposition extraction. Moreover, apposition extraction is rarely the main research goal and descriptions of the methods used are often accordingly terse or do not match our guidelines. Lee et al. (2011) use rules to extract appositions for coreference resolution, selecting only those that are explicitly flagged using commas or parentheses. They do not separately mark HEAD and ATTR and permit relative clauses as an ATTR. While such differences capture useful information for coreference resolution, these methods would be unfairly disadvantaged in a direct evaluation.

Favre and Hakkani-Tür (2009, FHT) directly evaluate three extraction systems on OntoNotes 2.9 news broadcasts. The first retrains the Berkeley parser (Petrov and Klein, 2007) on trees labelled with appositions by appending the HEAD and ATTR suffix to NPs – we refer to this as a Labelled Berkeley Parser (LBP). The second is a CRF labelling words using an IOB apposition scheme. Token, POS, NE and BP-label features are used, as are presence of speech pauses. The final system classifies parse tree phrases using an Adaboost classifier (Schapire and Singer, 2000) with similar features.

The LBP, IOB and phrase systems score 41.38%, 32.76% and 40.41%, while their best uses LBP tree labels as IOB features, scoring 42.31%. Their focus on BN automated speech recognition (ASR) output, which precludes punctuation cues, does not indicate how well the methods perform on textual genres. Moreover all systems use parsers or parse-label features and do not completely evaluate non-parser methods for extraction despite including baselines.

| Form   | #    | %    | Reverse form | #   | %    | Σ%   |
|--------|------|------|--------------|-----|------|------|
| H t A  | 2109 | 55.9 | A t H        | 724 | 19.2 | 75.1 |
| A H    | 482  | 12.8 | H A          | 205 | 5.4  | 93.3 |
| H , A  | 1843 | 48.9 | A , H        | 532 | 14.1 | 63.0 |
| A H    | 482  | 12.9 | H A          | 205 | 5.4  | 81.3 |
| H ( A  | 146  | 3.9  | A ( H        | 16  | 0.4  | 85.6 |
| A : H  | 94   | 2.5  | H : A        | 23  | 0.6  | 88.7 |
| H -- A | 66   | 1.8  | A -- H       | 35  | 0.9  | 91.4 |
| A - H  | 31   | 0.8  | H - A        | 21  | 0.6  | 92.8 |

Table 2: Apposition forms in TRAIN with abstract (top) and actual (bottom) tokens, e.g., H t A indicates an HEAD, one token then an ATTR.

### 3 Data

We use apposition-annotated documents from the English section of OntoNotes 4 (Weischedel et al., 2011). We manually adjust appositions that do not have exactly one HEAD and one or more ATTR<sup>1</sup>. Some appositions are nested, and we keep only “leaf” appositions, removing the higher-level appositions.

We follow the CoNLL-2011 scheme to select TRAIN, DEV and TEST datasets (Pradhan et al., 2011). OntoNotes 4 is made up of a wide variety of sources: broadcast conversation and news, magazine, newswire and web text. Appositions are most frequent in newswire (one per 192 words) and least common in broadcast conversation (one per 645 words) with the others in between (around one per 315 words).

We also replicate the OntoNotes 2.9 BN data used by FHT, selecting the same sentences from OntoNotes 4 (TRAIN<sub>F</sub>/DEV<sub>F</sub>/TEST<sub>F</sub>). We do not “speechify” our data and take a different approach to nested apposition. Table 1 shows the distribution of sentences and appositions (HEAD-ATTR pairs).

#### 3.1 Analysis

Most appositions in TRAIN have one ATTR (97.4%) with few having two (2.5%) or three (0.1%). HEADS are typically shorter (median 5 tokens, 95% < 7) than ATTRS (median 7 tokens, 95% < 15). Table 2 shows frequent apposition forms. Comma-separated apposition is the most common (63%) and 93% are separated by zero or one token. HEADS are often composed of NES: 52% PER and 13% ORG, indicating an entity about which the ATTR adds information.

<sup>1</sup>Available at <http://schwa.org/resources>



| Pattern and Example  | P    | R    | F    |
|--|------|------|------|
| $\{\text{ne:PER}\}_h \# \{\text{pos:NP} (\text{pos:IN ne:LOC ORG GPE})?\}_a \#$<br>“{Jian Zhang} <sub>h</sub> , {the head of Chinese delegation} <sub>a</sub> ,” | 73.1 | 21.9 | 33.7 |
| $\{\text{pos:DT gaz:role relation}\}_a \#? \{\text{ne:PER}\}_h$<br>“{his new wife} <sub>a</sub> {Camilla} <sub>h</sub> ”   | 45.9 | 9.5  | 15.8 |
| $\{\text{ne:ORG GPE}\}_h \# \{\text{pos:DT pos:NP}\}_a \#$<br>“{Capetronic Inc.} <sub>h</sub> , {a Taiwan electronics maker} <sub>a</sub> ,”                     | 60.4 | 6.0  | 10.9 |
| $\{\text{pos:NP}\}_a \# \{\text{ne:PER}\}_h \#$<br>“{The vicar} <sub>a</sub> , {W.D. Jones} <sub>h</sub> ,”  | 33.7 | 4.5  | 7.9  |
| $\{\text{ne:PER}\}_h \# \{\text{pos:NP pos:POS pos:NP}\}_a \#$<br>“{Laurence Tribe} <sub>h</sub> , {Gore ’s attorney} <sub>a</sub> ,”                            | 82.0 | 4.0  | 7.7  |

Table 3: The top-five patterns by recall in the TRAIN dataset. ‘#’ is a pause (e.g., punctuation), ‘|’ a disjunction and ‘?’ an optional part. Patterns are used to combine tokens into NPs for pos:NP.

## 4 Extracting Appositions

We investigate different extraction systems using a range of syntactic information. Our systems that use syntactic parses generate candidates (pairs of NPs:  $p_1$  and  $p_2$ ) that are then classified as apposition or not.

This paper contributes three complementary techniques for more faithfully modelling apposition. Any adjacent NPs, disregarding intervening punctuation, could be considered candidates, however stronger syntactic constraints that only allow sibling NP children provide higher precision candidate sets. Semantic compatibility features encoding that an ATTR provides consistent information for its HEAD. A joint classifier models the complete apposition rather than combining separate phrase-wise decisions. Taggers and parsers are trained on TRAIN and evaluated on DEV or TEST. We use the C&C tools (Curran and Clark, 2003) for POS and NE tagging and the and the Berkeley Parser (Petrov and Klein, 2007), trained with default parameters.

**Pattern** POS, NE and lexical patterns are used to extract appositions avoiding parsing’s computational overhead. Rules are applied independently to tokenized and tagged sentences, yielding HEAD-ATTR tuples that are later deduplicated. The rules were manually derived from TRAIN<sup>2</sup> and Table 3 shows the top five of sixteen rules by recall over TRAIN. The “role” gazetteer is the transitive closure of hyponyms of the WordNet (Miller, 1995) synset `person.n.01` and “relation” manually constructed (e.g., “father”, “colleague”). Tuples are post-processed to remove spurious appo-

<sup>2</sup>There is some overlap between TRAIN and DEV<sub>F</sub>/TEST<sub>F</sub> with appositions from the latter used in rule generation.

sitions such as comma-separated NE lists<sup>3</sup>.

**Adjacent NPs** This low precision, high recall baseline assumes all candidates, depending on generation strategy, are appositions.

**Rule** We only consider HEADS whose syntactic head is a PER, ORG, LOC or GPE NE. We formalise *semantic compatibility* by requiring the ATTR head to match a gazetteer dependent on the HEAD’s NE type. To create PER, ORG and LOC gazetteers, we identified common ATTR heads in TRAIN and looked for matching WordNet synsets, selecting the most general hypernym that was still semantically compatible with the HEAD’s NE type.

Gazetteer words are pluralized using `pattern.en` (De Smedt and Daelemans, 2012) and normalised. We use partitive and NML-aware rules (Collins, 1999; Vadas and Curran, 2007) to extract syntactic heads from ATTRs. These must match the type-appropriate gazetteer, with ORG and LOC/GPE falling back to PER (e.g., “the champion, Apple”).

Extracted tuples are post-processed as for Pattern and reranked by the OntoNotes specificity scale (i.e., NNP > PRO > Def. NP > Indef. NP > NP), and the more specific unit is assigned HEAD. Possible ATTRs further to the left or right are checked, allowing for cases such as Figure 1.

**Labelled Berkeley Parser** We train a LBP on TRAIN and recover appositions from parsed sentences. Without syntactic constraints this is equivalent to FHT’s LBP system (LBP<sub>F</sub>) and indicated by † in Tables.

**Phrase** Each NP is independently classified as HEAD, ATTR or None. We use a log-linear model with a SGD optimizer from scikit-learn (Pedregosa

<sup>3</sup>Full description: <http://schwa.org/resources>

| Model     | Full system |      |             | -syn |      |       | -sem |      |      | -both |      |       | +gold |      |      |
|-----------|-------------|------|-------------|------|------|-------|------|------|------|-------|------|-------|-------|------|------|
| Pattern   | 44.8        | 34.9 | 39.2        | -    | -    | -     | -    | -    | -    | -     | -    | 52.2  | 39.6  | 45.1 |      |
| Adj NPs   | 11.6        | 58.0 | 19.3        | 3.6  | 65.1 | 6.8   | -    | -    | -    | -     | -    | 16.0  | 85.3  | 27.0 |      |
| Rule      | 65.3        | 46.8 | 54.5        | 43.7 | 50.0 | 46.7  | -    | -    | -    | -     | -    | 79.1  | 62.0  | 69.5 |      |
| LBP       | 66.3        | 52.2 | 58.4        | 47.8 | 53.0 | †50.3 | -    | -    | -    | -     | -    | -     | -     | -    |      |
| Phrase    | 73.2        | 45.6 | 56.2        | 77.7 | 41.0 | 53.7  | 73.2 | 44.6 | 55.4 | 77.7  | 40.8 | ‡53.5 | 89.0  | 58.2 | 70.4 |
| Joint     | 66.3        | 49.0 | 56.4        | 68.5 | 48.6 | 56.9  | 70.4 | 47.0 | 56.4 | 68.9  | 48.0 | 56.6  | 87.9  | 69.5 | 77.6 |
| Joint LBP | 69.6        | 51.0 | <b>58.9</b> | 69.6 | 49.6 | 57.9  | 71.5 | 49.0 | 58.2 | 68.3  | 48.6 | 56.8  | -     | -    | -    |

Table 4: Results over DEV: each column shows precision, recall and F-score. -syn/-sem/-both show the impact of removing constraints/features, +gold shows the impact of parse and tagging errors.

et al., 2011). The binary features are calculated from a generated candidate phrase ( $p$ ) and are the same as FHT’s phrase system ( $\text{Phrase}_F$ ), denoted ‡ in Tables. In addition, we propose the features below and to decode classifications, adjacent apposition-classified NPs are re-ordered by specificity.

- $p$  precedes/follows punctuation/interjection
- $p$  starts with a DT or PRP\$ (e.g., “{the director}<sub>a</sub>” or “{her husband}<sub>a</sub>”)
- $p$ ’s syntactic head matches a NE-specific *semantic* gazetteer (e.g., “{the famous actor}<sub>a</sub>”  $\rightarrow$  PER, “{investment bank}<sub>a</sub>”  $\rightarrow$  ORG)
- $p$ ’s syntactic head has the POS CD (e.g., “{John Smith}<sub>h</sub>, {34}<sub>a</sub>, ...”)
- $p$ ’s NE type (e.g., “{John Smith}<sub>h</sub>”  $\rightarrow$  PER)
- Specificity rank

**Joint** The final system classifies *pairs* of phrases ( $p_1, p_2$ ) as: HEAD-ATTR, ATTR-HEAD or None. The system uses the phrase model features as above as well as pairwise features:

- the cross-product of selected features for  $p_1$  and  $p_2$ : gazetteer matches, NE type, specificity rank. This models the compatibility between  $p_1$  and  $p_2$ . For example, if the HEAD has the NE type PER and the ATTR has the syntactic head in the PER gazetteer, for example “{Tom Cruise}<sub>h</sub>, {famous actor}<sub>a</sub>,”  $\rightarrow$  ( $p_1$ : PER,  $p_2$ : PER-gaz)
- If semantic features are found in  $p_1$  **and**  $p_2$
- $p_1/p_2$  specificity (e.g., equal,  $p_1 > p_2$ )
- whether  $p_1$  is an acronym of  $p_2$  or vice-versa

## 5 Results

We evaluate by comparing the extracted HEAD-ATTR pairs against the gold-standard. Correct pairs match gold-standard bounds and label. We report precision (P), recall (R) and  $F_1$ -score (F).

Table 4 shows our systems’ performance on the multi-genre DEV dataset, the impact of removing syntactic constraints, semantic features and

parse/tag error. Pattern performance is reasonable at 39.2% F-score given its lack of full syntactic information. All other results use parses and, although it has a low F-score, the Adjacent NPs’ 65.1% recall, without syntactic constraints, is the upper bound for the parse-based systems. Statistical models improve performance, with the joint models better than the higher-precision phrase model as the latter must make two independently correct classification decisions. Our best system has an F-score of 58.9% using a joint model over the de-labelled trees produced by the LBP. This indicates that although our model does not use the apposition labels from the tree, the tree is a more suitable structure for extraction. This system substantially improves on our implementation of FHT’s LBP<sub>F</sub> (†) and  $\text{Phrase}_F$  (‡) systems by 8.6% and 5.4%<sup>4</sup>.

Removing syntactic constraints mostly reduces performance in parse-based systems as the system must consider lower-quality candidates. The F-score increase is driven by higher precision at minimal cost to recall. Removing semantic features has less impact and removing both is most detrimental to performance. These features have less impact on joint models; indeed, joint performance using BP trees increases without the features, perhaps as joint models already model the syntactic context.

We evaluate the impact of parser and tagger error by using gold-standard resources. Gold-standard tags and trees improve recall in all cases leading to F-score improvements (+gold). The pattern system is reasonably robust to automatic tagging errors, but parse-based models suffer considerably from automatic parses. To compare the impact of tagging and parsing error, we configure the joint system to use gold parses and automatic NE tags and vice versa. Using automatic tags does not greatly impact performance (-1.3%), whereas

<sup>4</sup>We do not implement the IOB or use LBP features for TRAIN as these would require n-fold parser training.

| Model                 | P    | R    | F           |
|-----------------------|------|------|-------------|
| LBP <sub>F</sub> †    | 53.1 | 46.9 | 49.8        |
| Phrase <sub>F</sub> ‡ | 71.5 | 30.2 | 42.5        |
| Pattern               | 44.8 | 34.3 | 38.8        |
| LBP                   | 63.9 | 45.1 | 52.9        |
| Joint LBP             | 66.9 | 45.7 | <b>54.3</b> |

Table 5: Results over TEST: FHT’s (top) and our (bottom) systems.

| Error                 | BP     | LBP    | $\delta$ |
|-----------------------|--------|--------|----------|
| PP Attachment         | 5,585  | 5,396  | -189     |
| NP Internal Structure | 1,483  | 1,338  | -145     |
| Other                 | 3,164  | 3,064  | -100     |
| Clause Attachment     | 3,960  | 3,867  | -93      |
| Modifier Attachment   | 1,523  | 1,700  | 177      |
| Co-ordination         | 3,095  | 3,245  | 150      |
| NP Attachment         | 2,615  | 2,680  | 65       |
| Total                 | 30,189 | 29,859 | -330     |

Table 6: Selected BP/LBP parse error distribution.

using automatic parses causes a drop of around 20% to 57.7%, demonstrating that syntactic information is crucial for apposition extraction.

We compare our work with Favre and Hakkani-Tür (2009), training with TRAIN<sub>F</sub> and evaluating over TEST<sub>F</sub>—exclusively BN data. Our implementations of their systems, Phrase<sub>F</sub> and LBP<sub>F</sub>, perform at 43.6% and 44.1%. Our joint LBP system is substantially better, scoring 54.9%.

Table 5 shows the performance of our best systems on the TEST dataset and these follow the same trend as DEV. Joint LBP performs the best at 54.3%, 4.5% above LBP<sub>F</sub>.

Finally, we test whether labelling appositions can help parsing. We parse DEV trees with LBP and BP, remove apposition labels and analyse the impact of labelling using the Berkeley Parser Analyser (Kummerfeld et al., 2012). Table 6 shows the LBP makes fewer errors, particularly NP internal structuring, PP and clause attachment classes at the cost of modifier attachment and co-ordination errors. Rather than increasing parsing difficulty, apposition labels seem complementary, improving performance.

## 6 Conclusion

We present three apposition extraction techniques. Linguistic tests for apposition motivate strict syntactic constraints on candidates and semantic features encode the addition of compatible informa-

tion. Joint models more faithfully capture apposition structure and our best system achieves state-of-the-art performance of 54.3%. Our results will immediately benefit the large number of systems with apposition extraction components for coreference resolution and IE.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their suggestions. Thanks must also go to Benoit Favre for his clear writing and help answering our questions as we replicated his dataset and system. This work has been supported by ARC Discovery grant DP1097291 and the Capital Markets CRC Computable News project.

## References

- BBN. 2004–2007. Co-reference guidelines for english ontonotes. Technical Report v6.0, BBN Technologies.
- Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, Honolulu, Hawaii.
- Elena Cabrio and Bernardo Magnini. 2010. Toward qualitative evaluation of textual entailment systems. In *Coling 2010: Posters*, pages 99–107. Coling 2010 Organizing Committee, Beijing, China.
- Arnaldo Candido, Erick Maziero, Lucia Specia, Caroline Gasperin, Thiago Pardo, and Sandra Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics, Boulder, Colorado.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 81–88. Association

- for Computational Linguistics, Rochester, New York.
- James Curran and Stephen Clark. 2003. Language independent ner using a maximum entropy tagger. In Walter Daelemans and Miles Osborne, editors, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 164–167.
- Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13:2013–2035.
- Benoit Favre and Dilek Hakkani-Tür. 2009. Phrase and word level strategies for detecting appositions in speech. In *Proceedings of Interspeech 2009*, pages 2711–2714. Brighton, UK.
- Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059. Jeju Island, South Korea.
- Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford’s multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the CoNLL-2011 Shared Task*. URL [pubs/conll1st2011-coref.pdf](http://pubs/conll1st2011-coref.pdf).
- Xiaoqiang Luo and Imed Zitouni. 2005. Multilingual coreference resolution with syntactic features. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 660–667. Association for Computational Linguistics, Vancouver, British Columbia, Canada.
- Charles F. Meyer. 1992. *Apposition in Contemporary English*. Cambridge University Press, Cambridge, UK.
- George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.
- Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun’ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796. Coling 2010 Organizing Committee, Beijing, China.
- Ani Nenkova, Advaith Siddharthan, and Kathleen McKeown. 2005. Automatically learning cognitive status for multi-document summarization of newswire. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 241–248. Association for Computational Linguistics, Vancouver, British Columbia, Canada.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Slav Petrov and Dan Klein. 2007. Learning and inference for hierarchically split PCFGs. In *Proceedings of the 22nd AAAI Conference of Artificial Intelligence*, pages 1642–1645. Vancouver, Canada.
- Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659. Association for Computational Linguistics, Honolulu, Hawaii.
- 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. Portland, OR USA.
- Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A unified relational semantic representation. In *Proceedings of the International Conference on Semantic Computing*, pages 517–526. Washington, DC USA.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. General Grammar Series. Longman, London, UK.

- Dan Roth and Mark Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112. Association for Computational Linguistics, Prague.
- Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168.
- Advaith Siddharthan. 2002. Resolving attachment and clause boundary ambiguities for simplifying relative clause constructs. In *Proceedings of the ACL Student Research Workshop (ACLSRW 2002)*, pages 60–65. Association for Computational Linguistics, Philadelphia.
- Vivek Srikumar, Roi Reichart, Mark Sammons, Ari Rappoport, and Dan Roth. 2008. Extraction of entailed semantic relations through syntax-based comma resolution. In *Proceedings of ACL-08: HLT*, pages 1030–1038. Columbus, OH USA.
- David Vadas and James R. Curran. 2007. Parsing internal noun phrase structure with collins’ models. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 109–116. Melbourne, Australia.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2011. OntoNotes Release 4.0. Technical report, Linguistic Data Consortium, Philadelphia, PA USA.
- Michael White and Rajakrishnan Rajkumar. 2008. A more precise analysis of punctuation for broad-coverage surface realization with CCG. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 17–24. Coling 2008 Organizing Committee, Manchester, England.

# Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation

**Kevin Duh, Graham Neubig**  
Graduate School of Information Science  
Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma, Japan  
kevinduh@is.naist.jp  
neubig@is.naist.jp

**Katsuhito Sudoh, Hajime Tsukada**  
NTT Communication Science Labs.  
NTT Corporation  
2-4 Hikaridai, Seika, Kyoto, Japan  
sudoh.katsuhito@lab.ntt.co.jp  
tsukada.hajime@lab.ntt.co.jp

## Abstract

Data selection is an effective approach to domain adaptation in statistical machine translation. The idea is to use language models trained on small in-domain text to select similar sentences from large general-domain corpora, which are then incorporated into the training data. Substantial gains have been demonstrated in previous works, which employ standard n-gram language models. Here, we explore the use of neural language models for data selection. We hypothesize that the continuous vector representation of words in neural language models makes them more effective than n-grams for modeling unknown word contexts, which are prevalent in general-domain text. In a comprehensive evaluation of 4 language pairs (English to German, French, Russian, Spanish), we found that neural language models are indeed viable tools for data selection: while the improvements are varied (i.e. 0.1 to 1.7 gains in BLEU), they are fast to train on small in-domain data and can sometimes substantially outperform conventional n-grams.

## 1 Introduction

A perennial challenge in building Statistical Machine Translation (SMT) systems is the dearth of high-quality bitext in the domain of interest. An effective and practical solution is *adaptation data selection*: the idea is to use language models (LMs) trained on in-domain text to select similar sentences from large general-domain corpora. The selected sentences are then incorporated into the SMT training data. Analyses have shown that this augmented data can lead to better statistical estimation or word coverage (Duh et al., 2010; Haddow and Koehn, 2012).

Although previous works in data selection (Axelrod et al., 2011; Koehn and Haddow, 2012; Yasuda et al., 2008) have shown substantial gains, we suspect that the commonly-used *n-gram* LMs may be sub-optimal. The small size of the in-domain text implies that a large percentage of general-domain sentences will contain words not observed in the LM training data. In fact, as many as 60% of general-domain sentences contain at least one unknown word in our experiments. Although the LM probabilities of these sentences could still be computed by resorting to back-off and other smoothing techniques, a natural question remains: will alternative, more robust LMs do better?

We hypothesize that the *neural language model* (Bengio et al., 2003) is a viable alternative, since its continuous vector representation of words is well-suited for modeling sentences with frequent unknown words, providing smooth probability estimates of unseen but similar contexts. Neural LMs have achieved positive results in speech recognition and SMT reranking (Schwenk et al., 2012; Mikolov et al., 2011a). To the best of our knowledge, this paper is the first work that examines neural LMs for adaptation data selection.

## 2 Data Selection Method

We employ the data selection method of (Axelrod et al., 2011), which builds upon (Moore and Lewis, 2010). The intuition is to select general-domain sentences that are similar to in-domain text, while being dis-similar to the average general-domain text.

To do so, one defines the score of an general-domain sentence pair  $(e, f)$  as:

$$[IN_E(e) - GEN_E(e)] + [IN_F(f) - GEN_F(f)] \quad (1)$$

where  $IN_E(e)$  is the *length-normalized* cross-entropy of  $e$  on the English in-domain LM.  $GEN_E(e)$  is the length-normalized cross-entropy

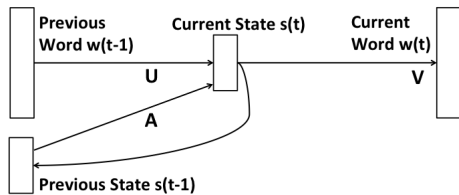


Figure 1: Recurrent neural LM.

of  $e$  on the English general-domain LM, which is built from a sub-sample of the general-domain text. Similarly,  $IN_F(f)$  and  $GEN_F(f)$  are the cross-entropies of  $f$  on Foreign-side LM. Finally, sentence pairs are ranked according to Eq. 1 and those with scores lower than some empirically-chosen threshold are added to the bitext for translation model training.

## 2.1 Neural Language Models

The four LMs used to compute Eq. 1 have conventionally been n-grams. N-grams of the form  $p(w(t)|w(t-1), w(t-2), \dots)$  predict words by using multinomial distributions conditioned on the context  $(w(t-1), w(t-2), \dots)$ . But when the context is rare or contains unknown words, n-grams are forced to back-off to lower-order models, e.g.  $p(w(t)|w(t-1))$ . These backoffs are unfortunately very frequent in adaptation data selection.

Neural LMs, in contrast, model word probabilities using continuous vector representations. Figure 1 shows a type of neural LMs called recurrent neural networks (Mikolov et al., 2011b).<sup>1</sup> Rather than representing context as an identity (n-gram hit-or-miss) function on  $[w(t-1), w(t-2), \dots]$ , neural LMs summarize the context by a hidden state vector  $s(t)$ . This is a continuous vector of dimension  $|S|$  whose elements are predicted by the previous word  $w(t-1)$  and previous state  $s(t-1)$ . This is robust to rare contexts because continuous representations enable *sharing* of statistical strength between similar contexts. Bengio (2009) shows that such representations are better than multinomials in alleviating sparsity issues.

<sup>1</sup>Another major type of neural LMs are the so-called feed-forward networks (Bengio et al., 2003; Schwenk, 2007; Nakamura et al., 1990). Both types of neural LMs have seen many improvements recently, in terms of computational scalability (Le et al., 2011) and modeling power (Arisoy et al., 2012; Wu et al., 2012; Alexandrescu and Kirchoff, 2006). We focus on recurrent networks here since there are fewer hyper-parameters and its ability to model infinite context using recursion is theoretically attractive. But we note that feed-forward networks are just as viable.

Now, given state vector  $s(t)$ , we can predict the probability of the current word. Figure 1 is expressed formally in the following equations:

$$w(t) = [w_0(t), \dots, w_k(t), \dots, w_{|W|}(t)] \quad (2)$$

$$w_k(t) = g \left( \sum_{j=0}^{|S|} s_j(t) V_{kj} \right) \quad (3)$$

$$s_j(t) = f \left( \sum_{i=0}^{|W|} w_i(t-1) U_{ji} + \sum_{i'=0}^{|S|} s_{i'}(t-1) A_{ji'} \right) \quad (4)$$

Here,  $w(t)$  is viewed as a vector of dimension  $|W|$  (vocabulary size) where each element  $w_k(t)$  represents the probability of the  $k$ -th vocabulary item at sentence position  $t$ . The function  $g(z_k) = e^{z_k} / \sum_k e^{z_k}$  is a softmax function that ensures the neural LM outputs are proper probabilities, and  $f(z) = 1/(1 + e^{-z})$  is a sigmoid activation that induces the non-linearity critical to the neural network's expressive power. The matrices  $V$ ,  $U$ , and  $A$  are trained by maximizing likelihood on training data using a "backpropagation-through-time" method.<sup>2</sup> Intuitively,  $U$  and  $A$  compress the context ( $|S| < |W|$ ) such that contexts predictive of the same word  $w(t)$  are close together.

Since proper modeling of unknown contexts is important in our problem, training text for both n-gram and neural LM is pre-processed by converting all low-frequency words in the training data (frequency=1 in our case) to a special "unknown" token. This is used only in Eq. 1 for selecting general-domain sentences; these words retain their surface forms in the SMT train pipeline.

## 3 Experiment Setup

We experimented with four language pairs in the WIT<sup>3</sup> corpus (Cettolo et al., 2012), with English (en) as source and German (de), Spanish (es), French (fr), Russian (ru) as target. This is the in-domain corpus, and consists of TED Talk transcripts covering topics in technology, entertainment, and design. As general-domain corpora, we collected bitext from the WMT2013 campaign, including CommonCrawl and NewsCommentary for all 4 languages, Europarl for de/es/fr, UN for es/fr, Gigaword for fr, and Yandex for ru. The in-domain data is divided into a training set (for SMT

<sup>2</sup>The recurrent states are unrolled for several time-steps, then stochastic gradient descent is applied.

|                        | en-de | en-es | en-fr | en-ru |
|------------------------|-------|-------|-------|-------|
| In-domain Training Set |       |       |       |       |
| #sentence              | 129k  | 140k  | 139k  | 117k  |
| #token (en)            | 2.5M  | 2.7M  | 2.7M  | 2.3M  |
| #vocab (en)            | 26k   | 27k   | 27k   | 25k   |
| #vocab (f)             | 42k   | 39k   | 34k   | 58k   |
| General-domain Bitext  |       |       |       |       |
| #sentence              | 4.4M  | 14.7M | 38.9M | 2.0M  |
| #token (en)            | 113M  | 385M  | 1012M | 51M   |
| %unknown               | 60%   | 58%   | 64%   | 65%   |

Table 1: Data statistics. ”%unknown”=fraction of general-domain sentences with unknown words.

pipeline and neural LM training), a tuning set (for MERT), a validation set (for choosing the optimal threshold in data selection), and finally a testset of 1616 sentences.<sup>3</sup> Table 1 lists data statistics.

For each language pair, we built a baseline **in-data** SMT system trained only on in-domain data, and an **alldata** system using combined in-domain and general-domain data.<sup>4</sup> We then built 3 systems from augmented data selected by different LMs:

- **ngram**: Data selection by 4-gram LMs with Kneser-Ney smoothing (Axelrod et al., 2011)
- **neuralnet**: Data selection by Recurrent neural LM, with the RNNLM Toolkit.<sup>5</sup>
- **combine**: Data selection by interpolated LM using n-gram & neuralnet (equal weight).

All systems are built using standard settings in the Moses toolkit (GIZA++ alignment, grow-diag-final-and, lexical reordering models, and SRILM). Note that standard n-grams are used as LMs for SMT; neural LMs are only used for data selection. Multiple SMT systems are trained by thresholding on {10k,50k,100k,500k,1M} general-domain sentence subsets, and we empirically determine the single system for testing based on results on a separate validation set (in practice, 500k was chosen for fr and 1M for es, de, ru.).

<sup>3</sup>The original data are provided by <http://wit3.fbk.eu> and <http://www.statmt.org/wmt13/>. Our domain adaptation scenario is similar to the IWSLT2012 campaign but we used our own random train/test splits, since we wanted to ensure the testset for all languages had identical source sentences for comparison purposes. For replicability, our software is available at <http://cl.naist.jp/~kevinduh/a/acl2013>.

<sup>4</sup>More advanced phrase table adaptation methods are possible. but our interest is in comparing data selection methods. The conclusions should transfer to advanced methods such as (Foster et al., 2010; Niehues and Waibel, 2012).

<sup>5</sup><http://www.fit.vutbr.cz/~imikolov/rnnlm/>

## 4 Results

### 4.1 LM Perplexity and Training Time

First, we measured perplexity to check the generalization ability of our neural LMs as language models. Recall that we train four LMs to compute each of the components of Eq. 1. In Table 2, we compared each of the four versions of **ngram**, **neuralnet**, and **combine** LMs on in-domain test sets or general-domain held-out sets. It re-affirms previous positive results (Mikolov et al., 2011a), with **neuralnet** outperforming **ngram** by 20-30% perplexity across all tasks. Also, **combine** slightly improves the perplexity of **neuralnet**.

| Task                        | ngram | neuralnet | combine   |
|-----------------------------|-------|-----------|-----------|
| In-Domain Test Set          |       |           |           |
| en-de de                    | 157   | 110 (29%) | 110 (29%) |
| en-de en                    | 102   | 81 (20%)  | 78 (24%)  |
| en-es es                    | 129   | 102 (20%) | 98 (24%)  |
| en-es en                    | 101   | 80 (21%)  | 77 (24%)  |
| en-fr fr                    | 90    | 67 (25%)  | 65 (27%)  |
| en-fr en                    | 102   | 80 (21%)  | 77 (24%)  |
| en-ru ru                    | 208   | 167 (19%) | 155 (26%) |
| en-ru en                    | 103   | 83 (19%)  | 79 (23%)  |
| General-Domain Held-out Set |       |           |           |
| en-de de                    | 234   | 174 (25%) | 161 (31%) |
| en-de en                    | 218   | 168 (23%) | 155 (29%) |
| en-es es                    | 62    | 43 (31%)  | 43 (31%)  |
| en-es en                    | 84    | 61 (27%)  | 59 (30%)  |
| en-fr fr                    | 64    | 43 (33%)  | 43 (33%)  |
| en-fr en                    | 95    | 67 (30%)  | 65 (32%)  |
| en-ru ru                    | 242   | 199 (18%) | 176 (27%) |
| en-ru en                    | 191   | 153 (20%) | 142 (26%) |

Table 2: Perplexity of various LMs. Number in parenthesis is percentage improvement vs. ngram.

Second, we show that the usual concern of neural LM training time is not so critical for the in-domain data sizes used domain adaptation. The complexity of training Figure 1 is dominated by computing Eq. 3 and scales as  $O(|W| \times |S|)$  in the number of tokens. Since  $|W|$  can be large, one practical trick is to cluster the vocabulary so that the output dimension is reduced. Table 3 shows the training times on a 3.3GHz XeonE5 CPU by varying these two main hyper-parameters ( $|S|$  and cluster size). Note that the setting  $|S| = 200$  and cluster size of 100 already gives good perplexity in reasonable training time. All neural LMs in this paper use this setting, without additional tuning.



| $ S $ | Cluster | Time   | Perplexity |
|-------|---------|--------|------------|
| 200   | 100     | 198m   | 110        |
| 100   | $ W $   | 12915m | 110        |
| 200   | 400     | 208m   | 113        |
| 100   | 100     | 52m    | 118        |
| 100   | 400     | 71m    | 120        |

Table 3: Training time (in minutes) for various neural LM architectures (Task: en-de de).

## 4.2 End-to-end SMT Evaluation

Table 4 shows translation results in terms of BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010), and TER (Snover et al., 2006). We observe that all three data selection methods essentially outperform **alldata** and **indata** for all language pairs, and **neuralnet** tend to be the best in all metrics. E.g., BLEU improvements over **ngram** are in the range of 0.4 for en-de, 0.5 for en-es, 0.1 for en-fr, and 1.7 for en-ru. Although not all improvements are large in absolute terms, many are statistically significant (95% confidence).

We therefore believe that neural LMs are generally worthwhile to try for data selection, as it rarely underperform n-grams. The open question is: what can explain the significant improvements in, for example Russian, Spanish, German, but the lack thereof in French? One conjecture is that neural LMs succeeded in lowering testset out-of-vocabulary (OOV) rate, but we found that OOV reduction is similar across all selection methods.

The improvements appear to be due to *better probability estimates* of the translation/reordering models. We performed a diagnostic by decoding the testset using LMs trained on the same testset, while varying the translation/reordering tables with those of **ngram** and **neuralnet**; this is a kind of pseudo forced-decoding that can inform us about which table has better coverage. We found that across all language pairs, BLEU differences of translations under this diagnostic become insignificant, implying that the raw probability value is the differentiating factor between **ngram** and **neuralnet**. Manual inspection of en-de revealed that many improvements come from lexical choice in morphological variants ("meinen Sohn" vs. "mein Sohn"), segmentation changes ("baking soda" → "Backpulver" vs. "baken Soda"), and handling of unaligned words at phrase boundaries.

Finally, we measured the intersection between the sentence set selected by **ngram** vs **neural-**

| Task         | System    | BLEU                    | RIBES                   | TER                     |
|--------------|-----------|-------------------------|-------------------------|-------------------------|
| <b>en-de</b> | indata    | 20.8                    | 80.1                    | 59.0                    |
|              | alldata   | 21.5                    | 80.1                    | 59.1                    |
|              | ngram     | 21.5                    | 80.3                    | 58.9                    |
|              | neuralnet | <b>21.9<sup>+</sup></b> | <b>80.5<sup>+</sup></b> | <b>58.4</b>             |
|              | combine   | 21.5                    | 80.2                    | 58.8                    |
| <b>en-es</b> | indata    | 30.4                    | 83.5                    | 48.7                    |
|              | alldata   | 31.2                    | 83.2                    | 49.9                    |
|              | ngram     | 32.0                    | 83.7                    | 48.4                    |
|              | neuralnet | <b>32.5<sup>+</sup></b> | 83.7                    | <b>48.3<sup>+</sup></b> |
|              | combine   | <b>32.5<sup>+</sup></b> | <b>83.8</b>             | <b>48.3<sup>+</sup></b> |
| <b>en-fr</b> | indata    | 31.4                    | 83.9                    | 51.2                    |
|              | alldata   | 31.5                    | 83.5                    | 51.4                    |
|              | ngram     | 32.7                    | 83.7                    | 50.4                    |
|              | neuralnet | <b>32.8</b>             | <b>84.2<sup>+</sup></b> | <b>50.3</b>             |
|              | combine   | 32.5                    | 84.0                    | 50.5                    |
| <b>en-ru</b> | indata    | 14.8                    | 72.5                    | 69.5                    |
|              | alldata   | 23.4                    | 75.0                    | 62.3                    |
|              | ngram     | 24.0                    | 75.7                    | 61.4                    |
|              | neuralnet | <b>25.7<sup>+</sup></b> | <b>76.1</b>             | <b>60.0<sup>+</sup></b> |
|              | combine   | 23.7                    | 75.9                    | 61.9 <sup>-</sup>       |

Table 4: End-to-end Translation Results. The best results are bold-faced. We also compare neural LMs to ngram using pairwise bootstrap (Koehn, 2004): "+" means statistically significant improvement and "-" means significant degradation.

**net**. They share 60-75% of the augmented training data. This high overlap means that **ngram** and **neuralnet** are actually *not* drastically different systems, and **neuralnet** with its slightly better selections represent an *incremental* improvement.<sup>6</sup>

## 5 Conclusions

We perform an evaluation of neural LMs for adaptation data selection, based on the hypothesis that their continuous vector representations are effective at comparing general-domain sentences, which contain frequent unknown words. Compared to conventional n-grams, we observed end-to-end translation improvements from 0.1 to 1.7 BLEU. Since neural LMs are fast to train in the small in-domain data setting and achieve equal or incrementally better results, we conclude that they are an worthwhile option to include in the arsenal of adaptation data selection techniques.

<sup>6</sup>This is corroborated by another analysis: taking the *union* of sentences found by **ngram** and **neuralnet** gives similar BLEU scores as **neuralnet**.

## Acknowledgments

We thank Amittai Axelrod for discussions about data selection implementation details, and an anonymous reviewer for suggesting the *union* idea for results analysis. K. D. would like to credit Spyros Matsoukas (personal communication, 2010) for the trick of using LM-based pseudo forced-decoding for error analysis.

## References

- Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ebru Arisoy, Tara N. Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. 2012. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28, Montréal, Canada, June. Association for Computational Linguistics.
- Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language models. *JMLR*.
- Yoshua Bengio. 2009. *Learning Deep Architectures for AI*, volume Foundations and Trends in Machine Learning. NOW Publishers.
- Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit<sup>3</sup>: Web inventory of transcribed and translated talks. In *Proceedings of the 16<sup>th</sup> Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.
- Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) - Technical Papers Track*.
- George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*.
- Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montréal, Canada, June. Association for Computational Linguistics.
- Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. 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, Cambridge, MA, October. Association for Computational Linguistics.
- Philipp Koehn and Barry Haddow. 2012. Towards effective use of training data in statistical machine translation. In *WMT*.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.
- Hai-Son Le, I. Oparin, A. Allauzen, J. Gauvain, and F. Yvon. 2011. Structured output layer neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5524–5527.
- Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. 2011a. Strategies for training large scale neural network language model. In *ASRU*.
- Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011b. Extensions of recurrent neural network language model. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
- Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July. Association for Computational Linguistics.
- Masami Nakamura, Katsuteru Maruyama, Takeshi Kawabata, and Kiyohiro Shikano. 1990. Neural network approach to word category prediction for english texts. In *Proceedings of the 13th conference on Computational linguistics - Volume 3, COLING '90*, pages 213–218, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jan Niehues and Alex Waibel. 2012. Detailed analysis of different strategies for phrase table adaptation in SMT. In *AMTA*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.
- Holger Schwenk, Anthony Rousseau, and Mohammed Attik. 2012. Large, pruned or continuous space language models on a gpu for statistical machine translation. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for*

*HLT*, pages 11–19, Montréal, Canada, June. Association for Computational Linguistics.

Holger Schwenk. 2007. Continuous space language models. *Comput. Speech Lang.*, 21(3):492–518, July.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.

Youzheng Wu, Xugang Lu, Hitoshi Yamamoto, Shigeki Matsuda, Chiori Hori, and Hideki Kashioka. 2012. Factored language model based on recurrent neural network. In *Proceedings of COLING 2012*, pages 2835–2850, Mumbai, India, December. The COLING 2012 Organizing Committee.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *ICJNLP*.

# Mapping Source to Target Strings without Alignment by Analogical Learning: A Case Study with Transliteration

Philippe Langlais

RALI / DIRO

Université de Montréal

Montréal, Canada, H3C 3J7

felipe@iro.umontreal.ca

## Abstract

Analogical learning over strings is a holistic model that has been investigated by a few authors as a means to map forms of a source language to forms of a target language. In this study, we revisit this learning paradigm and apply it to the transliteration task. We show that alone, it performs worse than a statistical phrase-based machine translation engine, but the combination of both approaches outperforms each one taken separately, demonstrating the usefulness of the information captured by a so-called formal analogy.

## 1 Introduction

A proportional analogy is a relationship between four objects, noted  $[x : y :: z : t]$ , which reads as “ $x$  is to  $y$  as  $z$  is to  $t$ ”. While some strategies have been proposed for handling semantic relationships (Turney and Littman, 2005; Duc et al., 2011), we focus in this study on formal proportional analogies (hereafter formal analogies or simply analogies), that is, proportional analogies involving relationships at the graphemic level, such as  $[atomkraftwerken : atomkriegen :: kraftwerks : kriegs]$  in German.

Analogical learning over strings has been investigated by several authors. Yvon (1997) addressed the task of grapheme-to-phoneme conversion, a problem which continues to be studied actively, see for instance (Bhargava and Kondrak, 2011). Stroppa and Yvon (2005) applied analogical learning to computing morphosyntactic features to be associated with a form (lemma, part-of-speech, and additional features such as number, gender, case, tense, mood, etc.). The performance of the analogical engine on the Dutch language was as good as or better than the one reported in (van den Bosch and Daelemans, 1993). Lepage

and Denoual (2005) pioneered the application of analogical learning to Machine Translation. Different variants of the system they proposed have been tested in a number of evaluation campaigns, see for instance (Lepage et al., 2009). Langlais and Patry (2007) investigated the more specific task of translating unknown words, a problem simultaneously studied in (Denoual, 2007).

Analogical learning has been applied to various other purposes, among which query expansion in information retrieval (Moreau et al., 2007), classification of nominal and binary data, and handwritten character recognition (Miclet et al., 2008). Formal analogy has also been used for solving Raven IQ tests (Correa et al., 2012).

In this study, we investigate the relevance of analogical learning for English proper name transliteration into Chinese. We compare it to the statistical phrase-based machine translation approach (Koehn et al., 2003) initially proposed for transliteration by Finch and Sumita (2010). We show that alone, analogical learning underperforms the phrase-based approach, but that a combination of both outperforms individual systems.

We describe in section 2 the principle of analogical learning. In section 3, we report on experiments we conducted in applying analogical learning on the NEWS 2009 English-to-Chinese transliteration task. Related works are discussed in section 4. We conclude in section 5 and identify avenues we believe deserve investigations.

## 2 Analogical Learning

### 2.1 Formal Analogy

In this study, we use the most general definition of formal analogy we found, initially described in (Yvon et al., 2004). It handles a large variety of relations, including but not limited to affixation operations (i.e.  $[capital : anticapitalisme :: commun : anticommuniste]$  in French), stem mu-

tations (i.e. [*lang : länge :: stark : stärke*] in German), and even templatic relations (i.e. [*KaaTiB : KuTaaB :: QaaRi' : QuRaa'*] in Arabic).

Informally,<sup>1</sup> this definition states that 4 forms  $x, y, z$  and  $t$  are in analogical relation iff we can find a  $d$ -factorization (a factorization into  $d$  factors) of each form, such that the  $i^{\text{th}}$  factors ( $i \in [1, d]$ ) of  $x$  and  $z$  equal (in ensemble terms) the  $i^{\text{th}}$  factors of  $y$  and  $t$ .

For instance, [*this guy drinks : this boat sinks :: these guys drank : these boats sank*] holds because of the following 4-uple of 5-factorizations, whose factors are aligned column-wise for clarity, and where spaces (underlined>) are treated as regular characters ( $\epsilon$  designates the empty factor):

$$\begin{aligned} f_x &\equiv ( \text{this} \quad \underline{\text{guy}} \quad \epsilon \quad \underline{\text{dr}} \quad \text{inks} ) \\ f_y &\equiv ( \text{this} \quad \underline{\text{boat}} \quad \epsilon \quad \text{s} \quad \text{inks} ) \\ f_z &\equiv ( \text{these} \quad \underline{\text{guy}} \quad \text{s} \quad \underline{\text{dr}} \quad \text{ank} ) \\ f_t &\equiv ( \text{these} \quad \underline{\text{boat}} \quad \text{s} \quad \text{s} \quad \text{ank} ) \end{aligned}$$

This analogy “captures” among other things that in English, changing *this* for *these* implies a plural mark (s) to the corresponding noun. Note that analogies can relate arbitrarily distant substrings. For instance the 3rd-person singular mark of the verbs relates to the first substring *this*.

## 2.2 Analogical Learning

We now clarify the process of analogical learning. Let  $\mathcal{L} = \{(i(x_k), o(x_k))_k\}$  be a training set (or memory) gathering pairs of input  $i(x_k)$  and output  $o(x_k)$  representations of elements  $x_k$ . In this study, the elements we consider are pairs of English / Chinese proper names in a transliteration relation. Given an element  $t$  for which we only know  $i(t)$ , analogical learning works by:

1. building  $\mathcal{E}_i(t) = \{(x, y, z) \in \mathcal{L}^3 \mid [i(x) : i(y) :: i(z) : i(t)]\}$ , the set of triples in the training set that stand in analogical proportion with  $t$  in the input space,
2. building  $\mathcal{E}_o(t) = \{[o(x) : o(y) :: o(z) : ?] \mid (x, y, z) \in \mathcal{E}_i(t)\}$ , the set of solutions to the output analogical equations obtained,
3. selecting  $o(t)$  among the solutions aggregated into  $\mathcal{E}_o(t)$ .

In this description, we define an *analogical equation* as an analogy with one form missing, and

<sup>1</sup>We refer the reader to (Stroppa and Yvon, 2005) for a more technical exposition.

we note  $[x : y :: z : ?]$  the set of its solutions (i.e.  $\text{undoable} \in [\text{reader} : \text{doer} :: \text{unreadable} : ?]$ ).<sup>2</sup>



Figure 1: Excerpt of a transliteration session for the English proper name *Zemansky*. 31 solutions have been identified in total (4 by the first equation reported); the one underlined (actually the most frequently generated) is the sanctioned one.

Figure 1 illustrates this process on a transliteration session for the English proper name *Zemansky*. The training corpus  $\mathcal{L}$  is a set of pairs of English proper names and their Chinese Transliteration(s). Step 1 identifies analogies among English proper names: 7 such analogies are identified, 3 of which are reported (marked with a ▷ sign). Step 2 projects the English forms in analogical proportion into their known transliteration (illustrated by a ↓ sign) in order to solve Chinese analogical equations. Step 3 aggregates the solutions produced during the second step. In the example, it consists in sorting the solutions in decreasing order of the number of time they have been generated during step 2 (see next section for a better strategy).

There are several important points to consider when deploying the learning procedure shown above. First, the search stage (step 1) has a time complexity that can be prohibitive in some applications of interest. We refer the reader to (Langlais and Yvon, 2008) for a practical solution to this. Second, we need a way to solve an analogical

<sup>2</sup>Analogical equation solvers typically produce several solutions to an equation.

equation. We applied the finite-state machine procedure described in (Yvon et al., 2004). Suffice it to say that typically, this solver produces several solutions to an equation, most of them spurious,<sup>3</sup> reinforcing the need for an efficient aggregation step (step 3). Last, it might happen that the overall approach fails at producing a solution, because no input analogy is identified during step 1, or because the input analogies identified do not lead to analogies in the output space. This *silence* issue is analyzed in section 3. A detailed account of those problems and possible solutions are discussed in (Somers et al., 2009).

We underline that analogies in both source and target languages are considered *independently*: the approach does not attempt to align source and target substrings, but relies instead on the inductive bias that input analogies (often) imply output ones.

### 3 Experiments

#### 3.1 Setting

The task we study is part of the NEWS evaluation campaign conducted in 2009 (Li et al., 2009). The dataset consists of 31 961 English-Chinese transliteration examples for training the system (TRAIN), 2 896 ones for tuning it (DEV), and 2 896 for testing them (TEST).

We compare two different approaches to transliteration: a statistical phrase-based machine translation engine — which according to Li et al. (2009) was popular among participating systems to NEWS — as well as differently flavored analogical systems.

We trained (on TRAIN) a phrase-based translation device with the Moses toolkit (Koehn et al., 2007), very similarly to (Finch and Sumita, 2010), that is, considering each character as a word. The coefficients of the log-linear function optimized by Moses' decoder were tuned (with MERT) on DEV.

For the analogical system, we investigated the use of classifiers trained in a supervised way to recognize the good solutions generated during step 2. For this, we first transliterated the DEV dataset using TRAIN as a memory. Then, we trained a classifier, taking advantage of the DEV corpus for the supervision. We tried two types of learners — support vector machines (Cortes and Vapnik, 1995) and voted perceptrons (Freund

<sup>3</sup>A spurious solution is a string that does not belong to the language under consideration. See Figure 1 for examples.

and Schapire, 1999)<sup>4</sup> — and found the former to slightly outperform the latter. Finally, we transliterated the TEST corpus using both the TRAIN and DEV corpora as a memory,<sup>5</sup> and applied our classifiers on the solutions generated.

The lack of space prevents us to describe the 61 features we used for characterizing a solution. We initially considered a set of features which characterizes a solution (frequency, rank in the candidate list, language model likelihood, etc.), and the process that generated the solution (i.e. number of analogies involved), but no feature that would use scored pairs of substrings (such as mutual information of substrings).<sup>6</sup> Thus, we also considered in a second stage a set of features that we collected thanks to a  $n$ -best list of solutions computed by Moses (Moses' score given to a solution, its rank in the  $n$ -best list, etc.).

#### 3.2 Results

We ran the NEWS 2009 official evaluation script<sup>7</sup> in order to compute ACC (the accuracy of the first solution),  $F_1$  (the F-measure which gives partial credits proportional to the longest subsequence between the reference transliteration and the first candidate), and the Mean Reciprocal Rank (MRR), where  $100/\text{MRR}$  roughly indicates the average rank of the correct solution over the session.

Table 1 reports the results of several transliteration configurations we tested. The first two systems are pure analogical devices, (M) is the Moses configuration, (AM<sub>1</sub>) is a variant discussed further, (AM<sub>2</sub>) is the best configuration we tested (a combination of Moses and analogical learning), and the last two lines show the lowest and highest performing systems among the 18 standard runs registered at NEWS 2009 (Li et al., 2009). Several observations have to be made.

First, none of the variants tested outperformed the best system reported at NEWS 2009. This is not surprising since we conducted only preliminary experiments with analogy. Still, we were pleased to observe that the best configuration we devised (AM<sub>2</sub>) would have ranked fourth on this task.

<sup>4</sup>We used libSVM (Chang and Lin, 2011) for training svms, and an in-house package for training voted perceptrons.

<sup>5</sup>This is fair since there is no training involved. Many participants to the NEWS campaign did this as well.

<sup>6</sup>We avoided this in order to keep the classifiers simple to train.

<sup>7</sup><http://translit.i2r.a-star.edu.sg/news2009/evaluation/>.

The `ana-freq` system is an analogical device where the aggregation step consists in sorting solutions in decreasing order of frequency. It is clearly outperformed by the Moses system. The `ana-svma` system is an analogical device where the solutions are selected by the SVM trained on analogical features only. Learning to recognize good solutions from spurious ones improves accuracy (over  $A_1$ ). Still, we are far from the accuracy we would observe by using an oracle classifier (ACC = 81.5). Clearly, further experiments with better feature engineering must be conducted. It is noteworthy that the pure analogical devices we tested ( $A_1$  and  $A_2$ ) did not return any solution for 3.7% of the test forms, which explains some loss in performance compared to the SMT approach, which always delivers a solution.<sup>8</sup>

System `ana-svma+m` ( $AM_1$ ) is an analogical device where the classifier makes use of the features extracted by Moses. Obviously, those features drastically improve accuracy of the classifier. Configuration ( $AM_2$ ) is a combination which cascades the hybrid device ( $AM_1$ ) with the SMT engine (M). This means that the former system is trusted whenever it produces a solution, and the latter one is used as a backup. This configuration outperforms Moses, which demonstrates the complementarity of the analogical information.

| Configuration                             | ACC  | F <sub>1</sub> | MRR  | rank |
|---|------|----------------|------|------|
| $A_1$ <code>ana-freq</code>               | 56.6 | 79.1           | 63.0 | 16   |
| $A_2$ <code>ana-svm<sub>a</sub></code>    | 58.0 | 80.0           | 58.8 | 15   |
| M <code>moses</code>                      | 66.6 | 85.9           | 66.6 | 6    |
| $AM_1$ <code>ana-svm<sub>a+m</sub></code> | 63.4 | 82.0           | 64.1 | 10   |
| $AM_2$ $AM_1 + M$                         | 68.5 | 86.9           | 69.0 | 4    |
| last NEWS 2009                            | 19.9 | 60.6           | 22.9 | 23   |
| first NEWS 2009                           | 73.1 | 89.5           | 81.2 | 1    |

Table 1: Evaluation of different configurations with the metrics used at NEWS. The last column indicates the rank of systems as if we had submitted the top 5 configurations to NEWS 2009.

## 4 Related Work

Most approaches to transliteration we know rely on some form of substring alignment. This alignment can be learnt explicitly as in (Knight and

<sup>8</sup>Removing the solutions produced by the SMT engine for the 3.7% test forms that receive no solution from the analogical devices would result in an accuracy score of 65.0.

Graehl, 1998; Li et al., 2004; Jiampojarn et al., 2007), or it can be indirectly modeled as in (Oh et al., 2009) where transliteration is seen as a tagging task (that is, labeling each source grapheme with a target one), and where the model learns correspondences at the substring level. See also the semi-supervised approach of (Sajjad et al., 2012). Analogical inference differs drastically from those approaches, since it finds relations in the source material and solves target equations independently. Therefore, no alignment whatsoever is required.

Transliteration by analogical learning has been attempted by Dandapat et al. (2010) for an English-to-Hindi transliteration task. They compared various heuristics to speed up analogical learning, and several combinations of phrase-based SMT and analogical learning. Our results confirm the observation they made that combining an analogical device with SMT leads to gains over individual systems. Still, their work differs from the present one in the fact that they considered the top frequency aggregator (similar to  $A_1$ ), which we showed to be suboptimal. Also, they used the definition of formal analogy of Lepage (1998), which is provably less general than the one we used. The impact of this choice for different language pairs remains to be investigated.

Aggregating solutions produced by analogical inference with the help of a classifier has been reported in (Langlais et al., 2009). The authors investigated an arguably more specific task: translating medical terms. Another difference is that we classify *solutions* produced by analogical learning (roughly 100 solutions per test form), while they classified *pairs of input/target analogies*, whose number can be rather high, leading to huge and highly unbalanced learning tasks. The authors report training experiments with millions of examples and only a few positive ones. In fact, we initially attempted to recognize fruitful analogical pairs, but found it especially slow and disappointing.

## 5 Conclusion

We considered the NEWS 2009 English-to-Chinese transliteration task for investigating analogical learning, a holistic approach that does not rely on an alignment or segmentation model. We have shown that alone, the approach fails to translate 3.7% of the test forms, underperforms the state-of-the-art SMT engine Moses, while still de-

livering decent performance. By combining both approaches, we obtained a system which outperforms the individual ones we tested.

We believe analogical inference over strings has not delivered all his potential yet. In particular, we have observed that there is a huge room for improvements in the aggregation step. We have tested a simple classifier approach, mining a tiny subset of the features that could be put at use. More research on this issue is warranted, notably looking at machine-learned ranking algorithms.

Also, the silence issue we faced could be tackled by the notion of *analogical dissimilarity* introduced by Miclet et al. (2008). The idea of using near analogies in analogical learning has been successfully investigated by the authors on a number of standard classification testbeds.

## Acknowledgments

This work has been founded by the Natural Sciences and Engineering Research Council of Canada. We are grateful to Fabrizio Gotti for his contribution to this work, and to the anonymous reviewers for their useful comments. We are also indebted to Min Zhang and Haizhou Li who provided us with the NEWS 2009 English-Chinese datasets.

## References

- Aditya Bhargava and Grzegorz Kondrak. 2011. How do you pronounce your name? Improving G2P with transliterations. In *49th ACL/HLT*, pages 399–408, Portland, USA.
- Chih-Chung Chang and Chih-Jen Lin. 2011. LIB-SVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27, May.
- William Fernando Correa, Henri Prade, and Gilles Richard. 2012. When intelligence is just a matter of copying. In *20th ECAI*, pages 276–281, Montpellier, France.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Mach. Learn.*, 20(3):273–297.
- Sandipan Dandapat, Sara Morrissey, Sudip Kumar Naskar, and Harold Somers. 2010. Mitigating Problems in Analogy-based EBMT with SMT and vice versa: a Case Study with Named Entity Transliteration. In *24th Pacific Asia Conference on Language Information and Computation (PACLIC'10)*, pages 365–372, Sendai, Japan.
- Étienne Denoual. 2007. Analogical translation of unknown words in a statistical machine translation framework. In *MT Summit XI*, pages 135–141, Copenhagen, Denmark.
- Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2011. Cross-Language Latent Relational Search: Mapping Knowledge across Languages. In *25th AAAI*, pages 1237 – 1242, San Francisco, USA.
- Andrew Finch and Eiichiro Sumita. 2010. Transliteration Using a Phrase-based Statistical Machine Translation System to Re-score the Output of a Joint Multigram Model. In *2nd Named Entities Workshop (NEWS'10)*, pages 48–52, Uppsala, Sweden.
- Y. Freund and R. E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Mach. Learn.*, 37(3):277–296.
- Sittichai Jiampojarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *NAACL/HLT'07*, pages 372–379.
- Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Comput. Linguist.*, 24(4):599–612.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *NAACL/HLT'03*, pages 48–54, Edmonton, Canada.
- 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 *45th ACL*, pages 177–180. Interactive Poster and Demonstration Sessions.
- Philippe Langlais and Alexandre Patry. 2007. Translating Unknown Words by Analogical Learning. In *EMNLP/CoNLL'07*, pages 877–886, Prague, Czech Republic.
- Philippe Langlais and François Yvon. 2008. Scaling up Analogical Learning. In *22nd COLING*, pages 51–54, Manchester, United Kingdom. Poster.
- Philippe Langlais, François Yvon, and Pierre Zweigenbaum. 2009. Improvements in Analogical Learning: Application to Translating multi-Terms of the Medical Domain. In *12th EACL*, pages 487–495, Athens, Greece.
- Yves Lepage and Étienne Denoual. 2005. Purest ever example-based machine translation: Detailed presentation and assesment. *Mach. Translat.*, 19:25–252.
- Yves Lepage, Adrien Lardilleux, and Julien Gosme. 2009. The GREYC Translation Memory for the IWSLT 2009 Evaluation Campaign: one step beyond translation memory. In *6th IWSLT*, pages 45–49, Tokyo, Japan.



- Yves Lepage. 1998. Solving Analogies on Words: an Algorithm. In *COLING/ACL*, pages 728–733, Montreal, Canada.
- Haizhou Li, Min Zhang, and Jian Su. 2004. A Joint Source-Channel Model for Machine Transliteration. In *42nd ACL*, pages 159–166, Barcelona, Spain.
- Haizhou Li, A. Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 Machine Transliteration Shared Task. In *1st Named Entities Workshop (NEWS'09): Shared Task on Transliteration*, pages 1–18, Singapore.
- Laurent Miclet, Sabri Bayroudh, and Arnaud Delhay. 2008. Analogical Dissimilarity: Definitions, Algorithms and two experiments in Machine Learning. *Journal of Artificial Intelligence Research*, pages 793–824.
- Fabienne Moreau, Vincent Claveau, and Pascale Sébillot. 2007. Automatic Morphological Query Expansion Using Analogy-based Machine Learning. In *29th European Conference on IR research (ECIR'07)*, pages 222–233, Rome, Italy.
- Jong-hoon Oh, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Machine Transliteration using Target-Language Grapheme and Phoneme: Multi-engine Transliteration Approach. In *1st Named Entities Workshop (NEWS'09): Shared Task on Transliteration*, pages 36–39, Singapore.
- Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A Statistical Model for Unsupervised and Semi-supervised Transliteration Mining. In *50th ACL*, pages 469–477, Jeju Island, Korea.
- Harold Somers, Sandipan Sandapat, and Sudip Kumar Naskar. 2009. A Review of EBMT Using Proportional Analogies. In *3rd Workshop on Example-based Machine Translation*, pages 53–60, Dublin, Ireland.
- Nicolas Stroppa and François Yvon. 2005. An Analogical Learner for Morphological Analysis. In *9th CoNLL*, pages 120–127, Ann Arbor, USA.
- P.D. Turney and M.L. Littman. 2005. Corpus-based Learning of Analogies and Semantic Relations. In *Machine Learning*, volume 60, pages 251–278.
- Antal van den Bosch and Walter Daelemans. 1993. Data-Oriented Methods for Grapheme-to-Phoneme Conversion. In *6th EACL*, pages 45–53, Utrecht, Netherlands.
- François Yvon, Nicolas Stroppa, Arnaud Delhay, and Laurent Miclet. 2004. Solving Analogies on Words. Technical Report D005, École Nationale Supérieure des Télécommunications, Paris, France.
- François Yvon. 1997. Paradigmatic Cascades: a Linguistically Sound Model of Pronunciation by Analogy. In *35th ACL*, pages 429–435, Madrid, Spain.

# Scalable Modified Kneser-Ney Language Model Estimation

Kenneth Heafield<sup>\*,†</sup>

Ivan Pouzyrevsky<sup>‡</sup>

Jonathan H. Clark<sup>†</sup>

Philipp Koehn<sup>\*</sup>

<sup>\*</sup>University of Edinburgh  
10 Crichton Street  
Edinburgh EH8 9AB, UK

<sup>†</sup>Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213, USA

<sup>‡</sup>Yandex  
Zelenograd, bld. 455 fl. 128  
Moscow 124498, Russia

heafield@cs.cmu.edu ivan.pouzyrevsky@gmail.com jhclark@cs.cmu.edu pkoehn@inf.ed.ac.uk

## Abstract

We present an efficient algorithm to estimate large modified Kneser-Ney models including interpolation. Streaming and sorting enables the algorithm to scale to much larger models by using a fixed amount of RAM and variable amount of disk. Using one machine with 140 GB RAM for 2.8 days, we built an unpruned model on 126 billion tokens. Machine translation experiments with this model show improvement of 0.8 BLEU point over constrained systems for the 2013 Workshop on Machine Translation task in three language pairs. Our algorithm is also faster for small models: we estimated a model on 302 million tokens using 7.7% of the RAM and 14.0% of the wall time taken by SRILM. The code is open source as part of KenLM.

## 1 Introduction

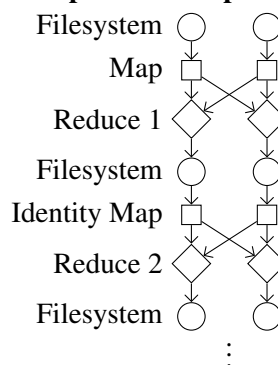
Relatively low perplexity has made modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998) a popular choice for language modeling. However, existing estimation methods require either large amounts of RAM (Stolcke, 2002) or machines (Brants et al., 2007). As a result, practitioners have chosen to use less data (Callison-Burch et al., 2012) or simpler smoothing methods (Brants et al., 2007).

Backoff-smoothed  $n$ -gram language models (Katz, 1987) assign probability to a word  $w_n$  in context  $w_1^{n-1}$  according to the recursive equation

$$p(w_n|w_1^{n-1}) = \begin{cases} p(w_n|w_1^{n-1}), & \text{if } w_1^n \text{ was seen} \\ b(w_1^{n-1})p(w_n|w_2^n), & \text{otherwise} \end{cases}$$

The task is to estimate probability  $p$  and backoff  $b$  from text for each seen entry  $w_1^n$ . This paper

## MapReduce Steps



## Optimized

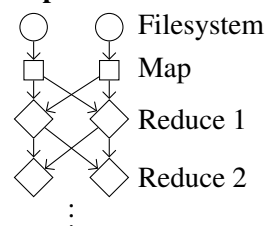


Figure 1: Each MapReduce performs three copies over the network when only one is required. Arrows denote copies over the network (i.e. to and from a *distributed* filesystem). Both options use local disk within each reducer for merge sort.

contributes an efficient multi-pass streaming algorithm using disk and a user-specified amount of RAM.

## 2 Related Work

Brants et al. (2007) showed how to estimate Kneser-Ney models with a series of five MapReduces (Dean and Ghemawat, 2004). On 31 billion words, estimation took 400 machines for two days. Recently, Google estimated a pruned Kneser-Ney model on 230 billion words (Chelba and Schalkwyk, 2013), though no cost was provided.

Each MapReduce consists of one layer of mappers and an optional layer of reducers. Mappers read from a network filesystem, perform optional processing, and route data to reducers. Reducers process input and write to a network filesystem. Ideally, reducers would send data directly to another layer of reducers, but this is not supported. Their workaround, a series of MapReduces, performs unnecessary copies over the network (Figure 1). In both cases, reducers use local disk.

Writing and reading from the distributed filesystem improves fault tolerance. However, the same level of fault tolerance could be achieved by checkpointing to the network filesystem then only reading in the case of failures. Doing so would enable reducers to start processing without waiting for the network filesystem to write all the data.

Our code currently runs on a single machine while MapReduce targets clusters. Appuswamy et al. (2013) identify several problems with the scale-out approach of distributed computation and put forward several scenarios in which a single machine scale-up approach is more cost effective in terms of both raw performance and performance per dollar.

Brants et al. (2007) contributed Stupid Backoff, a simpler form of smoothing calculated at runtime from counts. With Stupid Backoff, they scaled to 1.8 trillion tokens. We agree that Stupid Backoff is cheaper to estimate, but contend that this work makes Kneser-Ney smoothing cheap enough.

Another advantage of Stupid Backoff has been that it stores one value, a count, per  $n$ -gram instead of probability and backoff. In previous work (Heafield et al., 2012), we showed how to collapse probability and backoff into a single value without changing sentence-level probabilities. However, local scores do change and, like Stupid Backoff, are no longer probabilities.

MSRLM (Nguyen et al., 2007) aims to scalably estimate language models on a single machine. Counting is performed with streaming algorithms similarly to this work. Their parallel merge sort also has the potential to be faster than ours. The biggest difference is that their pipeline delays some computation (part of normalization and all of interpolation) until query time. This means that it cannot produce a standard ARPA file and that more time and memory are required at query time. Moreover, they use memory mapping on entire files and these files may be larger than physical RAM. We have found that, even with mostly-sequential access, memory mapping is slower because the kernel does not explicitly know where to read ahead or write behind. In contrast, we use dedicated threads for reading and writing. Performance comparisons are omitted because we were unable to compile and run MSRLM on recent versions of Linux.

SRILM (Stolcke, 2002) estimates modified Kneser-Ney models by storing  $n$ -grams in RAM.

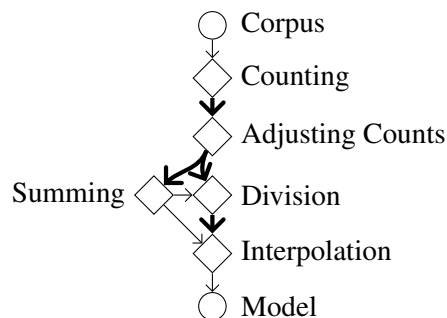


Figure 2: Data flow in the estimation pipeline. Normalization has two threads per order: summing and division. Thick arrows indicate sorting.

It also offers a disk-based pipeline for initial steps (i.e. counting). However, the later steps store all  $n$ -grams that survived count pruning in RAM. Without pruning, both options use the same RAM.

IRSTLM (Federico et al., 2008) does not implement modified Kneser-Ney but rather an approximation dubbed “improved Kneser-Ney” (or “modified shift-beta” depending on the version). Estimation is done in RAM. It can also split the corpus into pieces and separately build each piece, introducing further approximation.

### 3 Estimation Pipeline

Estimation has four streaming passes: counting, adjusting counts, normalization, and interpolation. Data is sorted between passes, three times in total. Figure 2 shows the flow of data.

#### 3.1 Counting

For a language model of order  $N$ , this step counts all  $N$ -grams (with length exactly  $N$ ) by streaming through the corpus. Words near the beginning of sentence also form  $N$ -grams padded by the marker  $\langle s \rangle$  (possibly repeated multiple times). The end of sentence marker  $\langle /s \rangle$  is appended to each sentence and acts like a normal token.

Unpruned  $N$ -gram counts are sufficient, so lower-order  $n$ -grams ( $n < N$ ) are not counted. Even pruned models require unpruned  $N$ -gram counts to compute smoothing statistics.

Vocabulary mapping is done with a hash table.<sup>1</sup> Token strings are written to disk and a 64-bit Mur-

<sup>1</sup>This hash table is the only part of the pipeline that can grow. Users can specify an estimated vocabulary size for memory budgeting. In future work, we plan to support local vocabularies with renumbering.

| Suffix |   |   | Context |   |   |
|--------|---|---|---------|---|---|
| 3      | 2 | 1 | 2       | 1 | 3 |
| Z      | B | A | Z       | A | B |
| Z      | A | B | B       | B | B |
| B      | B | B | Z       | B | A |

Figure 3: In suffix order, the last word is primary. In context order, the penultimate word is primary.

murHash<sup>2</sup> token identifier is retained in RAM.

Counts are combined in a hash table and spilled to disk when a fixed amount of memory is full. Merge sort also combines identical  $N$ -grams (Bitton and DeWitt, 1983).

### 3.2 Adjusting Counts

The counts  $c$  are replaced with adjusted counts  $a$ .

$$a(w_1^n) = \begin{cases} c(w_1^n), & \text{if } n = N \text{ or } w_1 = \langle s \rangle \\ |v : c(vw_1^n) > 0|, & \text{otherwise} \end{cases}$$

Adjusted counts are computed by streaming through  $N$ -grams sorted in suffix order (Figure 3). The algorithm keeps a running total  $a(w_i^N)$  for each  $i$  and compares consecutive  $N$ -grams to decide which adjusted counts to output or increment.

Smoothing statistics are also collected. For each length  $n$ , it collects the number  $t_{n,k}$  of  $n$ -grams with adjusted count  $k \in [1, 4]$ .

$$t_{n,k} = |\{w_1^n : a(w_1^n) = k\}|$$

These are used to compute closed-form estimates (Chen and Goodman, 1998) of discounts  $D_n(k)$

$$D_n(k) = k - \frac{(k+1)t_{n,1}t_{n,k+1}}{(t_{n,1} + 2t_{n,2})t_{n,k}}$$

for  $k \in [1, 3]$ . Other cases are  $D_n(0) = 0$  and  $D_n(k) = D_n(3)$  for  $k \geq 3$ . Less formally, counts 0 (unknown) through 2 have special discounts.

### 3.3 Normalization

Normalization computes pseudo probability  $u$

$$u(w_n | w_1^{n-1}) = \frac{a(w_1^n) - D_n(a(w_1^n))}{\sum_x a(w_1^{n-1}x)}$$

and backoff  $b$

$$b(w_1^{n-1}) = \frac{\sum_{i=1}^3 D_n(i) |\{x : a(w_1^{n-1}x) = i\}|}{\sum_x a(w_1^{n-1}x)}$$

<sup>2</sup><https://code.google.com/p/smhasher/>

The difficulty lies in computing denominator  $\sum_x a(w_1^{n-1}x)$  for all  $w_1^{n-1}$ . For this, we sort in context order (Figure 3) so that, for every  $w_1^{n-1}$ , the entries  $w_1^{n-1}x$  are consecutive. One pass collects both the denominator and backoff<sup>3</sup> terms  $|\{x : a(w_1^{n-1}x) = i\}|$  for  $i \in [1, 3]$ .

A problem arises in that denominator  $\sum_x a(w_1^{n-1}x)$  is known only after streaming through all  $w_1^{n-1}x$ , but is needed immediately to compute each  $u(w_n | w_1^{n-1})$ . One option is to buffer in memory, taking  $O(N|\text{vocabulary}|)$  space since each order is run independently in parallel. Instead, we use two threads for each order. The sum thread reads ahead to compute  $\sum_x a(w_1^{n-1}x)$  and  $b(w_1^{n-1})$  then places these in a secondary stream. The divide thread reads the input and the secondary stream then writes records of the form

$$(w_1^n, u(w_n | w_1^{n-1}), b(w_1^{n-1})) \quad (1)$$

The secondary stream is short so that data read by the sum thread will likely be cached when read by the divide thread. This sort of optimization is not possible with most MapReduce implementations.

Because normalization streams through  $w_1^{n-1}x$  in context order, the backoffs  $b(w_1^{n-1})$  are computed in suffix order. This will be useful later (§3.5), so backoffs are written to secondary files (one for each order) as bare values without keys.

### 3.4 Interpolation

Chen and Goodman (1998) found that perplexity improves when the various orders within the same model are interpolated. The interpolation step computes final probability  $p$  according to the recursive equation

$$p(w_n | w_1^{n-1}) = u(w_n | w_1^{n-1}) + b(w_1^{n-1})p(w_n | w_2^{n-1}) \quad (2)$$

Recursion terminates when unigrams are interpolated with the uniform distribution

$$p(w_n) = u(w_n) + b(\epsilon) \frac{1}{|\text{vocabulary}|}$$

where  $\epsilon$  denotes the empty string. The unknown word counts as part of the vocabulary and has count zero,<sup>4</sup> so its probability is  $b(\epsilon)/|\text{vocabulary}|$ .

<sup>3</sup>Sums and counts are done with exact integer arithmetic. Thus, every floating-point value generated by our toolkit is the result of  $O(N)$  floating-point operations. SRILM has numerical precision issues because it uses  $O(N|\text{vocabulary}|)$  floating-point operations to compute backoff.

<sup>4</sup>SRILM implements “another hack” that computes  $p_{\text{SRILM}}(w_n) = u(w_n)$  and  $p_{\text{SRILM}}(\langle \text{unk} \rangle) = b(\epsilon)$  whenever  $p(\langle \text{unk} \rangle) < 3 \times 10^{-6}$ , as it usually is. We implement both and suspect their motivation was numerical precision.

Probabilities are computed by streaming in suffix lexicographic order:  $w_n$  appears before  $w_{n-1}^n$ , which in turn appears before  $w_{n-2}^n$ . In this way,  $p(w_n)$  is computed before it is needed to compute  $p(w_n|w_{n-1})$ , and so on. This is implemented by jointly iterating through  $N$  streams, one for each length of  $n$ -gram. The relevant pseudo probability  $u(w_n|w_1^{n-1})$  and backoff  $b(w_1^{n-1})$  appear in the input records (Equation 1).

### 3.5 Joining

The last task is to unite  $b(w_1^n)$  computed in §3.3 with  $p(w_n|w_1^{n-1})$  computed in §3.4 for storage in the model. We note that interpolation (Equation 2) used the different backoff  $b(w_1^{n-1})$  and so  $b(w_1^n)$  is not immediately available. However, the backoff values were saved in suffix order (§3.3) and interpolation produces probabilities in suffix order. During the same streaming pass as interpolation, we merge the two streams.<sup>5</sup> Suffix order is also convenient because the popular reverse trie data structure can be built in the same pass.<sup>6</sup>

## 4 Sorting

Much work has been done on efficient disk-based merge sort. Particularly important is arity, the number of blocks that are merged at once. Low arity leads to more passes while high arity incurs more disk seeks. Abello and Vitter (1999) modeled these costs and derived an optimal strategy: use fixed-size read buffers (one for each block being merged) and set arity to the number of buffers that fit in RAM. The optimal buffer size is hardware-dependent; we use 64 MB by default. To overcome the operating system limit on file handles, multiple blocks are stored in the same file.

To further reduce the costs of merge sort, we implemented pipelining (Dementiev et al., 2008). If there is enough RAM, input is lazily merged and streamed to the algorithm. Output is cut into blocks, sorted in the next step’s desired order, and then written to disk. These optimizations eliminate up to two copies to disk if enough RAM is available. Input, the algorithm, block sorting, and output are all threads on a chain of producer-consumer queues. Therefore, computation and disk operations happen simultaneously.

<sup>5</sup>Backoffs only exist if the  $n$ -gram is the context of some  $n + 1$ -gram, so merging skips  $n$ -grams that are not contexts.

<sup>6</sup>With quantization (Whittaker and Raj, 2001), the quantizer is trained in a first pass and applied in a second pass.

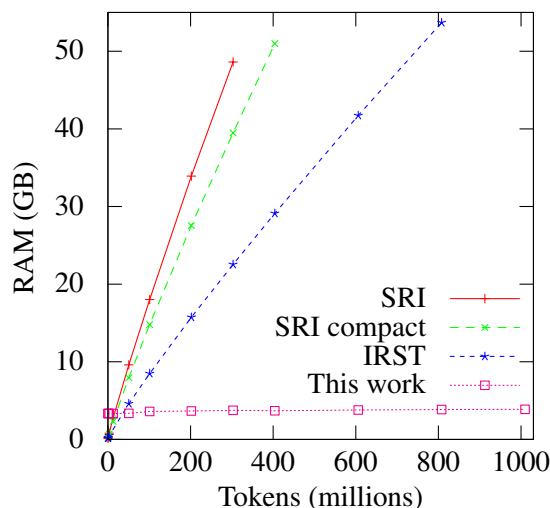


Figure 4: Peak virtual memory usage.

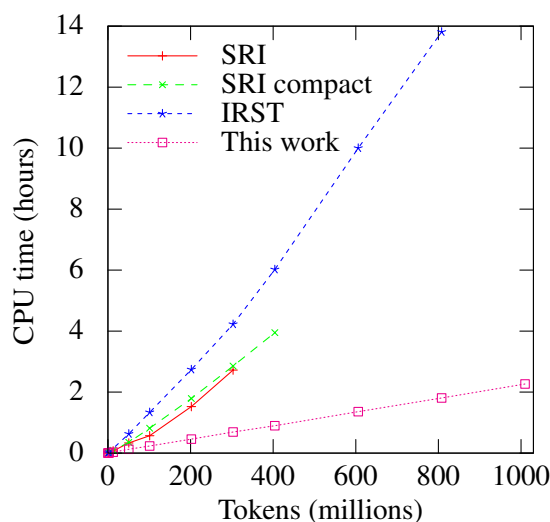


Figure 5: CPU usage (system plus user).

Each  $n$ -gram record is an array of  $n$  vocabulary identifiers (4 bytes each) and an 8-byte count or probability and backoff. At peak, records are stored twice on disk because lazy merge sort is not easily amenable to overwriting the input file. Additional costs are the secondary backoff file (4 bytes per backoff) and the vocabulary in plaintext.

## 5 Experiments

Experiments use ClueWeb09.<sup>7</sup> After spam filtering (Cormack et al., 2011), removing markup, selecting English, splitting sentences (Koehn, 2005), deduplicating, tokenizing (Koehn et al., 2007), and truecasing, 126 billion tokens remained.

<sup>7</sup><http://lemurproject.org/clueweb09/>

| 1   | 2     | 3      | 4      | 5      |
|-----|-------|--------|--------|--------|
| 393 | 3,775 | 17,629 | 39,919 | 59,794 |

Table 1: Counts of unique  $n$ -grams (in millions) for the 5 orders in the large LM.

### 5.1 Estimation Comparison

We estimated unpruned language models in binary format on sentences randomly sampled from ClueWeb09. SRILM and IRSTLM were run until the test machine ran out of RAM (64 GB). For our code, the memory limit was set to 3.5 GB because larger limits did not improve performance on this small data. Results are in Figures 4 and 5. Our code used an average of 1.34–1.87 CPUs, so wall time is better than suggested in Figure 5 despite using disk. Other toolkits are single-threaded. SRILM’s partial disk pipeline is not shown; it used the same RAM and took more time. IRSTLM’s splitting approximation took 2.5 times as much CPU and about one-third the memory (for a 3-way split) compared with normal IRSTLM.

For 302 million tokens, our toolkit used 25.4% of SRILM’s CPU time, 14.0% of the wall time, and 7.7% of the RAM. Compared with IRSTLM, our toolkit used 16.4% of the CPU time, 9.0% of the wall time, and 16.6% of the RAM.

### 5.2 Scaling

We built an unpruned model (Table 1) on 126 billion tokens. Estimation used a machine with 140 GB RAM and six hard drives in a RAID5 configuration (sustained read: 405 MB/s). It took 123 GB RAM, 2.8 days wall time, and 5.4 CPU days. A summary of Google’s results from 2007 on different data and hardware appears in §2.

We then used this language model as an additional feature in unconstrained Czech-English, French-English, and Spanish-English submissions to the 2013 Workshop on Machine Translation.<sup>8</sup> Our baseline is the University of Edinburgh’s phrase-based Moses (Koehn et al., 2007) submission (Durrani et al., 2013), which used all constrained data specified by the evaluation (7 billion tokens of English). It placed first by BLEU (Papineni et al., 2002) among constrained submissions in each language pair we consider.

In order to translate, the large model was quantized (Whittaker and Raj, 2001) to 10 bits and compressed to 643 GB with KenLM (Heafield,

<sup>8</sup><http://statmt.org/wmt13/>

| Source  | Baseline | Large |
|---------|----------|-------|
| Czech   | 27.4     | 28.2  |
| French  | 32.6     | 33.4  |
| Spanish | 31.8     | 32.6  |

Table 2: Uncased BLEU results from the 2013 Workshop on Machine Translation.

2011) then copied to a machine with 1 TB RAM. Better compression methods (Guthrie and Hepple, 2010; Talbot and Osborne, 2007) and distributed language models (Brants et al., 2007) could reduce hardware requirements. Feature weights were re-tuned with PRO (Hopkins and May, 2011) for Czech-English and batch MIRA (Cherry and Foster, 2012) for French-English and Spanish-English because these worked best for the baseline. Uncased BLEU scores on the 2013 test set are shown in Table 2. The improvement is remarkably consistent at 0.8 BLEU point in each language pair.

## 6 Conclusion

Our open-source (LGPL) estimation code is available from [kheafield.com/code/kenlm/](http://kheafield.com/code/kenlm/) and should prove useful to the community. Sorting makes it scalable; efficient merge sort makes it fast. In future work, we plan to extend to the Common Crawl corpus and improve parallelism.

## Acknowledgements

Miles Osborne preprocessed ClueWeb09. Mohammed Mediani contributed to early designs. Jianfeng Gao clarified how MSRLM operates. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. We used Stampede and Trestles under allocation TG-CCR110017. System administrators from the Texas Advanced Computing Center (TACC) at The University of Texas at Austin made configuration changes on our request. This work made use of the resources provided by the Edinburgh Compute and Data Facility (<http://www.ecdf.ed.ac.uk/>). The ECDF is partially supported by the eDIKT initiative (<http://www.edikt.org.uk/>). The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 287658 (EU BRIDGE).

## References

- James M. Abello and Jeffrey Scott Vitter, editors. 1999. *External memory algorithms*. American Mathematical Society, Boston, MA, USA.
- Raja Appuswamy, Christos Gkantsidis, Dushyanth Narayanan, Orion Hodson, and Antony Rowstron. 2013. Nobody ever got fired for buying a cluster. Technical Report MSR-TR-2013-2, Microsoft Research.
- Dina Bitton and David J DeWitt. 1983. Duplicate record elimination in large data files. *ACM Transactions on database systems (TODS)*, 8(2):255–265.
- Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 858–867, June.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.
- Ciprian Chelba and Johan Schalkwyk. 2013. *Empirical Exploration of Language Modeling for the google.com Query Stream as Applied to Mobile Voice Search*, pages 197–229. Springer, New York.
- Stanley Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, August.
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.
- Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. 2011. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465.
- Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified data processing on large clusters. In *OSDI’04: Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, USA, 12.
- Roman Dementiev, Lutz Kettner, and Peter Sanders. 2008. STXXL: standard template library for XXL data sets. *Software: Practice and Experience*, 38(6):589–637.
- Nadir Durrani, Barry Haddow, Kenneth Heafield, and Philipp Koehn. 2013. Edinburgh’s machine translation systems for European language pairs. In *Proceedings of the ACL 2013 Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August.
- Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, Brisbane, Australia.
- David Guthrie and Mark Hepple. 2010. Storing the web in memory: Space efficient language models with constant time retrieval. In *Proceedings of EMNLP 2010*, Los Angeles, CA.
- Kenneth Heafield, Philipp Koehn, and Alon Lavie. 2012. Language model rest costs and space-efficient storage. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea.
- Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July. Association for Computational Linguistics.
- Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, July.
- Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(3):400–401, March.
- 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*, pages 181–184.
- 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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, June.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.
- Patrick Nguyen, Jianfeng Gao, and Milind Mahajan. 2007. MSRLM: a scalable language modeling toolkit. Technical Report MSR-TR-2007-144, Microsoft Research.

Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, July.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of ACL*, pages 512–519, Prague, Czech Republic.

Edward Whittaker and Bhiksha Raj. 2001. Quantization-based language model compression. In *Proceedings of Eurospeech*, pages 33–36, Aalborg, Denmark, December.



# Incremental Topic-Based Translation Model Adaptation for Conversational Spoken Language Translation

Sanjika Hewavitharana, Dennis N. Mehay, Sankaranarayanan Ananthakrishnan and Prem Natarajan

Speech, Language and Multimedia Business Unit  
Raytheon BBN Technologies  
Cambridge, MA 02138, USA

{shewavit,dmehay,sanantha,pnataraj}@bbn.com

## Abstract

We describe a translation model adaptation approach for conversational spoken language translation (CSLT), which encourages the use of contextually appropriate translation options from relevant training conversations. Our approach employs a monolingual LDA topic model to derive a similarity measure between the test conversation and the set of training conversations, which is used to bias translation choices towards the current context. A significant novelty of our adaptation technique is its *incremental* nature; we continuously update the topic distribution on the evolving test conversation as new utterances become available. Thus, our approach is well-suited to the causal constraint of spoken conversations. On an English-to-Iraqi CSLT task, the proposed approach gives significant improvements over a baseline system as measured by BLEU, TER, and NIST. Interestingly, the incremental approach outperforms a non-incremental oracle that has up-front knowledge of the whole conversation.

## 1 Introduction

Conversational spoken language translation (CSLT) systems facilitate communication between subjects who do not speak the same language. Current systems are typically used to achieve a specific task (e.g. vehicle checkpoint search, medical diagnosis, etc.). These task-driven

conversations typically revolve around a set of central topics, which may not be evident at the beginning of the interaction. As the conversation progresses, however, the gradual accumulation of contextual information can be used to infer the topic(s) of discussion, and to deploy contextually appropriate translation phrase pairs. For example, the word ‘*drugs*’ will predominantly translate into Spanish as ‘*medicamentos*’ (medicines) in a medical scenario, whereas the translation ‘*drogas*’ (illegal drugs) will predominate in a law enforcement scenario. Most CSLT systems do not take high-level global context into account, and instead translate each utterance in isolation. This often results in contextually inappropriate translations, and is particularly problematic in conversational speech, which usually exhibits short, spontaneous, and often ambiguous utterances.

In this paper, we describe a novel topic-based adaptation technique for phrase-based statistical machine translation (SMT) of spoken conversations. We begin by building a monolingual latent Dirichlet allocation (LDA) topic model on the training conversations (each conversation corresponds to a “document” in the LDA paradigm). At run-time, this model is used to infer a topic distribution over the evolving test conversation up to and including the current utterance. Translation phrase pairs that originate in training conversations whose topic distribution is similar to that of the current conversation are given preference through a single similarity feature, which augments the standard phrase-based SMT log-linear model. The topic distribution for the test conversation is updated incrementally for each new utterance as the available history grows. With this approach, we demonstrate significant improvements over a baseline phrase-based SMT system as measured by BLEU, TER and NIST scores on an English-to-Iraqi CSLT task.

Disclaimer: This paper is based upon work supported by the DARPA BOLT program. The views expressed here are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Distribution Statement A (Approved for Public Release, Distribution Unlimited)

## 2 Relation to Prior Work

Domain adaptation to improve SMT performance has attracted considerable attention in recent years (Foster and Kuhn, 2007; Finch and Sumita, 2008; Matsoukas et al., 2009). The general theme is to divide the training data into partitions representing different domains, and to prefer translation options for a test sentence from training domains that most resemble the current document context. Weaknesses of this approach include (a) assuming the existence of discrete, non-overlapping domains; and (b) the unreliability of models generated by segments with little training data.

To avoid the need for hard decisions about domain membership, some have used topic modeling to improve SMT performance, e.g., using latent semantic analysis (Tam et al., 2007) or ‘biTAM’ (Zhao and Xing, 2006). In contrast to our source language approach, these authors use both source and target information.

Perhaps most relevant are the approaches of Gong et al. (2010) and Eidelman et al. (2012), who both describe adaptation techniques where monolingual LDA topic models are used to obtain a topic distribution over the training data, followed by dynamic adaptation of the phrase table based on the inferred topic of the test document. While our proposed approach also employs monolingual LDA topic models, it deviates from the above methods in the following important ways. First, the existing approaches are geared towards batch-mode text translation, and assume that the full document context of a test sentence is always available. This assumption is incompatible with translation of spoken conversations, which are inherently causal. Our proposed approach infers topic distributions *incrementally* as the conversation progresses. Thus, it is not only consistent with the causal requirement, but is also capable of tracking topical changes during the course of a conversation.

Second, we do not directly augment the translation table with the inferred topic distribution. Rather, we compute a similarity between the current conversation history and each of the training conversations, and use this measure to dynamically score the relevance of candidate translation phrase pairs during decoding.

## 3 Corpus Data and Baseline SMT

We use the DARPA TransTac English-Iraqi parallel two-way spoken dialogue collection to train both translation and LDA topic models. This data set contains a variety of scenarios, including medical diagnosis; force protection (e.g. checkpoint, reconnaissance, patrol); aid, maintenance and infrastructure, etc.; each transcribed from spoken bilingual conversations and manually translated. The SMT parallel training corpus contains approximately 773K sentence pairs (7.3M English words). We used this corpus to extract translation phrase pairs from bidirectional IBM Model 4 word alignment (Och and Ney, 2003) based on the heuristic approach of (Koehn et al., 2003). A 4-gram target LM was trained on all Iraqi Arabic transcriptions. Our phrase-based decoder is similar to Moses (Koehn et al., 2007) and uses the phrase pairs and target LM to perform beam search stack decoding based on a standard log-linear model, the parameters of which were tuned with MERT (Och, 2003) on a held-out development set (3,534 sentence pairs, 45K words) using BLEU as the tuning metric. Finally, we evaluated translation performance on a separate, unseen test set (3,138 sentence pairs, 38K words).

Of the 773K training sentence pairs, about 100K (corresponding to 1,600 conversations) are marked with conversation boundaries. We use the English side of these conversations for training LDA topic models. All other sentence pairs are assigned to a “background conversation”, which signals the absence of the topic similarity feature for phrase pairs derived from these instances. All of the development and test set data were marked with conversation boundaries. The training, development and test sets were partitioned at the conversation level, so that we could model a topic distribution for entire conversations, both during training and during tuning and testing.

## 4 Incremental Topic-Based Adaptation

Our approach is based on the premise that biasing the translation model to favor phrase pairs originating in training conversations that are contextually similar to the current conversation will lead to better translation quality. The topic distribution is incrementally updated as the conversation history grows, and we recompute the topic similarity between the current conversation and the training conversations for each new source utterance.

## 4.1 Topic modeling with LDA

We use latent Dirichlet allocation, or LDA, (Blei et al., 2003) to obtain a topic distribution over conversations. For each conversation  $d_i$  in the training collection (1,600 conversations), LDA infers a topic distribution  $\theta_{d_i} = p(z_k|d_i)$  for all latent topics  $z_k = \{1, \dots, K\}$ , where  $K$  is the number of topics. In this work, we experiment with values of  $K \in \{20, 30, 40\}$ . The full conversation history is available for training the topic models and estimating topic distributions in the training set.

At run-time, however, we construct the conversation history for the tuning and test sets incrementally, one utterance at a time, mirroring a real-world scenario where our knowledge is limited to the utterances that have been spoken up to that point in time. Thus, each development/test utterance is associated with a different conversation history  $d^*$ , for which we infer a topic distribution  $\theta_{d^*} = p(z_k|d^*)$  using the trained LDA model. We use Mallet (McCallum, 2002) for training topic models and inferring topic distributions.

## 4.2 Topic Similarity Computation

For each test utterance, we are able to infer the topic distribution  $\theta_{d^*}$  based on the accumulated history of the current conversation. We use this to compute a measure of similarity between the evolving test conversation and each of the training conversations, for which we already have topic distributions  $\theta_{d_i}$ . Because  $\theta_{d_i}$  and  $\theta_{d^*}$  are probability distributions, we use the Jensen-Shannon divergence (JSD) to evaluate their similarity (Manning and Schütze, 1999). The JSD is a smoothed and symmetric version of Kullback-Leibler divergence, which is typically used to compare two probability distributions. We define the similarity score as  $sim(\theta_{d_i}, \theta_{d^*}) = 1 - JSD(\theta_{d_i} || \theta_{d^*})$ .<sup>1</sup> Thus, we obtain a vector of similarity scores indexed by the training conversations.

## 4.3 Integration with the Decoder

We provide the SMT decoder with the similarity vector for each test utterance. Additionally, the SMT phrase table tracks, for each phrase pair, the set of parent training conversations (including the “background conversation”) from which that phrase pair originated. Using this information, the decoder evaluates, for each candidate phrase pair

<sup>1</sup> $JSD(\theta_{d_i} || \theta_{d^*}) \in [0, 1]$  when defined using  $\log_2$ .

| REFERENCE TRANSCRIPTIONS |               |               |              |
|--------------------------|---------------|---------------|--------------|
| SYSTEM                   | BLEU↑         | TER↓          | NIST↑        |
| <b>Baseline</b>          | 19.32         | 58.66         | 6.22         |
| incr20                   | 19.39         | 58.44         | 6.26*        |
| incr30                   | 19.36         | 58.32*        | 6.26         |
| incr40                   | <b>19.68*</b> | <b>58.19*</b> | <b>6.28*</b> |
| conv20                   | 19.60*        | 58.36*        | 6.27*        |
| conv30                   | 19.48         | 58.38*        | 6.27*        |
| conv40                   | 19.50         | 58.33*        | 6.28*        |
| ASR TRANSCRIPTIONS       |               |               |              |
| SYSTEM                   | BLEU↑         | TER↓          | NIST↑        |
| <b>Baseline</b>          | 16.92         | 62.57         | 5.75         |
| incr20                   | 16.99         | 62.28*        | 5.77         |
| incr30                   | 16.96         | 62.33*        | 5.78         |
| incr40                   | <b>17.31*</b> | <b>61.97*</b> | <b>5.83*</b> |
| conv20                   | 17.29*        | 62.28*        | 5.81*        |
| conv30                   | 17.12         | 62.19*        | 5.80*        |
| conv40                   | 17.00         | 62.14*        | 5.79*        |

Table 1: Stemmed results on 3,138-utterance test set. Asterisked results are significantly better than the baseline ( $p \leq 0.05$ ) using 1,000 iterations of paired bootstrap re-sampling (Koehn, 2004). (**Key:** incr $N$  = incremental LDA with  $N$  topics; conv $N$  = non-incremental, whole-conversation LDA with  $N$  topics.)

$X \rightarrow Y$  added to the search graph, its topic similarity score as follows:

$$F_{X \rightarrow Y} = \max_{i \in Par(X \rightarrow Y)} sim(\theta_{d_i}, \theta_{d^*}) \quad (1)$$

where  $Par(X \rightarrow Y)$  is the set of training conversations from which the candidate phrase pair originated. Phrase pairs from the “background conversation” only are assigned a similarity score  $F_{X \rightarrow Y} = 0.00$ . In this way we distill the inferred topic distributions down to a single feature for each candidate phrase pair. We add this feature to the log-linear translation model with its own weight, which is tuned with MERT. The intuition behind this feature is that the lower bound of suitability of a candidate phrase pair should be directly proportional to the similarity between its most relevant conversational provenance and the current context. Phrase pairs which only occur in the background conversation are not directly penalized, but contribute nothing to the topic similarity score.

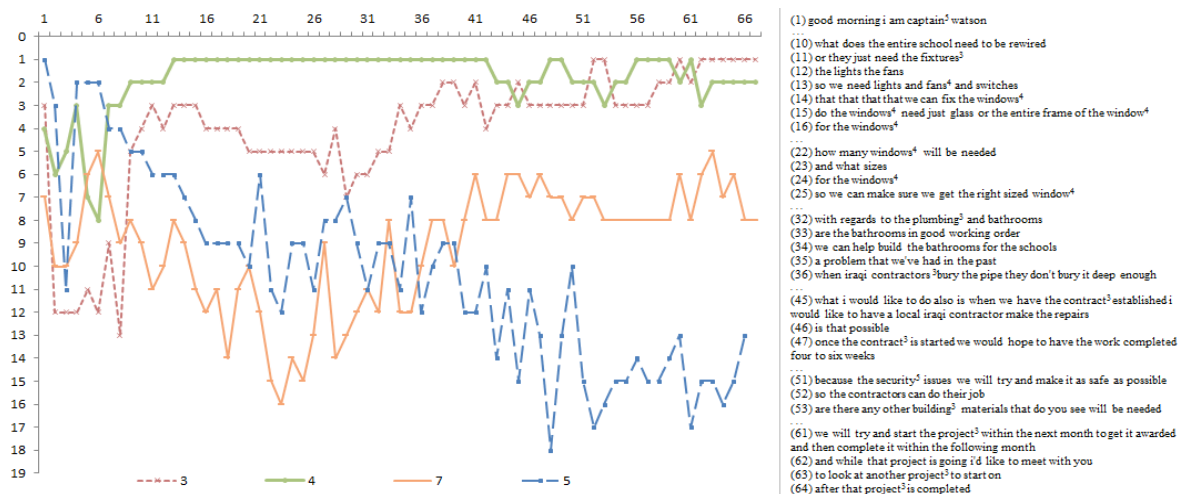


Figure 1: Rank trajectories of 4 LDA inferred topics, with incremental topic inference. The x-axis indicates the utterance number. The y-axis indicates a topic’s rank at each utterance.

## 5 Experimental Setup and Results

The baseline English-to-Iraqi phrase-based SMT system was built as described in Section 3. This system translated each utterance independently, ignoring higher-level conversational context.

For the topic-adapted system, we compared translation performance with a varying number of LDA topics. In intuitive agreement with the approximate number of scenario types known to be covered by our data set, a range of 20-40 topics yielded the best results. We compared the proposed incremental topic tracking approach to a non-causal oracle approach that had up-front access to the entire source conversations at run-time.

In all cases, we compared translation performance on both clean-text and automatic speech recognition (ASR) transcriptions of the source utterances. ASR transcriptions were generated using a high-performance two-pass HMM-based system, which delivered a word error rate (WER) of 10.6% on the test set utterances.

Table 1 summarizes test set performance in BLEU (Papineni et al., 2001), NIST (Doddington, 2002) and TER (Snover et al., 2006). Given the morphological complexity of Iraqi Arabic, computing string-based metrics on raw output can be misleadingly low and does not always reflect whether the core message was conveyed. Since the primary goal of CSLT is information transfer, we present automatic results that are computed after stemming with an Iraqi Arabic stemmer.

We note that in all settings (incremental and non-causal oracle) our adaptation approach

matches or significantly outperforms the baseline across multiple evaluation metrics. In particular, the incremental LDA system with 40 topics is the top-scoring system in both clean-text and ASR settings. In the ASR setting, which simulates a real-world deployment scenario, this system achieves improvements of 0.39 (BLEU), -0.6 (TER) and 0.08 (NIST).

## 6 Discussion and Future Directions

We have presented a novel, incremental topic-based translation model adaptation approach that obeys the causality constraint imposed by spoken conversations. This approach yields statistically significant gains in standard MT metric scores.

We have also demonstrated that incremental adaptation on an evolving conversation performs better than oracle adaptation based on the complete conversation history. Although this may seem counter-intuitive, Figure 1 gives clues as to why this happens. This figure illustrates the rank trajectory of four LDA topics as the incremental conversation grows. The accompanying text shows excerpts from the conversation. We indicate (in superscript) the topic identity of most relevant words in an utterance that are associated with that topic. At the first utterance, the top-ranked topic is “5”, due to the occurrence of “captain” in the greeting. As the conversation evolves, we note that this topic become less prominent. The conversation shifts to a discussion on “windows”, raising the prominence of topic “4”. Finally, topic “3” becomes prominent due to the presence of the

words “project” and “contract”. Thus, the incremental approach is able to track the topic trajectories in the conversation, and is able to select more relevant phrase pairs than oracle LDA, which estimates one topic distribution for the entire conversation.

In this work we have used only the source language utterance in inferring the topic distribution. In a two-way CLST system, we also have access to SMT-generated back-translations in the Iraqi-English direction. As a next step, we plan to use SMT-generated English translation of Iraqi utterances to improve topic estimation.

## References

- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 115–119, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 208–215, Stroudsburg, PA, USA. Association for Computational Linguistics.
- George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhengxian Gong, Yu Zhang, and Guodong Zhou. 2010. Statistical machine translation based on LDA. In *Universal Communication Symposium (IUCS), 2010 4th International*, pages 286–290.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395, Barcelona, Spain, July.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.
- Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings AMTA*, pages 223–231, August.
- Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual LSA-based adaptation for statistical machine translation. *Machine Translation*, 21(4):187–207, December.
- Bing Zhao and Eric P. Xing. 2006. BiTAM: Bilingual topic admixture models for word alignment. In *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*.

# A Lightweight and High Performance Monolingual Word Aligner

Xuchen Yao and Benjamin Van Durme

Johns Hopkins University  
Baltimore, MD, USA

Chris Callison-Burch\*  
University of Pennsylvania  
Philadelphia, PA, USA

Peter Clark  
Vulcan Inc.  
Seattle, WA, USA

## Abstract

Fast alignment is essential for many natural language tasks. But in the setting of monolingual alignment, previous work has not been able to align more than one sentence pair per second. We describe a discriminatively trained monolingual word aligner that uses a Conditional Random Field to globally decode the best alignment with features drawn from source and target sentences. Using just part-of-speech tags and WordNet as external resources, our aligner gives state-of-the-art result, while being an order-of-magnitude faster than the previous best performing system.

## 1 Introduction

In statistical machine translation, alignment is typically done as a one-off task during training. However for monolingual tasks, like recognizing textual entailment or question answering, alignment happens repeatedly: once or multiple times per test item. Therefore, the efficiency of the aligner is of utmost importance for monolingual alignment tasks. Monolingual word alignment also has a variety of distinctions than the bilingual case, for example: there is often less training data but more lexical resources available; semantic relatedness may be cued by distributional word similarities; and, both the source and target sentences share the same grammar.

These distinctions suggest a model design that utilizes arbitrary features (to make use of word similarity measure and lexical resources) and exploits deeper sentence structures (especially in the case of major languages where robust parsers are available). In this setting the balance between precision and speed becomes an issue: while we might leverage an extensive NLP pipeline for a

language like English, such pipelines can be computationally expensive. One earlier attempt, the MANLI system (MacCartney et al., 2008), used roughly 5GB of lexical resources and took 2 seconds per alignment, making it hard to be deployed and run in large scale. On the other extreme, a simple non-probabilistic Tree Edit Distance (TED) model (c.f. §4.2) is able to align 10,000 pairs per second when the sentences are pre-parsed, but with significantly reduced performance. Trying to embrace the merits of both worlds, we introduce a discriminative aligner that is able to align tens to hundreds of sentence pairs per second, and needs access only to a POS tagger and WordNet.

This aligner gives state-of-the-art performance on the MSR RTE2 alignment dataset (Brockett, 2007), is faster than previous work, and we release it publicly as the first open-source monolingual word aligner: *Jacana.Align*.<sup>1</sup>

## 2 Related Work

The MANLI aligner (MacCartney et al., 2008) was first proposed to align premise and hypothesis sentences for the task of natural language inference. It applies perceptron learning and handles phrase-based alignment of arbitrary phrase lengths. Thadani and McKeown (2011) optimized this model by decoding via Integer Linear Programming (ILP). Benefiting from modern ILP solvers, this led to an order-of-magnitude speedup. With extra syntactic constraints added, the exact alignment match rate for whole sentence pairs was also significantly improved.

Besides the above supervised methods, indirect supervision has also been explored. Among them, Wang and Manning (2010) extended the work of McCallum et al. (2005) and modeled alignment as latent variables. Heilman and Smith (2010) used tree kernels to search for the alignment that

\*Performed while faculty at Johns Hopkins University.

<sup>1</sup><http://code.google.com/p/jacana/>

yields the lowest tree edit distance. Other tree or graph matching work for alignment includes that of (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Chambers et al., 2007; Mehdad, 2009; Roth and Frank, 2012).

Finally, feature and model design in monolingual alignment is often inspired by bilingual work, including distortion modeling, phrasal alignment, syntactic constraints, etc (Och and Ney, 2003; DeNero and Klein, 2007; Bansal et al., 2011).

### 3 The Alignment Model

#### 3.1 Model Design

Our work is heavily influenced by the bilingual alignment literature, especially the discriminative model proposed by Blunsom and Cohn (2006). Given a source sentence  $\mathbf{s}$  of length  $M$ , and a target sentence  $\mathbf{t}$  of length  $N$ , the alignment from  $\mathbf{s}$  to  $\mathbf{t}$  is a sequence of target word indices  $\mathbf{a}$ , where  $a_{m \in [1, M]} \in [0, N]$ . We specify that when  $a_m = 0$ , source word  $s_t$  is aligned to a NULL state, i.e., deleted. This models a many-to-one alignment from source to target. Multiple source words can be aligned to the same target word, but not vice versa. One-to-many alignment can be obtained by running the aligner in the other direction. The probability of alignment sequence  $\mathbf{a}$  conditioned on both  $\mathbf{s}$  and  $\mathbf{t}$  is then:

$$p(\mathbf{a} | \mathbf{s}, \mathbf{t}) = \frac{\exp(\sum_{m,k} \lambda_k f_k(a_{m-1}, a_m, \mathbf{s}, \mathbf{t}))}{Z(\mathbf{s}, \mathbf{t})}$$

This assumes a first-order Conditional Random Field (Lafferty et al., 2001). The word alignment task is evaluated over  $F_1$ . Instead of directly optimizing  $F_1$ , we employ softmax-margin training (Gimpel and Smith, 2010) and add a cost function to the normalizing function  $Z(\mathbf{s}, \mathbf{t})$  in the denominator, which becomes:

$$\sum_{\hat{\mathbf{a}}} \exp(\sum_{m,k} \lambda_k f_k(\hat{a}_{m-1}, \hat{a}_m, \mathbf{s}, \mathbf{t}) + cost(\mathbf{a}_t, \hat{\mathbf{a}}))$$

where  $\mathbf{a}_t$  is the true alignments.  $cost(\mathbf{a}_t, \hat{\mathbf{a}})$  can be viewed as special “features” with uniform weights that encourage consistent with true alignments. It is only computed during training in the denominator because  $cost(\mathbf{a}_t, \mathbf{a}_t) = 0$  in the numerator. Hamming cost is used in practice.

One distinction of this alignment model compared to other commonly defined CRFs is that

the input is two dimensional: at each position  $m$ , the model inspects both the entire sequence of source words (as the observation) and target words (whose offset indices are states). The other distinction is that the size of its state space is not fixed (e.g., unlike POS tagging, where states are for instance 45 Penn Treebank tags), but depends on  $N$ , the length of target sentence. Thus we can not “memorize” what features are mostly associated with what states. For instance, in the task of tagging mail addresses, a feature of “5 consecutive digits” is highly indicative of a POSTCODE. However, in the alignment model, it does not make sense to design features based on a hard-coded state, say, a feature of “source word lemma matching target word lemma” fires for state index 6.

To avoid this data sparsity problem, all features are defined *implicitly* with respect to the state. For instance:

$$f_k(a_{m-1}, a_m, \mathbf{s}, \mathbf{t}) = \begin{cases} 1 & \text{lemmas match: } s_m, t_{a_m} \\ 0 & \text{otherwise} \end{cases}$$

Thus this feature fires for, e.g.:  
 $(s_3 = \text{sport}, t_5 = \text{sports}, a_3 = 5)$ , and:  
 $(s_2 = \text{like}, t_{10} = \text{liked}, a_2 = 10)$ .

#### 3.2 Feature Design

**String Similarity Features** include the following similarity measures: Jaro Winkler, Dice Sorensen, Hamming, Jaccard, Levenshtein, NGram overlapping and common prefix matching.<sup>2</sup> Also, two binary features are added for identical match and identical match ignoring case.

**POS Tags Features** are binary indicators of whether the POS tags of two words match. Also, a “ $pos_{src}2pos_{tgt}$ ” feature fires for each word pair, with respect to their POS tags. This would capture, e.g., “vbz2nn”, when a verb such as *arrests* aligns with a noun such as *custody*.

**Positional Feature** is a real-valued feature for the positional difference of the source and target word ( $abs(\frac{m}{M} - \frac{a_m}{N})$ ).

**WordNet Features** indicate whether two words are of the following relations of each other: hypernym, hyponym, synonym, derived form, entailing, causing, members of, have member, substances of, have substances, parts of, have part; or whether

<sup>2</sup>Of these features the trained aligner preferred Dice Sorensen and NGram overlapping.

their lemmas match.<sup>3</sup>

**Distortion Features** measure how far apart the aligned target words of two consecutive source words are:  $\text{abs}(a_m + 1 - a_{m-1})$ . This learns a general pattern of whether these two target words aligned with two consecutive source words are usually far away from each other, or very close. We also added special features for corner cases where the current word starts or ends the source sentence, or both the previous and current words are deleted (a transition from NULL to NULL).

**Contextual Features** indicate whether the left or the right neighbor of the source word and aligned target word are identical or similar. This helps especially when aligning functional words, which usually have multiple candidate target functional words to align to and string similarity features cannot help. We also added features for neighboring POS tags matching.

### 3.3 Symmetrization

To expand from many-to-one alignment to many-to-many, we ran the model in both directions and applied the following symmetrization heuristics (Koehn, 2010): INTERSECTION, UNION, GROW-DIAG-FINAL.

## 4 Experiments

### 4.1 Setup

Since no generic off-the-shelf CRF software is designed to handle the special case of dynamic state indices and feature functions (Blunsom and Cohn, 2006), we implemented this aligner model in the Scala programming language, which is fully interoperable with Java. We used the L2 regularizer and LBFGS for optimization. OpenNLP<sup>4</sup> provided the POS tagger and JWNL<sup>5</sup> interfaced with WordNet (Fellbaum, 1998).

To make results directly comparable, we closely followed the setup of MacCartney et al. (2008) and Thadani and McKeown (2011). Training and test data (Brockett, 2007) each contains 800 manually aligned premise and hypothesis pairs from RTE2. Note that the premises contain 29 words on average, and the hypotheses only 11 words. We take the premise as the source and hypothesis as the target, and use S2T to indicate the model aligns from

<sup>3</sup>We found that each word has to be POS tagged to get an accurate relation, otherwise this feature will not help.

<sup>4</sup><http://opennlp.apache.org/>

<sup>5</sup><http://jwordnet.sf.net/>

source to target and T2S from target to source.

### 4.2 Simple Baselines

We additionally used two baseline systems for comparison. One was GIZA++, with the INTERSECTION tricks post-applied, which worked the best among all other symmetrization heuristics. The other was a Tree Edit Distance (TED) model, popularly used in a series of NLP applications (Punyakanok et al., 2004; Kouylekov and Magnini, 2005; Heilman and Smith, 2010). We used uniform cost for deletion, insertion and substitutions, and applied a dynamic program algorithm (Zhang and Shasha, 1989) to decode the tree edit sequence with the minimal cost, based on the Stanford dependency tree (De Marneffe and Manning, 2008). This non-probabilistic approach turned out to be extremely fast, processing about 10,000 sentence pairs per second with pre-parsed trees, performing quantitatively better than the Stanford RTE aligner (Chambers et al., 2007).

### 4.3 MANLI Baselines

MANLI was first developed by MacCartney et al. (2008), and then improved by Thadani and McKeown (2011) with faster and exact decoding via ILP. There are four versions to be compared here:

**MANLI** the original version.

**MANLI-approx.** re-implemented version by Thadani and McKeown (2011).

**MANLI-exact** decoding via ILP solvers.

**MANLI-constraint** MANLI-exact with hard syntactic constraints, mainly on common “light” words (determiners, prepositions, etc.) attachment to boost exact match rate.

### 4.4 Results

Following Thadani and McKeown (2011), performance is evaluated by macro-averaged precision, recall,  $F_1$  of aligned token pairs, and exact (perfect) match rate for a whole pair, shown in Table 1. As our baselines, GIZA++ (with alignment intersection of two directions) and TED are on par with previously reported results using the Stanford RTE aligner. The MANLI-family of systems provide stronger baselines, notably MANLI-constraint, which has the best  $F_1$  and exact match rate among themselves.

We ran our aligner in two directions: S2T and T2S, then merged the results with INTERSECTION, UNION and GROW-DIAG-FINAL. Our system beats



| System                           | P %         | R %         | F <sub>1</sub> % | E %         |
|----------------------------------|-------------|-------------|------------------|-------------|
| GIZA++, $\cap$                   | 82.5        | 74.4        | 78.3             | 14.0        |
| TED                              | 80.6        | 79.0        | 79.8             | 13.5        |
| Stanford RTE*                    | 82.7        | 75.8        | 79.1             | -           |
| MANLI*                           | 85.4        | 85.3        | 85.3             | 21.3        |
| MANLI-approx. $\triangleleft$    | 87.2        | 86.3        | 86.7             | 24.5        |
| MANLI-exact $\triangleleft$      | 87.2        | 86.1        | 86.8             | 24.8        |
| MANLI-constraint $\triangleleft$ | 89.5        | 86.2        | 87.8             | 33.0        |
| this work, S2T                   | 91.8        | 83.4        | 87.4             | 25.9        |
| this work, T2S                   | 93.7        | 84.0        | <b>88.6</b>      | <b>35.3</b> |
| S2T $\cap$ T2S                   | <b>95.4</b> | 80.8        | 87.5             | 31.3        |
| S2T $\cup$ T2S                   | 90.3        | <b>86.6</b> | 88.4             | 29.6        |
| GROW-DIAG-FINAL                  | 94.4        | 81.8        | 87.6             | 30.8        |

Table 1: Results on the 800 pairs of test data. E% stands for exact (perfect) match rate. Systems marked with \* are reported by MacCartney et al. (2008), with  $\triangleleft$  by Thadani and McKeown (2011).

the weak and strong baselines<sup>6</sup> in all measures except recall. Some patterns are very clearly shown: **Higher precision, lower recall** is due to the higher-quality and lower-coverage of WordNet, where the MANLI-family systems used additional, automatically derived lexical resources.

**Imbalance of exact match rate** between S2T and T2S with a difference of 9.4% is due to the many-to-one nature of the aligner. When aligning from source (longer) to target (shorter), multiple source words can align to the same target word. This is not desirable since multiple duplicate “light” words are aligned to the same “light” word in the target, which breaks perfect match. When aligning T2S, this problem goes away: the shorter target sentence contains less duplicate words, and in most cases there is an one-to-one mapping.

**MT heuristics** help, with INTERSECTION and UNION respectively improving precision and recall.

#### 4.5 Runtime Test

Table 2 shows the runtime comparison. Since the RTE2 corpus is imbalanced, with premise length (words) of 29 and hypothesis length of 11, we also compare on the corpus of FUSION (McKeown et al., 2010), with both sentences in a pair averaging 27. MANLI-approx. is the slowest, with quadratic growth in the number of edits with sentence length. MANLI-exact is in second place, relying on the ILP solver. This work has a precise  $O(MN^2)$  decoding time, with  $M$  the source sentence length and  $N$  the target sentence length.

<sup>6</sup>Unfortunately both MacCartney and Thadani no longer have their original output files (personal communication), so we cannot run a significance test against their result.

| corpus | sent. pair length | MANLI-approx. | MANLI-exact | this work |
|--------|-------------------|---------------|-------------|-----------|
| RTE2   | 29/11             | 1.67          | 0.08        | 0.025     |
| FUSION | 27/27             | 61.96         | 2.45        | 0.096     |

Table 2: Alignment runtime in seconds per sentence pair on two corpora: RTE2 (Cohn et al., 2008) and FUSION (McKeown et al., 2010). The MANLI-\* results are from Thadani and McKeown (2011), on a Xeon 2.0GHz with 6MB Cache. The runtime for this work takes the longest timing from S2T and T2S, on a Xeon 2.2GHz with 4MB cache (the closest we can find to match their hardware). Horizontally in a real-world application where sentences have similar length, this work is roughly 20x faster (0.096 vs. 2.45). Vertically, the decoding time for our work increases less dramatically when sentence length increases (0.025→0.096 vs. 0.08→2.45).

| features   | P %  | R %  | F1 % | E %  |
|------------|------|------|------|------|
| full (T2S) | 93.7 | 84.0 | 88.6 | 35.3 |
| - POS      | 93.2 | 83.5 | 88.1 | 31.4 |
| - WordNet  | 93.2 | 83.7 | 88.2 | 33.5 |
| - both     | 93.1 | 83.2 | 87.8 | 30.1 |

Table 3: Performance without POS and/or WordNet features.

While MANLI-exact is about twenty-fold faster than MANLI-approx., our aligner is at least another twenty-fold faster than MANLI-exact when the sentences are longer and balanced. We also benefit from shallower pre-processing (no parsing) and can store all resources in main memory.<sup>7</sup>

#### 4.6 Ablation Test

Since WordNet and the POS tagger is the only used external resource, we removed them<sup>8</sup> from the feature sets and reported performance in Table 3. This somehow reflects how the model would perform for a language without a suitable POS tagger, or more commonly, WordNet in that language. At this time, the model falls back to relying on string similarities, distortion, positional and contextual features, which are almost language-independent. A loss of less than 1% in  $F_1$  suggests that the aligner can still run reasonably well without a POS tagger and WordNet.

<sup>7</sup>WordNet (~30MB) is a smaller footprint than the 5GB of external resources used by MANLI.

<sup>8</sup>per request of reviewers. Note that WordNet is less precise without a POS tagger. When we removed the POS tagger, we enumerated all POS tags for a word to find its hypernym/synonym/... synsets.

## 4.7 Error Analysis

There were three primary categories of error:<sup>9</sup>

1. Token-based paraphrases that are not covered by WordNet, such as *program* and *software*, *business* and *venture*. This calls for broader-coverage paraphrase resources.
2. Words that are semantically related but not exactly paraphrases, such as *married* and *wife*, *beat* and *victory*. This calls for resources of close distributional similarity.
3. *Phrases* of the above kinds, such as *elected* and *won a seat*, *politician* and *presidential candidate*. This calls for further work on phrase-based alignment.<sup>10</sup>

There is a trade-off using WordNet vs. larger, noisier resources in exchange of higher precision vs. recall and memory/disk allocation. We think this is an application-specific decision; other resources could be easily incorporated into our model, which we may explore in the future to explore the trade-off in addressing items 1 and 2.

## 5 Conclusion

We presented a model for monolingual sentence alignment that gives state-of-the-art performance, and is significantly faster than prior work. We release our implementation as the first open-source monolingual aligner, which we hope to be of benefit to other researchers in the rapidly expanding area of natural language inference.

## Acknowledgement

We thank Vulcan Inc. for funding this work. We also thank Jason Smith, Travis Wolfe, Frank Ferraro for various discussion, suggestion, comments and the three anonymous reviewers.

## References

Mohit Bansal, Chris Quirk, and Robert Moore. 2011. Gappy phrasal alignment by agreement. In *Proceedings of ACL*, Portland, Oregon, June.

<sup>9</sup>We submitted a browser in JavaScript (AlignmentBrowser.html) in the supporting material that compares the gold alignment and test output; readers are encouraged to try it out.

<sup>10</sup>Note that MacCartney et al. (2008) showed that in the MANLI system setting phrase size to larger than one there was only a 0.2% gain in  $F_1$ , while the complexity became much larger.

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of ACL2006*, pages 65–72.

Chris Brockett. 2007. Aligning the RTE 2006 corpus. Technical report, Microsoft Research.

N. Chambers, D. Cer, T. Grenager, D. Hall, C. Kid-don, B. MacCartney, M.C. de Marneffe, D. Ramage, E. Yeh, and C.D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170.

Trevor Cohn, Chris Callison-Burch, and Mirella Lap-ata. 2008. Constructing corpora for the develop-ment and evaluation of paraphrase systems. *Comput. Linguist.*, 34(4):597–614, December.

Marie-Catherine De Marneffe and Christopher D Man-ning. 2008. The stanford typed dependencies rep-resentation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of ACL2007*.

C. Fellbaum. 1998. *WordNet: An Electronical Lexical Database*.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin crfs: training log-linear models with cost functions. In *NAACL 2010*, pages 733–736.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, para-phrases, and answers to questions. In *Proceedings of NAACL 2010*, pages 1011–1019, Los Angeles, Cali-fornia, June.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA.

Milen Kouylekov and Bernardo Magnini. 2005. Rec-ognizing textual entailment with tree edit distance algorithms. In *PASCAL Challenges on RTE*, pages 17–20.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling se-quence data. In *Proceedings of the Eighteenth Inter-national Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA.

B. MacCartney, M. Galley, and C.D. Manning. 2008. A phrase-based alignment model for natural lan-guage inference. In *Proceedings of EMNLP2008*, pages 802–811.

Andrew McCallum, Kedar Bellare, and Fernando Pereira. 2005. A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, July.

- Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *ACL2010 short*, pages 317–320.
- Y. Mehdad. 2009. Automatic cost estimation for tree edit distance using particle swarm optimization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 289–292.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- Vasin Punyakanok, Dan Roth, and Wen T. Yih. 2004. Mapping Dependencies Trees: An Application to Question Answerin. In *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics*, Fort Lauderdale, Florida.
- Michael Roth and Anette Frank. 2012. Aligning predicates across monolingual comparable texts using graph-based clustering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 171–182, Jeju Island, Korea, July.
- Kapil Thadani and Kathleen McKeown. 2011. Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *Proceedings of ACL short*.
- Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 1164–1172, Stroudsburg, PA, USA.
- K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, December.

# A Learner Corpus-based Approach to Verb Suggestion for ESL

Yu Sawai

Mamoru Komachi\*

Yuji Matsumoto

Graduate School of Information Science  
Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma, Nara, 630-0192, Japan  
{yu-s, komachi, matsu}@is.naist.jp

## Abstract

We propose a verb suggestion method which uses candidate sets and domain adaptation to incorporate error patterns produced by ESL learners. The candidate sets are constructed from a large scale learner corpus to cover various error patterns made by learners. Furthermore, the model is trained using both a native corpus and the learner corpus via a domain adaptation technique. Experiments on two learner corpora show that the candidate sets increase the coverage of error patterns and domain adaptation improves the performance for verb suggestion.

Previous work on verb selection usually treats the task as a multi-class classification problem (Wu et al., 2010; Wang and Hirst, 2010; Liu et al., 2010; Liu et al., 2011). In this formalization, it is important to restrict verbs by a **candidate set** because verb vocabulary is more numerous than other classes, such as determiners. Candidate sets for verb selection are often extracted from thesauri and/or round-trip translations. However, these resources may not cover certain error patterns found in actual learner corpora, and suffer from low-coverage. Furthermore, all the existing classifier models are trained only using a native corpus, which may not be adequate for correcting learner errors.

## 1 Introduction

In this study, we address verb selection errors in the writing of English learners. Selecting the right verb based on the context of a sentence is difficult for the learners of English as a Second Language (ESL). This error type is one of the most common errors in various learner corpora ranging from elementary to proficient levels<sup>1</sup>.

*They ?connect/communicate with other businessmen and do their jobs with the help of computers.*<sup>2</sup>

This sentence is grammatically acceptable with either verb. However, native speakers of English would less likely use “connect”, which means “forming a relationship (with other businessmen)”, whereas “communicate” means “exchanging information or ideas”, which is what the sentence is trying to convey.

\*Now at Tokyo Metropolitan University.

<sup>1</sup>For example, in the CLC-FCE dataset, the replacement error of verbs is the third most common out of 75 error types. In the KJ corpus, lexical choice of verb is the sixth most common out of 47 error types.

<sup>2</sup>This sentence is taken from the CLC-FCE dataset.

In this paper, we propose to use error patterns in ESL writing for verb suggestion task by using candidate sets and a domain adaptation technique. First, to increase the coverage, candidate sets are extracted from a large scale learner corpus derived from a language learning website. Second, a domain adaptation technique is applied to the model to fill the gap between two domains: native corpus and ESL corpus. Experiments are carried out on publicly available learner corpora, the Cambridge Learner Corpus First Certificate of English dataset (CLC-FCE) and the Konan JIEM corpus (KJ). The results show that the proposed candidate sets improve the coverage, compared to the baseline candidate sets derived from the WordNet and a round-trip translation table. Domain adaptation also boosts the suggestion performance.

To our knowledge, this is the first work for verb suggestion that uses (1) a learner corpus as a source of candidate sets and (2) the domain adaptation technique to take learner errors into account.

## 2 Verb Suggestion Considering Error Patterns

The proposed verb suggestion system follows the standard approach in related tasks (Rozovskaya and Roth, 2011; Wu et al., 2010), where the candidate selection is formalized as a multi-class classification problem with predefined candidate sets.

### 2.1 Candidate Sets

For reflecting tendency of learner errors to the candidate sets, we use a large scale corpus obtained from learners' writing on an SNS (Social Networking Service), *Lang-8*<sup>3</sup>. An advantage of using the learner corpus from such website is the size of annotated portion (Mizumoto et al., 2011). This SNS has over 1 million manually annotated English sentences written by ESL learners. We have collected the learner writings on the site, and released the dataset for research purpose<sup>4</sup>.

First, we performed POS tagging for the dataset using the treebank POS tagger in the NLTK toolkit 2.10. Second, we extracted the correction pairs which have "VB\*" tag. The set of correction pairs given an incorrect verb is considered as a candidate set for the verb.

We then performed the following preprocessing for the dataset because we focus on lexical selection of verbs:

- Lemmatize verbs to reduce data sparseness.
- Remove non-English verbs using WordNet.
- Remove incorrect verbs which occur only once in the dataset.

The target verbs are limited to the 500 most common verbs in the CLC-FCE corpus<sup>5</sup>. Therefore, verbs that do not appear in the target list are not included in the candidate sets. The topmost 500 verbs cover almost 90 percent of the vocabulary of verbs in the CLC-FCE corpus<sup>6</sup>.

The average number of candidates in a set is 20.3<sup>7</sup>. Note that the number of candidates varies across each target verb<sup>8</sup>.

<sup>3</sup><http://lang-8.com>

<sup>4</sup>Further details can be found at <http://cl.naist.jp/nldata/lang-8/>. Candidate sets will also be available at the same URL.

<sup>5</sup>They are extracted from all "VB" tagged tokens, and they contain 1,292 unique verbs after removing non-English words.

<sup>6</sup>This number excludes "be".

<sup>7</sup>In this paper, we limit the maximum number of candidates in each set to 50.

<sup>8</sup>For instance, the candidate set for "get" has 315 correction pairs, whereas "refund" has only 4.

### 2.2 Suggestion Model

The verb suggestion model consists of multi-class classifiers for each target verb; and based on the classifiers' output, it suggests alternative verbs. Instances are in a fill-in-the-blank format, where the labels are verbs. Features in this format are extracted from the surrounding context of a verb. When testing on the learner corpus, the model suggests a ranking of the possible verbs for the blank corresponding to a given context. Note that unlike the fill-in-the-blank task, the candidate sets and domain adaptation can be applied to this task to take the original word into account.

The model is trained on a huge native corpus, namely the ukWaC corpus, because the data-size of learner corpora is limited compared to native corpora. It is then adapted to the target domain, i.e., learner writing. In our experiment, the Lang-8 corpus is used as the target domain corpus, since we assume that it shares the same characteristics with the CLC-FCE and the KJ corpora used for testing.

### 2.3 Domain Adaptation

To adapt the models to the learner corpus, we employ a domain adaptation technique to emphasize the importance of learner domain information. Although there are many studies on domain adaptation, we chose to use **Feature Augmentation** technique introduced by (Daumé III, 2007) for its simplicity. Recently, (Imamura et al., 2012) proposed to apply this method to grammatical error correction for writings of Japanese learners and confirmed that this is more effective for correcting learner errors than simply adding the target domain instances.

In this study, the source domain is the native writing, and the target domain is the ESL writing. Our motivation is to use the ESL corpus together with the huge native corpus to employ both an advantage of the size of training data and the ESL writing specific features.

In this method, adapting a model to another model is achieved by extending the feature space. Given a feature vector of  $F$  dimensions as  $x \in \mathbb{R}^F (F > 0)$ , using simple mapping, the augmented feature vectors for source and target domains are obtained as follows,

$$\text{Source domain: } \langle x_S, x_S, \mathbf{0} \rangle \quad (1)$$

$$\text{Target domain: } \langle x_T, \mathbf{0}, x_T \rangle \quad (2)$$

where  $\mathbf{0}$  denotes a zero-vector of  $F$  dimensions. The three partitions mean a common, a source-specific, and a target-specific feature space. When testing on the ESL corpora, the target-specific features are emphasized.

## 2.4 Features

In previous work, various features were used: lexical and POS n-grams, dependencies, and arguments in the verb context. (Liu et al., 2011) has shown that shallow parse features, such as lexical n-grams and chunks, work well in realistic settings, in which the input sentence may not be correctly parsed. Considering this, we use shallow parse features as context features for robustness.

The features include lexical and POS n-grams, and lexical head words of the nearest NPs, and clustering features of these head words. An example of extracted features is shown in Table 2.4. Note that those features are also used when extracting examples from the target domain dataset (the learner domain corpus). As shown in Table 2.4, the n-gram features are 3-gram and extracted from  $\pm 2$  context window. The nearest NP’s head features are divided into two (Left, Right).

The additional clustering features are used for reducing sparseness, because the NP’s head words are usually proper nouns. To create the word clusters, we employ Brown clustering, a hierarchical clustering algorithm proposed by (Brown et al., 1992). The structure of clusters is a complete binary tree, in which each node is represented as a bit-string. By varying the length of the prefix of bit-string, it is possible to change the granularity of cluster representation. As illustrated in Table 2.4, we use the clustering features with three levels of granularity: 256, 128, and 64 dimensions. We used Percy Liang’s implementation<sup>9</sup> to create 256 dimensional model from the ukWaC corpus, which is used as the native corpus.

## 3 Experiments

Performance of verb suggestion is evaluated on two error-tagged learner corpora: CLC-FCE and KJ. In the experiments, we assume that the target verb and its context for suggestion are already given.

For the experiment on the CLC-FCE dataset, the targets are all words tagged with “RV” (re-

<sup>9</sup><https://github.com/percyliang/brown-cluster>

| Feature                       | Example  |
|-------------------------------|--|
| n-grams (surface)             | they-*V*-with<br><S>-they-*V*<br>*V*-with-other                    |
| n-grams (POS)                 | PRP-*V*-IN<br><S>-PRP-*V*<br>*V*-IN-JJ                             |
| NP head (Left, Right)         | L_they, L_PRP<br>R_businessmen, R_NNS                              |
| NP head cluster (Left, Right) | L_01110001, L_0111000, L_011100<br>R_11011001, R_1101100, R_110110 |

(e.g., *They (communicate) with other businessmen and do their jobs with the help of computers.*)

“<S>” denotes the beginning of the sentence, “\*V\*” denotes the blanked out verb.

Table 1: Example of extracted features as the fill-in-the-blank form.

placement error of verbs). We assume that all the verb selection errors are covered with this error tag. All error tagged parts with nested correction or multi-word expressions are excluded. The resulting number of “true” targets is 1,083, which amounts to 4% of all verbs. Therefore the dataset is highly skewed to correct usages, though this setting expresses well the reality of ESL writing, as shown in (Chodorow et al., 2012).

We carried out experiments with a variety of resources used for creating candidate sets.

- **WordNet**

Candidates are retrieved from the synsets and verbs sharing the same hypernyms in the WordNet 3.0.

- **LearnerSmall**

Candidates are retrieved from following learner corpora: NUS corpus of learner English (NUCLE), Konan-JIEM (KJ), and NICT Japanese learner English (JLE) corpus.

- **Roundtrip**

Candidates are collected by performing “round-trip” translation, which is similar to (Bannard and Callison-Burch, 2005)<sup>10</sup>.

- **WordNet+Roundtrip**

A combination of the thesaurus-based and the translation table-based candidate sets, similar to (Liu et al., 2010) and (Liu et al., 2011).

- **Lang-8**

The proposed candidate sets obtained from a large scale learner corpus.

- **Lang-8+DA**

Lang-8 candidate sets with domain adapta-

<sup>10</sup>Our roundtrip translation lexicons are built using a subset of the WIT<sup>3</sup> corpus (Cettolo et al., 2012), which is available at <http://wit3.fbk.eu>.

| Settings             | Candidates/set (Avg.) |
|----------------------|-----------------------|
| WordNet              | 14.8                  |
| LearnerSmall         | 5.1                   |
| Roundtrip            | 50                    |
| Roundtrip (En-Ja-En) | 50                    |
| WordNet+Roundtrip    | 50                    |
| Lang-8               | 20.3                  |

Table 2: Comparison of candidate set size for each setting.

tion via feature augmentation.

Table 3 shows a comparison of the average number of candidates in each setting. In all configurations above, the parameters of the models underlying the system are identical. We used a L2-regularized generalized linear model with log-loss function via Scikit-learn ver. 0.13.

### Inter-corpus Evaluation

We also evaluate the suggestion performance on the KJ corpus. The corpus contains diary-style writing by Japanese university students. The proficiency of the learners ranges from elementary to intermediate, so it is lower than that of the CLC-FCE learners. The targets are all verbs tagged with “v\_lxc” (lexical selection error of verbs).

To see the effect of L1 on the verb suggestion task, we added an alternative setting for the Roundtrip using only the English-Japanese and Japanese-English round-trip translation tables (En-Ja-En). For the experiment on this test-corpus, the LearnerSmall is not included.

### Datasets

The ukWaC web-corpus (Ferraresi et al., 2008) is used as a native corpus for training the suggestion model. Although this corpus consists of over 40 million sentences, 20,000 randomly selected sentences are used for each verb<sup>11</sup>.

The Lang-8 learner corpus is used for domain adaptation of the model in the Lang-8+DA configuration. The portion of data is the same as that used for constructing candidate sets.

### Metrics

Mean Reciprocal Rank (MRR) is used for evaluating the performance of alternative suggestions. The mean reciprocal rank is calculated by taking

<sup>11</sup>e.g., a classifier with a candidate set containing 50 verbs is trained with 1 million sentences in total.

the average of the reciprocal ranks for each instance. Given  $r\_gold_i$  as the position of the gold correction candidate in the suggestion list  $S_i$  for  $i$ -th checkpoint, the reciprocal rank  $RR_i$  is defined as,

$$RR_i = \begin{cases} \frac{1}{r\_gold_i} & (gold_i \in S_i) \\ 0 & (\text{otherwise}) \end{cases} \quad (3)$$

## 4 Results

Tables 5 and 5 show the results of suggestion performance on the CLC-FCE dataset and the KJ corpus, respectively. In both cases, the Lang-8 and its domain adaptation variant outperformed the others. The coverage of error patterns in the tables is the percentage of the cases where the suggestion list includes the gold correction. Generally, the suggestion performance and the coverage improve as the size of the candidate sets increases.

## 5 Discussions

Although the expert-annotated learner corpora contain candidates which are more reliable than a web-crawled Lang-8 corpus, the Lang-8 setting performed better as shown in Table 5. This can be explained by the broader coverage by the Lang-8 candidate sets than that of the LearnerSmall. Similarly, the WordNet performed the worst because it contains only synonym-like candidates. We can conclude that, for the verb suggestion task, the coverage (recall) of candidate sets is more important than the quality (precision).

We see little influence of learners’ L1 in the results of Table 5, since the Roundtrip performed better than the Roundtrip (En-Ja-En). As already mentioned, the number of error patterns contained in the candidate sets seems to have more importance than the quality.

As shown in Tables 5 and 5, a positive effect of domain adaptation technique appeared in both test-corpora. In the case of the CLC-FCE, 280 out of 624 suggestions were improved compared to the setting without domain adaptation. For instance, confusions between synonyms such as “?live/stay”, “?say/tell”, and “?solve/resolve” are improved, because sentences containing these confusions appear more frequently in the Lang-8 corpus. Although the number of test-cases for the KJ corpus is smaller than the CLC-FCE, we can see the improvements for 33 out of 66 sug-

| Settings          | MRR           | Coverage |
|-------------------|---------------|----------|
| WordNet           | 0.066         | 14.0 %   |
| LearnerSmall      | 0.128         | 23.5 %   |
| Roundtrip         | 0.185         | 48.1 %   |
| WordNet+Roundtrip | 0.173         | 48.1 %   |
| Lang-8            | 0.220         | 57.6 %   |
| <b>Lang-8+DA</b>  | <b>0.269*</b> | 57.6 %   |

The value marked with the asterisk indicates statistically significant improvement over the baselines, where  $p < 0.05$  bootstrap test.

Table 3: Suggestion performance on the CLC-FCE dataset.

| Settings             | MRR           | Coverage |
|----------------------|---------------|----------|
| WordNet              | 0.044         | 5.0 %    |
| Roundtrip            | 0.241         | 53.8 %   |
| Roundtrip (En-Ja-En) | 0.188         | 38.8 %   |
| WordNet+Roundtrip    | 0.162         | 53.8 %   |
| Lang-8               | 0.253         | 68.9 %   |
| <b>Lang-8+DA</b>     | <b>0.412*</b> | 68.9 %   |

The value marked with the asterisk indicates statistically significant improvement over the baselines, except “Roundtrip”, where  $p < 0.05$  bootstrap test.

Table 4: Suggestion performance on the KJ corpus.

gestions. The improvements appeared for frequent confusions of Japanese ESL learners such as “?see/watch” and “?tell/teach”.

Comparing the results of the Lang-8+DA on both test-corpora, the domain adaptation technique worked more effectively on the KJ corpus than on the CLC-FCE. This can be explained by the fact that the style of writing of the additional data, i.e., the Lang-8 corpus, is closer to KJ than it is to CLC-FCE. More precisely, unlike the examination-type writing style of CLC-FCE, the KJ corpus consists of diary writing similar in style to the Lang-8 corpus, and it expresses more closely the proficiency of the learners.

We think that the next step is to refine the suggestion models, since we currently take a simple fill-in-the-blank approach. As future work, we plan to extend the models as follows: (1) use both incorrect and correct sentences in learner corpora for training, and (2) employ ESL writing specific features such as learners’ L1 for domain adaptation.

## Acknowledgments

We thank YangYang Xi of Lang-8, Inc. for kindly allowing us to use the Lang-8 learner corpus. We also thank the anonymous reviewers for their insightful comments. This work was partially supported by Microsoft Research CORE Project.

## References

- Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604.
- Peter F Brown, Vincent J Della Pietra, Peter V DeSouza, Jenifer C Lai, Robert L Mercer, and Vincent J Della Pietra. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, December.
- M Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT<sup>3</sup>: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 28–30.
- Martin Chodorow, Markus Dickinson, Ross Israel, and Joel Tetreault. 2012. Problems in Evaluating Grammatical Error Detection Systems. In *Proceedings of the 24th International Conference on Computational Linguistics (Coling2012)*, pages 611–628.
- Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.
- Adriano Ferraresi, Eros Zanchetta, and Marco Baroni. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, pages 45–54.
- Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 388–392.
- Xiaohua Liu, Bo Han, Kuan Li, Stephan Hyeonjun Stiller, and Ming Zhou. 2010. SRL-based verb selection for ESL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1068–1076.
- Xiaohua Liu, Bo Han, and Ming Zhou. 2011. Correcting verb selection errors for ESL with the perceptron. In *12th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 411–423.



- 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 the 5th International Joint Conference on Natural Language Processing*, pages 147–155.
- Alla Rozovskaya and Dan Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 924–933.
- Tong Wang and Graeme Hirst. 2010. Near-synonym lexical choice in latent semantic space. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1182–1190.
- Jian-Cheng Wu, Yu-Chia Chang, Teruko Mitamura, and Jason S Chang. 2010. Automatic collocation suggestion in academic writing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics Short Papers*, pages 115–119.

# Learning Semantic Textual Similarity with Structural Representations

Aliaksei Severyn<sup>(1)</sup> and Massimo Nicosia<sup>(1)</sup> and Alessandro Moschitti<sup>1,2</sup>

<sup>(1)</sup>DISI, University of Trento, 38123 Povo (TN), Italy

{severyn,m.nicosia,moschitti}@disi.unitn.it

<sup>(2)</sup>QCRI, Qatar Foundation, Doha, Qatar

amoschitti@qf.org.qa

## Abstract

Measuring semantic textual similarity (STS) is at the cornerstone of many NLP applications. Different from the majority of approaches, where a large number of pairwise similarity features are used to represent a text pair, our model features the following: (i) it directly encodes input texts into relational syntactic structures; (ii) relies on tree kernels to handle feature engineering automatically; (iii) combines both structural and feature vector representations in a single scoring model, i.e., in Support Vector Regression (SVR); and (iv) delivers significant improvement over the best STS systems.

## 1 Introduction

In STS the goal is to learn a scoring model that given a pair of two short texts returns a similarity score that correlates with human judgement. Hence, the key aspect of having an accurate STS framework is the design of features that can adequately represent various aspects of the similarity between texts, e.g., using lexical, syntactic and semantic similarity metrics.

The majority of approaches treat input text pairs as feature vectors where each feature is a score corresponding to a certain type of similarity. This approach is conceptually easy to implement and the STS shared task at SemEval 2012 (Agirre et al., 2012) (STS-2012) has shown that the best systems were built following this idea, i.e., a number of features encoding similarity of an input text pair were combined in a single scoring model, e.g., SVR. Nevertheless, one limitation of using only similarity features to represent a text pair is that of low representation power.

The novelty of our approach is that we treat the input text pairs as structural objects and rely on the power of kernel learning to extract relevant structures. To link the documents in a pair we mark the

nodes in the related structures with a special relational tag. This way effective structural relational patterns are implicitly encoded in the trees and can be automatically learned by the kernel-based machines. We combine our relational structural model with the features from two best systems of STS-2012. Finally, we use the approach of classifier stacking to combine several structural models into the feature vector representation.

The contribution of this paper is as follows: (i) it provides a convincing evidence that adding structural features automatically extracted by structural kernels yields a significant improvement in accuracy; (ii) we define a combination kernel that integrates both structural and feature vector representations within a single scoring model, e.g., Support Vector Regression; (iii) we provide a simple way to construct relational structural models that can be built using off-the-shelf NLP tools; (iv) we experiment with four structural representations and show that constituency and dependency trees represent the best source for learning structural relationships; and (v) using a classifier stacking approach, structural models can be easily combined and integrated into existing feature-based STS models.

## 2 Structural Relational Similarity

The approach of relating pairs of input structures by learning predictable syntactic transformations has shown to deliver state-of-the-art results in question answering, recognizing textual entailment, and paraphrase detection, e.g. (Wang et al., 2007; Wang and Manning, 2010; Heilman and Smith, 2010). Previous work relied on fairly complex approaches, e.g. applying quasi-synchronous grammar formalism and variations of tree edit distance alignments, to extract syntactic patterns relating pairs of input structures. Our approach is conceptually simpler, as it regards the problem within the kernel learning framework, where we first encode salient syntactic/semantic proper-

ties of the input text pairs into tree structures and rely on tree kernels to automatically generate rich feature spaces. This work extends in several directions our earlier work in question answering, e.g., (Moschitti et al., 2007; Moschitti and Quarteroni, 2008), in textual entailment recognition, e.g., (Moschitti and Zanzotto, 2007), and more in general in relational text categorization (Moschitti, 2008; Severyn and Moschitti, 2012).

In this section we describe: (i) a kernel framework to combine structural and vector models; (ii) structural kernels to handle feature engineering; and (iii) suitable structural representations for relational learning.

## 2.1 Structural Kernel Learning

In supervised learning, given labeled data  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ , the goal is to estimate a decision function  $h(\mathbf{x}) = \mathbf{y}$  that maps input examples to their targets. A conventional approach is to represent a pair of texts as a set of similarity features  $\{f_i\}$ , s.t. the predictions are computed as  $h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} = \sum_i w_i f_i$ , where  $\mathbf{w}$  is the model weight vector. Hence, the learning problem boils down to estimating individual weights of each of the similarity features  $f_i$ . One downside of such approach is that a great deal of similarity information encoded in a given text pair is lost when modeled by single real-valued scores.

A more versatile approach in terms of the input representation relies on kernels. In a typical kernel learning approach, e.g., SVM, the prediction function for a test input  $\mathbf{x}$  takes on the following form  $h(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$ , where  $\alpha_i$  are the model parameters estimated from the training data,  $y_i$  are target variables,  $\mathbf{x}_i$  are support vectors, and  $K(\cdot, \cdot)$  is a kernel function.

To encode both structural representation and similarity feature vectors of a given text pair in a single model we define each document in a pair to be composed of a tree and a vector:  $\langle \mathbf{t}, \mathbf{v} \rangle$ . To compute a kernel between two text pairs  $\mathbf{x}_i$  and  $\mathbf{x}_j$  we define the following all-vs-all kernel, where all possible combinations of components,  $\mathbf{x}^{(1)}$  and  $\mathbf{x}^{(2)}$ , from each text pair are considered:  $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}) + K(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)}) + K(\mathbf{x}_i^{(2)}, \mathbf{x}_j^{(1)}) + K(\mathbf{x}_i^{(2)}, \mathbf{x}_j^{(2)})$ . Each of the kernel computations  $K$  can be broken down into the following:  $K(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = K_{\text{TK}}(\mathbf{t}^{(1)}, \mathbf{t}^{(2)}) + K_{\text{fvec}}(\mathbf{v}^{(1)}, \mathbf{v}^{(2)})$ , where  $K_{\text{TK}}$  computes a structural kernel and  $K_{\text{fvec}}$  is a kernel over feature vectors, e.g., linear, polynomial or RBF, etc. Further

in the text we refer to structural tree kernel models as TK and explicit feature vector representation as fvec.

Having defined a way to jointly model text pairs using structural TK representations along with the similarity features fvec, we next briefly review tree kernels and our relational structures.

## 2.2 Tree Kernels

We use tree structures as our base representation since they provide sufficient flexibility in representation and allow for easier feature extraction than, for example, graph structures. Hence, we rely on tree kernels to compute  $K_{\text{TK}}(\cdot, \cdot)$ . Given two trees it evaluates the number of substructures (or *fragments*) they have in common, i.e., it is a measure of their overlap. Different TK functions are characterized by alternative fragment definitions. In particular, we focus on the Syntactic Tree kernel (STK) (Collins and Duffy, 2002) and a Partial Tree Kernel (PTK) (Moschitti, 2006).

**STK** generates all possible substructures rooted in each node of the tree with the constraint that production rules can not be broken (i.e., any node in a tree fragment must include either all or none of its children).

**PTK** can be more effectively applied to both constituency and dependency parse trees. It generalizes STK as the fragments it generates can contain any subset of nodes, i.e., PTK allows for breaking the production rules and generating an extremely rich feature space, which results in higher generalization ability.

## 2.3 Structural representations

In this paper, we define simple-to-build relational structures based on: (i) a shallow syntactic tree, (ii) constituency, (iii) dependency and (iv) phrase-dependency trees.

**Shallow tree** is a two-level syntactic hierarchy built from word lemmas (leaves), part-of-speech tags (preterminals) that are further organized into chunks. It was shown to significantly outperform feature vector baselines for modeling relationships between question answer pairs (Severyn and Moschitti, 2012).

**Constituency tree.** While shallow syntactic parsing is very fast, here we consider using constituency structures as a potentially richer source of syntactic/semantic information.

**Dependency tree.** We propose to use dependency relations between words to derive an alternative structural representation. In particular, de-

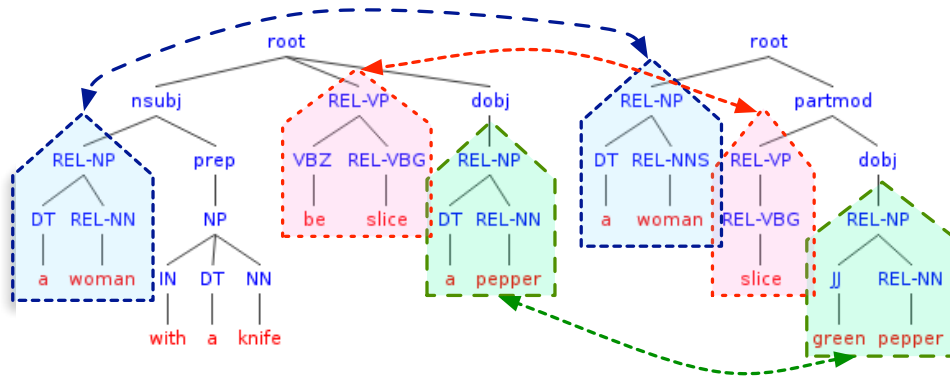


Figure 1: A phrase dependency-based structural representation of a text pair (s1, s2): *A woman with a knife is slicing a pepper* (s1) vs. *A women slicing green pepper* (s2) with a high semantic similarity (human judgement score 4.0 out of 5.0). Related tree fragments are linked with a REL tag.

pendency relations are used to link words in a way that they are always at the leaf level. This reordering of the nodes helps to avoid the situation where nodes with words tend to form long chains. This is essential for PTK to extract meaningful fragments. We also plug part-of-speech tags between the word nodes and nodes carrying their grammatical role.

**Phrase-dependency tree.** We explore a phrase-dependency tree similar to the one defined in (Wu et al., 2009). It represents an alternative structure derived from the dependency tree, where the dependency relations between words belonging to the same phrase (chunk) are collapsed in a unified node. Different from (Wu et al., 2009), the collapsed nodes are stored as a shallow subtree rooted at the unified node. This node organization is particularly suitable for PTK that effectively runs a sequence kernel on the tree fragments inside each chunk subtree. Fig 1 gives an example of our variation of a phrase dependency tree.

As a final consideration, if a document contains multiple sentences they are merged in a single tree with a common root. To encode the structural relationships between documents in a pair a special REL tag is used to link the related structures. We adopt a simple strategy to establish such links: words from two documents that have a common lemma get their parents (POS tags) and grandparents, non-terminals, marked with a REL tag.

### 3 Pairwise similarity features.

Along with the direct representation of input text pairs as structural objects our framework is also capable of encoding pairwise similarity feature vectors ( $f_{vec}$ ), which we describe below.

**Baseline features.** (base) We adopt similarity features from two best performing systems of STS-2012, which were publicly released<sup>1</sup>: namely, the Takelab<sup>2</sup> system (Šarić et al., 2012) and the UKP Lab’s system<sup>3</sup> (Bar et al., 2012). Both systems represent input texts with similarity features combining multiple text similarity measures of varying complexity.

UKP (U) provides metrics based on matching of character, word n-grams and common subsequences. It also includes features derived from Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) and aggregation of word similarity based on lexical-semantic resources, e.g., WordNet. In total it provides 18 features.

Takelab (T) includes n-gram matching of varying size, weighted word matching, length difference, WordNet similarity and vector space similarity where pairs of input sentences are mapped into Latent Semantic Analysis (LSA) space. The features are computed over several sentence representations where stop words are removed and/or lemmas are used in place of raw tokens. The total number of Takelab’s features is 21. The combined system consists of 39 features.

**Additional features.** We also augment the U and T feature sets, with an additional set of features (A) which includes: a cosine similarity scores computed over (i) n-grams of part-of-speech tags (up to 4-grams), (ii) SuperSense tags (Ciaramita and

<sup>1</sup>Note that only a subset of the features used in the final evaluation was released, which results in lower accuracy when compared to the official rankings.

<sup>2</sup><http://takelab.fer.hr/sts/>

<sup>3</sup><https://code.google.com/p/dkpro-similarity-asl/wiki/SemEval2013>

Altun, 2006), (iii) named entities, (iv) dependency triplets, and (v) PTK syntactic similarity scores computed between documents in a pair, where as input representations we use raw *dependency* and *constituency* trees. To alleviate the problem of domain adaptation, where datasets used for training and testing are drawn from different sources, we include additional features to represent the combined text of a pair: (i) bags (B) of lemmas, dependency triplets, production rules (from the constituency parse tree) and a normalized length of the entire pair; and (ii) a manually encoded corpus type (M), where we use a binary feature with a non-zero entry corresponding to a dataset type. This helps the learning algorithm to learn implicitly the individual properties of each dataset.

**Stacking.** To integrate multiple TK representations into a single model we apply a classifier stacking approach (Fast and Jensen, 2008). Each of the learned TK models is used to generate predictions which are then plugged as features into the final  $f_{vec}$  representation, s.t. the final model uses only explicit feature vector representation. To obtain prediction scores, we apply 5-fold cross-validation scheme, s.t. for each of the held-out folds we obtain independent predictions.

## 4 Experiments

We present the results of our model tested on the data from the Core STS task at SemEval 2012.

### 4.1 Setup

**Data.** To compare with the best systems of the STS-2012 we followed the same setup used in the final evaluation, where 3 datasets (*MSRpar*, *MSRvid* and *SMTeuroparl*) are used for training and 5 for testing (two “surprise” datasets were added: *OnWN* and *SMTnews*). We use the entire training data to obtain a single model for making predictions on each test set.

**Software.** To encode TK models along with the similarity feature vectors into a single regression scoring model, we use an SVR framework implemented in SVM-Light-TK<sup>4</sup>. We use the following parameter settings  $-t\ 5\ -F\ 1\ -W\ A\ -C\ +$ , which specifies a combination of trees and feature vectors ( $-C\ +$ ), STK over trees ( $-F\ 1$ ) ( $-F\ 3$  for PTK) computed in all-vs-all mode ( $-W\ A$ ) and polynomial kernel of degree 3 for the feature vector (active by default).

<sup>4</sup><http://disi.unitn.it/moschitti/Tree-Kernel.htm>

**Metrics.** We report the following metrics employed in the final evaluation: *Pearson* correlation for individual test sets<sup>5</sup> and *Mean* – an average score weighted by the test set size.

### 4.2 Results

Table 1 summarizes the results of combining TK models with a strong feature vector model. We test structures defined in Sec. 2.3 when using STK and PTK. The results show that: (i) combining all three features sets (U, T, A) provides a strong baseline system that we attempt to further improve with our relational structures; (ii) the generality of PTK provides an advantage over STK for learning more versatile models; (iii) constituency and dependency representations seem to perform better than shallow and phrase-dependency trees; (iv) using structures with no relational linking does not work; (v) TK models provide a far superior source of structural similarity than  $U + T + A$  that already includes PTK similarity scores as features, and finally (vi) the domain adaptation problem can be addressed by including corpus specific features, which leads to a large improvement over the previous best system.

## 5 Conclusions and Future Work

We have presented an approach where text pairs are directly treated as structural objects. This provides a much richer representation for the learning algorithm to extract useful syntactic and shallow semantic patterns. We have provided an extensive experimental study of four different structural representations, e.g. shallow, constituency, dependency and phrase-dependency trees using STK and PTK. The novelty of our approach is that it goes beyond a simple combination of tree kernels with feature vectors as: (i) it directly encodes input text pairs into relationally linked structures; (ii) the learned structural models are used to obtain prediction scores thus making it easy to plug into existing feature-based models, e.g. via stacking; (iii) to our knowledge, this work is the first to apply structural kernels and combinations in a regression setting; and (iv) our model achieves the state of the art in STS largely improving the best previous systems. Our structural learning approach to STS is conceptually simple and does not require additional linguistic sources other than off-the-shelf syntactic parsers. It is particularly suitable for NLP tasks where the input domain comes

<sup>5</sup>we also report the results for a concatenation of all five test sets (ALL)

| Experiment                          | U | T | A | S | C | D | P | STK | PTK | B | M | ALL   | Mean  | MSRp  | MSRv  | SMTe  | OnWN  | SMTn  |  |  |
|-------------------------------------|---|---|---|---|---|---|---|-----|-----|---|---|-------|-------|-------|-------|-------|-------|-------|--|--|
| fvec<br>model                       | • |   |   |   |   |   |   |     |     |   |   | .7060 | .6087 | .6080 | .8390 | .2540 | .6820 | .4470 |  |  |
|                                     |   | • |   |   |   |   |   |     |     |   |   | .7589 | .6863 | .6814 | .8637 | .4950 | .7091 | .5395 |  |  |
|                                     | • | • |   |   |   |   |   |     |     |   |   | .8079 | .7161 | .7134 | .8837 | .5519 | .7343 | .5607 |  |  |
|                                     | • | • | • |   |   |   |   |     |     |   |   | .8187 | .7137 | .7157 | .8833 | .5131 | .7355 | .5809 |  |  |
| TK<br>models<br>with STK<br>and PTK | • | • | • | • |   |   |   | •   |     |   |   | .8261 | .6982 | .7026 | .8870 | .4807 | .7258 | .5333 |  |  |
|                                     | • | • | • |   | • |   |   | •   |     |   |   | .8326 | .6970 | .7020 | .8925 | .4826 | .7190 | .5253 |  |  |
|                                     | • | • | • |   |   | • |   | •   |     |   |   | .8341 | .7024 | .7086 | .8921 | .4671 | .7319 | .5495 |  |  |
|                                     | • | • | • |   |   |   | • | •   |     |   |   | .8211 | .6693 | .6994 | .8903 | .2980 | .7035 | .5603 |  |  |
|                                     | • | • | • | • |   |   |   |     | •   |   |   | .8362 | .7026 | .6927 | .8896 | .5282 | .7144 | .5485 |  |  |
|                                     | • | • | • |   | • |   |   |     | •   |   |   | .8458 | .7047 | .6935 | .8953 | .5080 | .7101 | .5834 |  |  |
| REL tag                             | • | • | • |   |   | ◦ |   |     |     |   |   | .8218 | .6899 | .6644 | .8726 | .4846 | .7228 | .5684 |  |  |
|                                     | • | • | • |   |   |   | ◦ |     |     |   |   | .8250 | .7000 | .6806 | .8822 | .5171 | .7145 | .5769 |  |  |
|                                     | • | • | • |   | • |   |   |     |     | • |   | .8539 | .7132 | .6993 | .9005 | .4772 | .7189 | .6481 |  |  |
|                                     | • | • | • |   |   | • |   |     |     | • |   | .8529 | .7249 | .7080 | .8984 | .5142 | .7263 | .6700 |  |  |
| domain<br>adaptation                | • | • | • |   | • |   |   |     |     | • |   | .8546 | .7156 | .6989 | .8979 | .4884 | .7181 | .6609 |  |  |
|                                     | • | • | • |   | • |   |   |     |     | • |   | .8810 | .7416 | .7210 | .8971 | .5912 | .7328 | .6778 |  |  |
|                                     | • | • | • |   | • | • |   |     |     | • | • | .8239 | .6773 | .6830 | .8739 | .5280 | .6641 | .4937 |  |  |
| UKP (best system of STS-2012)       |   |   |   |   |   |   |   |     |     |   |   | .8239 | .6773 | .6830 | .8739 | .5280 | .6641 | .4937 |  |  |

Table 1: Results on STS-2012. First set of experiments studies the combination of fvec models from UKP (U), Takelab (T) and (A). Next we show results for four structural representations: shallow (S), constituency (C), dependency (D) and phrase-dependency (P) trees with STK and PTK; next row set demonstrates the necessity of relational linking for two best structures, i.e. C and D (empty circle denotes a structures with no relational linking.); finally, domain adaptation via bags of features (B) of the entire pair and (M) manually encoded dataset type show the state of the art results.

as pairs of objects, e.g., question answering, paraphrasing and recognizing textual entailment.

## 6 Acknowledgements

This research is supported by the EU’s Seventh Framework Program (FP7/2007-2013) under the #288024 LIMOSINE project.

## References

- Eneko Agirre, Daniel Cer, Mona Diab, and Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *\*SEM*.
- Daniel Bar, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *SemEval*.
- Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *EMNLP*.
- Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *ACL*.
- Andrew S. Fast and David Jensen. 2008. Why stacked models perform effective collective classification. In *ICDM*.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*.
- Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL*.
- Alessandro Moschitti and Silvia Quarteroni. 2008. Kernels on linguistic structures for answer extraction. In *ACL*.
- Alessandro Moschitti and Fabio Massimo Zanzotto. 2007. Fast and effective kernels for relational learning from texts. In *ICML*.
- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL*.
- Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*.
- Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*.
- Aliaksei Severyn and Alessandro Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. In *SIGIR*.
- Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *SemEval*.
- Mengqiu Wang and Christopher D. Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *ACL*.
- Mengqiu Wang, Noah A. Smith, and Teruko Mitaura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP*.
- Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*.

# Typesetting for Improved Readability using Lexical and Syntactic Information

Ahmed Salama

Kemal Oflazer

Susan Hagan

Carnegie Mellon University – Qatar

Doha, Qatar

ahmedsaa@qatar.cmu.edu ko@cs.cmu.edu

hagan@cmu.edu

## Abstract

We present results from our study of which uses syntactically and semantically motivated information to group segments of sentences into *unbreakable units* for the purpose of typesetting those sentences in a region of a fixed width, using an otherwise standard dynamic programming line breaking algorithm, to minimize raggedness. In addition to a rule-based baseline segmenter, we use a very modest size text, manually annotated with positions of breaks, to train a maximum entropy classifier, relying on an extensive set of lexical and syntactic features, which can then predict whether or not to break after a certain word position in a sentence. We also use a simple genetic algorithm to search for a subset of the features optimizing  $F_1$ , to arrive at a set of features that delivers 89.2% Precision, 90.2% Recall (89.7%  $F_1$ ) on a test set, improving the rule-based baseline by about 11 points and the classifier trained on all features by about 1 point in  $F_1$ .

## 1 Introduction and Motivation

Current best practice in typography focuses on several interrelated factors (Humar et al., 2008; Tinkel, 1996). These factors include typeface selection, the color of the type and its contrast with the background, the size of the type, the length of the lines of type in the body of the text, the media in which the type will live, the distance between each line of type, and the appearance of the justified or ragged right side edge of the paragraphs, which should maintain either the appearance of a straight line on both sides of the block of type (justified) or create a gentle wave on the ragged right side edge.

This paper addresses one aspect of current “best practice,” concerning the alignment of text in a paragraph. While current practice values that gentle “wave,” which puts the focus on the elegant look of the overall paragraph, it does so at the expense of meaning-making features. Meaning-making features enable typesetting to maintain the integrity of phrases within sentences, giving those interests equal consideration with the overall look of the paragraph. Figure 1 (a) shows a text fragment typeset without any regard to natural breaks while (b) shows an example of a typesetting that we would like to get, where many natural breaks are respected.

While current practice works well enough for native speakers, fluency problems for non-native speakers lead to uncertainty when the beginning and end of English phrases are interrupted by the need to move to the next line of the text before completing the phrase. This pause is a potential problem for readers because they try to interpret content words, relate them to their referents and anticipate the role of the next word, as they encounter them in the text (Just and Carpenter, 1980). While incorrect anticipation might not be problematic for native speakers, who can quickly re-adjust, non-native speakers may find inaccurate anticipation more troublesome. This problem could be more significant because English as a second language (ESL) readers are engaged not only in understanding a foreign language, but also in processing the “anticipated text” as they read a partial phrase, and move to the next line in the text, only to discover that they anticipated meaning incorrectly. Even native speakers with less skill may experience difficulty comprehending text and work with young readers suggests that “[c]omprehension difficulties may be localized at points of high processing demands whether from syntax or other sources” (Perfetti et al., 2005). As ESL readers process a partial phrase, and move to

the next line in the text, instances of incorrectly anticipated meaning would logically increase processing demands to a greater degree. Additionally, as readers make meaning, we assume that they don't parse their thoughts using the same phrasal divisions "needed to diagram a sentence." Our perspective not only relies on the immediacy assumption, but also develops as an outgrowth of other ways that we make meaning outside of the form or function rules of grammar. Specifically, Halliday and Hasan (1976) found that rules of grammar do not explain how cohesive principals engage readers in meaning making across sentences. In order to make meaning across sentences, readers must be able to refer anaphorically backward to the previous sentence, and cataphorically forward to the next sentence. Along similar lines, readers of a single sentence assume that transitive verbs will include a direct object, and will therefore speculate about what that object might be, and sometimes get it wrong.

Thus proper typesetting of a segment of text must explore ways to help readers avoid incorrect anticipation, while also considering those moments in the text where readers tend to pause in order to integrate the meaning of a phrase. Those decisions depend on the context. A phrasal break between a one-word subject and its verb tends to be more unattractive, because the reader does not have to make sense of relationships between the noun/subject and related adjectives before moving on to the verb. In this case, the reader will be more likely to anticipate the verb to come. However, a break between a subject preceded by multiple adjectives and its verb is likely to be more useful to a reader (if not ideal), because the relationships between the noun and its related adjectives are more likely to have thematic importance leading to longer gaze time on the relevant words in the subject phrase (Just and Carpenter, 1980).

We are not aware of any prior work for bringing computational linguistic techniques to bear on this problem. A relatively recent study (Levasseur et al., 2006) that accounted only for breaks at commas and ends of sentences, found that even those breaks improved reading fluency. While the participants in that study were younger (7 to 9+ years old), the study is relevant because the challenges those young participants face, are faced again when readers of any age encounter new and complicated texts that present words they do not

know, and ideas they have never considered.

On the other hand, there is ample work on the basic algorithm to place a sequence of words in a typesetting area with a certain width, commonly known as the *optimal line breaking problem* (e.g., Plass (1981), Knuth and Plass (1981)). This problem is quite well-understood and basic variants are usually studied as an elementary example application of dynamic programming.

In this paper we explore the problem of learning where to break sentences in order to avoid the problems discussed above. Once such unbreakable segments are identified, a simple application of the dynamic programming algorithm for optimal line breaking, using unbreakable segments as "words", easily typesets the text to a given width area.

## 2 Text Breaks

The rationale for content breaks is linked to our interest in preventing inaccurate anticipation, which is based on the immediacy assumption. The immediacy assumption (Just and Carpenter, 1980) considers, among other things, the reader's interest in trying to relate content words to their referents as soon as possible. Prior context also encourages the reader to anticipate a particular role or case for the next word, such as agent or the manner in which something is done. Therefore, in defining our breaks, we consider not only the need to maintain the syntactic integrity of phrases, such as the prepositional phrase, but also the semantic integrity across syntactical divisions. For example, semantic integrity is important when transitive verbs anticipate direct objects. Strictly speaking, we define a bad break as one that will cause (i) unintended anaphoric collocation, (ii) unintended cataphoric collocation, or (iii) incorrect anticipation.

Using these broad constraints, we derived a set of about 30 rules that define acceptable and non-acceptable breaks, with exceptions based on context and other special cases. Some of the rules are very simple and are only related to the word position in the sentence:

- Break at the end of a sentence.
- Keep the first and last words of a sentence with the rest of it.

The rest of the rule set are more complex and depend on the structure of the sentence in question,



sanctions and UN charges of gross rights abuses. Military tensions on the Korean peninsula have risen to their highest level for years, with the communist state under the youthful Kim threatening nuclear war in response to UN sanctions imposed after its third atomic test last month. It has also

(a) Text with standard typesetting

from US sanctions and UN charges of gross rights abuses. Military tensions on the Korean peninsula have risen to their highest level for years, with the communist state under the youthful Kim threatening nuclear war in response to UN sanctions imposed after its third atomic test last month.

(b) Text with syntax-directed typesetting

Figure 1: Short fragment of text with standard typesetting (a) and with syntax and semantics motivated typesetting (b), both in a 75 character width.

e.g.:

- Keep a single word subject with the verb.
  - Keep an appositive phrase with the noun it renames.
  - Do not break inside a prepositional phrase.
  - Keep marooned prepositions with the word they modify.
  - Keep the verb, the object and the preposition together in a phrasal verb phrase.
  - Keep a gerund clause with its adverbial complement.
- *Precision*: Percentage of the breaks posited that were actually correct breaks in the gold-standard hand-annotated data. It is possible to get 100% precision by putting a single break at the end.
  - *Recall*: Percentage of the actual breaks correctly posited. It is possible to get 100% recall by positing a break after each token.
  - $F_1$ : The geometric mean of precision and recall divided by their average.

It should be noted that when a text is typeset into an area of width of a certain number of characters, an erroneous break need not necessarily lead to an actual break in the final output, that is an error may not be too bad. On the other hand, a missed break while not hurting the readability of the text may actually lead to a long segment that may eventually worsen raggedness in the final typesetting.

There are exceptions to these rules in certain cases such as overly long phrases.

### 3 Experimental Setup

Our data set consists of a modest set of 150 sentences (3918 tokens) selected from four different documents and *manually* annotated by a human expert relying on the 30 or so rules. The annotation consists of marking after each token whether one is allowed to break at that position or not.<sup>1</sup>

We developed three systems for predicting breaks: a rule-based baseline system, a maximum-entropy classifier that learns to classify breaks using about 100 lexical, syntactic and collocational features, and a maximum entropy classifier that uses a subset of these features selected by a simple genetic algorithm in a hill-climbing fashion. We evaluated our classifiers *intrinsically* using the usual measures:

**Baseline Classifier** We implemented a subset of the rules (those that rely only on lexical and part-of-speech information), as a baseline rule-based break classifier. The baseline classifier avoids breaks:

- after the first word in a sentence, quote or parentheses,
- before the last word in a sentence, quote or parentheses, and
- between a punctuation mark following a word or between two consecutive punctuation marks.

It posits breaks (i) before a word following a punctuation, and (ii) before prepositions, auxiliary verbs, coordinating conjunctions, subordinate conjunctions, relative pronouns, relative adverbs, conjunctive adverbs, and correlative conjunctions.

<sup>1</sup>We expect to make our annotated data available upon the publication of the paper.

**Maximum Entropy Classifier** We used the *CRF++ Tool*<sup>2</sup> but with the option to run it only as a maximum entropy classifier (Berger et al., 1996), to train a classifier. We used a large set of about 100 features grouped into the following categories:

- *Lexical features*: These features include the token and the POS tag for the previous, current and the next word. We also encode whether the word is part of a compound noun or a verb, or is an adjective that subcategorizes a specific preposition in WordNet, (e.g., *familiar with*).
- *Constituency structure features*: These are unlexicalized features that take into account in the parse tree, for a word and its previous and next words, the labels of the parent, the grandparent and their siblings, and number of siblings they have. We also consider the label of the closest common ancestor for a word and its next word.
- *Dependency structure features*: These are unlexicalized features that essentially capture the number of dependency relation links that cross-over a given word boundary. The motivation for these comes from the desire to limit the amount of information that would need to be carried over that boundary, assuming this would be captured by the number of dependency links over the break point.
- *Baseline feature*: This feature reflects whether the rule-based baseline break classifier posits a break at this point or not.

We use the following tools to process the sentences to extract some of these features:

- Stanford constituency and dependency parsers, (De Marneffe et al., 2006; Klein and Manning, 2002; Klein and Manning, 2003),
- lemmatization tool in NLTK (Bird, 2006),
- WordNet for compound nouns and verbs (Fellbaum, 1998).

<sup>2</sup>Available at <http://crfpp.googlecode.com/svn/trunk/doc/index.html>.

|                      | Baseline | ME-All | ME-GA |
|----------------------|----------|--------|-------|
| <b>Precision</b>     | 77.9     | 87.3   | 89.2  |
| <b>Recall</b>        | 80.4     | 90.2   | 90.2  |
| <b>F<sub>1</sub></b> | 79.1     | 88.8   | 89.7  |

Table 1: Results from Baseline and Maximum Entropy break classifiers

**Maximum Entropy Classifier with GA Feature Selection** We used a genetic algorithm on a development data set, to select a subset of the features above. Basically, we start with a randomly selected set of features and through mutation and crossover try to obtain feature combinations that perform better over the development set in terms of  $F_1$  score. After a few hundred generations of this kind of hill-climbing, we get a subset of features that perform the best.

## 4 Results

Our current evaluation is only intrinsic in that we measure our performance in getting the break and no-break points correctly in a test set. The results are shown in Table 1. The column ME-All shows the results for a maximum entropy classifier using all the features and the column ME-GA shows the results for a maximum entropy classifier using about 50 of the about 100 features available, as selected by the genetic algorithm.

Our best system delivers 89.2% precision and 90.2% recall (with 89.7%  $F_1$ ), improving the rule-based baseline by about 11 points and the classifier trained on all features by about 1 point in  $F_1$ .

After processing our test set with the ME-GA classifier, we can feed the segments into a standard word-wrapping dynamic programming algorithm (along with a maximum width) and obtain a typeset version with minimum raggedness on the right margin. This algorithm is fast enough to use even dynamically when resizing a window if the text is displayed in a browser on a screen. Figure 1 (b) displays an example of a small fragment of text typeset using the output of our best break classifier. One can immediately note that this typesetting has more raggedness overall, but avoids the bad breaks in (a). We are currently in the process of designing a series of experiments for extrinsic evaluation to determine if such typeset text helps comprehension for secondary language learners.

## 4.1 Error Analysis

An analysis of the errors our best classifier makes (which may or may not be translated into an actual error in the final typesetting) shows that the majority of the errors basically can be categorized into the following groups:

- Incorrect breaks posited for multiword collocations (e.g., *act\* of war*,<sup>3</sup> *rule\* of law*, *far ahead\* of*, *raining cats\* and dogs*, etc.)
- Missed breaks after a verb (e.g., *calls | an act of war*, *proceeded to | implement*, etc.)
- Missed breaks before or after prepositions or adverbials (e.g., *the day after | the world realized*, *every kind | of interference*)

We expect to overcome such cases by increasing our training data size significantly by using our classifier to break new texts and then have a human annotator to manually correct the breaks.

## 5 Conclusions and Future Work

We have used syntactically motivated information to help in typesetting text to facilitate better understanding of English text especially by secondary language learners, by avoiding breaks which may cause unnecessary anticipation errors. We have cast this as a classification problem to indicate whether to break after a certain word or not, by taking into account a variety of features. Our best system maximum entropy framework uses about 50 such features, which were selected using a genetic algorithm and performs significantly better than a rule-based break classifier and better than a maximum entropy classifier that uses all available features.

We are currently working on extending this work in two main directions: We are designing a set of experiments to *extrinsically* test whether typesetting by our system improves reading ease and comprehension. We are also looking into a break labeling scheme that is not binary but based on a notion of “badness” – perhaps quantized into 3-4 grades, that would allow flexibility between preventing bad breaks and minimizing raggedness. For instance, breaking a noun-phrase right after an initial `the` may be considered very bad. On the other hand, although it is desirable to keep an object NP together with the preceding transitive verb,

<sup>3</sup>\* indicates a spurious incorrect break, | indicates a missed break.

breaking before the object NP, could be OK, if not doing so causes an inordinate amount of raggedness. Then the final typesetting stage can optimize a combination of raggedness and the total “badness” of all the breaks it posits.

## Acknowledgements

This publication was made possible by grant NPRP-09-873-1-129 from the Qatar National Research Fund (a member of the Qatar Foundation). Susan Hagan acknowledges the generous support of the Qatar Foundation through Carnegie Mellon University’s Seed Research program. The statements made herein are solely the responsibility of this author(s), and not necessarily those of the Qatar Foundation.

## References

- Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.
- Steven Bird. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL*, pages 69–72. Association for Computational Linguistics.
- Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.
- Christiane Fellbaum. 1998. WordNet: An electronic lexical database. *The MIT Press*.
- M. A. K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman, London.
- I. Humar, M. Gradisar, and T. Turk. 2008. The impact of color combinations on the legibility of a web page text presented on crt displays. *International Journal of Industrial Ergonomics*, 38(11-12):885–899.
- Marcel A. Just and Patricia A. Carpenter. 1980. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87:329–354.
- 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.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

- Donald E Knuth and Michael F. Plass. 1981. Breaking paragraphs into lines. *Software: Practice and Experience*, 11(11):1119–1184.
- Valerie Marciarille Levasseur, Paul Macaruso, Laura Conway Palumbo, and Donald Shankweiler. 2006. Syntactically cued text facilitates oral reading fluency in developing readers. *Applied Psycholinguistics*, 27(3):423–445.
- C. A. Perfetti, N. Landi, and J. Oakhill. 2005. The acquisition of reading comprehension skill. In M. J. Snowling and C. Hulme, editors, *The science of reading: A handbook*, pages 227–247. Blackwell, Oxford.
- Michael Frederick Plass. 1981. *Optimal Pagination Techniques for Automatic Typesetting Systems*. Ph.D. thesis, Stanford University.
- K. Tinkel. 1996. Taking it in: What makes type easier to read. *Adobe Magazine*, pages 40–50.

# Annotation of regular polysemy and underspecification

Héctor Martínez Alonso,  
Bolette Sandford Pedersen  
University of Copenhagen  
Copenhagen (Denmark)

alonso@hum.ku.dk, bsp@hum.ku.dk

Núria Bel  
Universitat Pompeu Fabra  
Barcelona (Spain)  
nuria.bel@upf.edu

## Abstract

We present the result of an annotation task on regular polysemy for a series of semantic classes or *dot types* in English, Danish and Spanish. This article describes the annotation process, the results in terms of inter-encoder agreement, and the sense distributions obtained with two methods: majority voting with a theory-compliant backoff strategy, and MACE, an unsupervised system to choose the most likely sense from all the annotations.

## 1 Introduction

This article shows the annotation task of a corpus in English, Danish and Spanish for regular polysemy. Regular polysemy (Apresjan, 1974; Pustejovsky, 1995; Briscoe et al., 1995; Nunberg, 1995) has received a lot of attention in computational linguistics (Boleda et al., 2012; Rumshisky et al., 2007; Shutova, 2009). The lack of available sense-annotated gold standards with underspecification is a limitation for NLP applications that rely on dot types<sup>1</sup> (Rumshisky et al., 2007; Poibeau, 2006; Pustejovsky et al., 2009).

Our goal is to obtain human-annotated corpus data to study regular polysemy and to detect it in an automatic manner. We have collected a corpus of annotated examples in English, Danish and Spanish to study the alternation between senses and the cases of underspecification, including a contrastive study between languages. Here we describe the annotation process, its results in terms of inter-encoder agreement, and the sense distributions obtained with two methods: majority voting with a theory-compliant backoff strategy and, MACE an unsupervised system to choose the most likely sense from all the annotations.

<sup>1</sup>The corpus is freely available at <http://metashare.cst.dk/repository/search/?q=regular+polysemy>

## 2 Regular polysemy

Very often a word that belongs to a semantic type, like Location, can behave as a member of another semantic type, like Organization, as shown by the following examples from the American National Corpus (Ide and Macleod, 2001) (ANC):

- a) *Manuel died in exile in 1932 in England.*
- b) *England was being kept busy with other concerns*
- c) *England was, after all, an important wine market*

In case a), *England* refers to the English territory (Location), whereas in b) it refers arguably to England as a political entity (Organization). The third case refers to both. The ability of certain words to switch between semantic types in a predictable manner is referred to as *regular polysemy*. Unlike other forms of meaning variation caused by metaphor or homonymy, regular polysemy is considered to be caused by metonymy (Apresjan, 1974; Lapata and Lascarides, 2003). Regular polysemy is different from other forms of polysemy in that both senses can be active at the same in a predicate, which we refer to as *underspecification*. Underspecified instances can be broken down in:

1. Contextually complex: *England was, after all, an important wine market*
2. Zeugmatic, in which two mutually exclusive readings are coordinated: *England is conservative and rainy*
3. Vague, in which no contextual element enforces a reading: *The case of England is similar*

## 3 Choice of semantic classes

The Generative Lexicon (GL) (Pustejovsky, 1995) groups nouns with their most frequent metonymic sense in a semantic class called a *dot type*. For English, we annotate 5 dot types from the GL:

1. **Animal/Meat:** *"The chicken ran away"* vs.

"the chicken was delicious".

2. **Artifact/Information** : "The book fell" vs. "the book was boring".
3. **Container/Content**: "The box was red" vs. "I hate the whole box".
4. **Location/Organization**: "England is far" vs. "England starts a tax reform".
5. **Process/Result**: "The building took months to finish" vs. "the building is sturdy".

For Danish and Spanish, we have chosen Container/Content and Location/Organization. We chose the first one because we consider it the most prototypical case of metonymy from the ones listed in the GL. We chose the second one because the metonymies in locations are a common concern for Named-Entity Recognition (Johannessen et al., 2005) and a previous area of research in metonymy resolution (Markert and Nissim, 2009).

#### 4 Annotation Scheme

For each of the nine (five for English, two for Danish, two for Spanish) dot types, we have randomly selected 500 corpus examples. Each example consists of a sentence with a selected *headword* belonging to the corresponding dot type. In spite of a part of the annotation being made with a contrastive study in mind, no parallel text was used to avoid using translated text. For English and Danish we used freely available reference corpora (Ide and Macleod, 2001; Andersen et al., 2002) and, for Spanish, a corpus built from newswire and technical text (Vivaldi, 2009).

For most of the English examples we used the words in Rumshisky (2007), except for Location/Organization. For Danish and Spanish we translated the words from English. We expanded the lists using each language's wordnet (Pedersen et al., 2009; Gonzalez-Agirre et al., 2012) as thesaurus to make the total of occurrences reach 500 after we had removed homonyms and other forms of semantic variation outside of the purview of regular polysemy.

For Location/Organization we have used high-frequency names of geopolitical locations from each of the corpora. Many of them are corpus-specific (e.g. *Madrid* is more frequent in the Spanish corpus) but a set of words is shared: *Afghanistan, Africa, America, China, England, Europe, Germany, London*.

Every dot type has its particularities that we had to deal with. For instance, English has lexical al-

ternatives for the meat of several common animals, like *venison* or *pork* instead of *deer* and *pig*. This lexical phenomenon does not impede metonymy for the animal names, it just makes it less likely. In order to assess this, we have included 20 examples of *cow*. The rest of the dataset consists of animal names that do not participate in this lexical alternation, like *eel, duck, chicken, or sardine*.

We call the first sense in the pair of metonyms that make up the dot type the *literal* sense, and the second sense the *metonymic* sense, e.g. Location is the literal sense in Location/Organization.

Each block of 500 sentences belonging to a dot type was an independent annotation subtask with an isolated description. The annotator was shown an example and had to determine whether the headword in the example had the literal, metonymic or the underspecified sense. Figure 1 shows an instance of the annotation process.

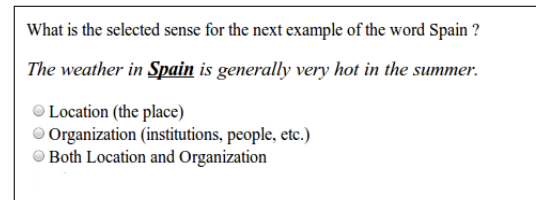


Figure 1: Screen capture for a Mechanical Turk annotation instance or HIT

This annotation scheme is designed with the intention of capturing literal, metonymic and underspecified senses, and we use an inventory of three possible answers, instead of using Markert and Nissim's (Markert and Nissim, 2002; Nissim and Markert, 2005) approach with fine-grained sense distinctions, which are potentially more difficult to annotate and resolve automatically. Markert and Nissim acknowledge a *mixed* sense they define as being literal and metonymic at the same time.

For English we used Amazon Mechanical Turk (AMT) with five annotations per example by turkers certified as Classification Masters. Using AMT provides annotations very quickly, possibly at the expense of reliability, but it has been proven suitable for sense-disambiguation task (Snow et al., 2008). Moreover, it is not possible to obtain annotations for every language using AMT. Thus, for Danish and Spanish, we obtained annotations from volunteers, most of them native or very proficient non-natives. See Table 1 for a summary of the annotation setup for each language.

After the annotation task we obtained the agree-

| Language | annotators | type      |
|----------|------------|-----------|
| Danish   | 3-4        | volunteer |
| English  | 5          | AMT       |
| Spanish  | 6-7        | volunteer |

Table 1: Amount and type of annotators per instance for each language.

ment values shown in Table 2. The table also provides the abbreviated names of the datasets.

| Dot type     | $\overline{A}_o \pm \sigma$ | $\alpha$ |
|--------------|-----------------------------|----------|
| eng:animeat  | $0.86 \pm 0.24$             | 0.69     |
| eng:artinfo  | $0.48 \pm 0.23$             | 0.12     |
| eng:contcont | $0.65 \pm 0.28$             | 0.31     |
| eng:locorg   | $0.72 \pm 0.29$             | 0.46     |
| eng:procrs   | $0.5 \pm 0.24$              | 0.10     |
| da:contcont  | $0.32 \pm 0.37$             | 0.39     |
| da:locorg    | $0.73 \pm 0.37$             | 0.47     |
| spa:contcont | $0.36 \pm 0.3$              | 0.42     |
| spa:locorg   | $0.52 \pm 0.28$             | 0.53     |

Table 2: Averaged observed agreement and its standard deviation and alpha

Average observed agreement ( $\overline{A}_o$ ) is the mean across examples for the proportion of matching senses assigned by the annotators. Krippendorff’s alpha is an aggregate measure that takes chance disagreement in consideration and accounts for the replicability of an annotation scheme. There are large differences in  $\alpha$  across datasets.

The scheme can only provide *reliable* (Artstein and Poesio, 2008) annotations ( $\alpha > 0.6$ ) for one dot type<sup>2</sup>. This indicates that not all dot types are equally easy to annotate, regardless of the kind of annotator. In spite of the number and type of annotators, the Location/Organization dot type gives fairly high agreement values for a semantic task, and this behavior is consistent across languages.

## 5 Assigning sense by majority voting

Each example has more than one annotation and we need to determine a single sense tag for each example. However, if we assign senses by majority voting, we need a backoff strategy in case of ties.

The common practice of backing off to the most frequent sense is not valid in this scenario, where there can be a tie between the metonymic and the underspecified sense. We use a backoff that incorporates our assumption about the relations

<sup>2</sup>We have made the data freely available at <http://metashare.cst.dk/repository/search/?q=regular+polysemy>

between senses, namely that the underspecified sense sits between the literal and the metonymic senses:

1. If there is a tie between the underspecified and literal senses, the sense is **literal**.
2. If there is a tie between the underspecified and metonymic sense, the sense is **metonymic**.
3. If there is a tie between the literal and metonymic sense or between all three senses, the sense is **underspecified**.

| Dot type     | L   | M   | U  | V  | B  |
|--------------|-----|-----|----|----|----|
| eng:animeat  | 358 | 135 | 7  | 3  | 4  |
| eng:artinfo  | 141 | 305 | 54 | 8  | 48 |
| eng:contcont | 354 | 120 | 25 | 0  | 25 |
| eng:locorg   | 307 | 171 | 22 | 3  | 19 |
| eng:procrs   | 153 | 298 | 48 | 3  | 45 |
| da:contcont  | 328 | 82  | 91 | 53 | 38 |
| da:locorg    | 322 | 95  | 83 | 44 | 39 |
| spa:contcont | 291 | 140 | 69 | 54 | 15 |
| soa:locorg   | 314 | 139 | 47 | 40 | 7  |

Table 3: Literal, Metonymic and Underspecified sense distributions, and underspecified senses broken down in Voting and Backoff

The columns labelled L, M and U in Table 3 provide the sense distributions for each dot type. The preference for the underspecified sense varies greatly, from the very infrequent for English in Animal/Meat to the two Danish datasets where the underspecified sense evens with the metonymic one. However, the Danish examples have mostly three annotators, and chance disagreement is the highest for this language in this setup, i.e., the chance for an underspecified sense in Danish to be assigned by our backoff strategy is the highest.

Columns V and B show respectively whether the underspecified senses are a result of majority voting or backoff. In contrast to volunteers, turkers disprefer the underspecified option and most of the English underspecified senses are assigned by backoff. However, it cannot be argued that turkers have overused clicking on the first option (a common spamming behavior) because we can see that two of the English dot types (eng:artinfo, eng:procrs) have majority of metonymic senses, which are always second in the scheme (cf. Fig. 1). Looking at the amount of underspecified senses that have been obtained by majority voting for Danish and Spanish, we suggest that the level of abstraction required by this annotation is too high for turkers to perform at a level compara-

ble to that of our volunteer annotators.

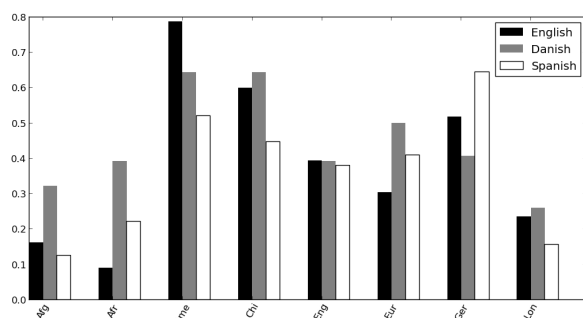


Figure 2: Proportion of non-literality in location names across languages

Figure 2 shows the proportion of non-literal (metonymic+underspecified) examples for the Location/Organization words that are common across languages. We can see that individual words show sense skewdness. This skewdness is a consequence of the kind of text in the corpus: e.g. *America* has a high proportion of non-literal senses in the ANC, where it usually means “the population or government of the US”. Similarly, it is literal less than 50% of the times for the other two languages. In contrast, *Afghanistan* is most often used in its literal location sense.

## 6 Assigning senses with MACE

Besides using majority voting with backoff, we use MACE (Hovy et al., 2013) to obtain the sense tag for each example.

| Dot type     | L   | M   | U   | D    | I  |
|--------------|-----|-----|-----|------|----|
| eng:animeat  | 340 | 146 | 14  | .048 | 3  |
| eng:artinfo  | 170 | 180 | 150 | .296 | 46 |
| eng:contcont | 295 | 176 | 28  | .174 | 0  |
| eng:locorg   | 291 | 193 | 16  | .084 | 3  |
| eng:procris  | 155 | 210 | 134 | .272 | 33 |
| da:contcont  | 223 | 134 | 143 | .242 | 79 |
| da:locorg    | 251 | 144 | 105 | .206 | 53 |
| spa:contcont | 270 | 155 | 75  | .074 | 56 |
| spa:locorg   | 302 | 146 | 52  | .038 | 40 |

Table 4: Sense distributions calculated with MACE, plus Difference and Intersection of underspecified senses between methods

MACE is an unsupervised system that uses Expectation-Maximization (EM) to estimate the competence of annotators and recover the most likely answer. MACE is designed as a Bayesian network that treats the “correct” labels as latent variables. This EM method can also be understood

as a clustering that assigns the value of the closest calculated latent variable (the sense tag) to each data point (the distribution of annotations).

Datasets that show less variation between senses calculated using majority voting and using MACE will be more reliable. Along the sense distribution in the first three columns, Table 4 provides the proportion of the senses that is different between majority voting and MACE (D), and the size of the intersection (I) of the set of underspecified examples by voting and by MACE, namely the overlap of the U columns of Tables 3 and 4.

Table 4 shows a smoother distribution of senses than Table 3, as majority classes are down-weighted by MACE. It takes very different decisions than majority voting for the two English datasets with lowest agreement (eng:artinfo, eng:procris) and for the Danish datasets, which have the fewest annotators. For these cases, the differences oscillate between 0.206 and 0.296.

Although MACE increases the frequency of the underspecified senses for all datasets but one (eng:locorg), the underspecified examples in Table 3 are not subsumed by the MACE results. The values in the I column show that none of the underspecified senses of eng:contcont receive the underspecified sense by MACE. All of these examples, however, were resolved by backoff, as well as most of the other underspecified cases in the other English datasets. In contrast to the voting method, MACE does not operate with any theoretical assumption about the relation between the three senses and treats them independently when assigning the most likely sense tag to each distribution of annotations.

## 7 Comparison between methods

The voting system and MACE provide different sense tags. The following examples (three from eng:contcont and four from eng:locorg) show disagreement between the sense tag assigned by voting and by MACE:

- d) *To ship a **crate** of lettuce across the country, a trucker needed permission from a federal regulatory agency.*
- e) *Controls were sent a package containing stool collection **vials** and instructions for collection and mailing of samples.*
- f) *In fact, it was the social committee, and our chief responsibilities were to arrange for bands and **kegs** of beer .*



- g) *The most unpopular PM in Canada’s modern history, he introduced the Goods and Services Tax , a VAT-like national sales tax.*
- h) *This is Boston’s commercial and financial heart , but it s far from being an homogeneous district [...]*
- i) *California has the highest number of people in poverty in the nation — 6.4 million, including nearly one in five children.*
- j) *Under the Emperor Qianlong (Chien Lung), Kangxi’s grandson, conflict arose between Europe’s empires and the Middle Kingdom.*

All of the previous examples were tagged as underspecified by either the voting system or MACE, but not by both. Table 5 breaks down the five annotations that each example received by turkers in literal, metonymic and underspecified. The last two columns show the sense tag provided by voting or MACE.

| Example | L | M | U | VOTING | MACE |
|---------|---|---|---|--------|------|
| d)      | 2 | 2 | 1 | U      | L    |
| e)      | 3 | 1 | 1 | L      | U    |
| f)      | 1 | 2 | 2 | M      | U    |
| g)      | 2 | 2 | 1 | U      | M    |
| h)      | 2 | 2 | 1 | U      | M    |
| i)      | 3 | 0 | 2 | L      | U    |
| j)      | 1 | 2 | 2 | M      | U    |

Table 5: Annotation summary and sense tags for the examples in this section

Just by looking at the table it is not immediate which method is preferable to assign sense tags in cases that are not clear-cut. In the case of i), we consider the underspecified sense more adequate than the literal one obtained by voting, just like we are also more prone to prefer the underspecified meaning in f), which has been assigned by MACE. In the case of h), we consider that the strictly metonymic sense assigned by MACE does not capture both the organization- (“commercial and financial”) and location-related (“district”) aspects of the meaning, and we would prefer the underspecified reading. However, MACE can also overgenerate the underspecified sense, as the vials mentioned in example e) are empty and have no content yet, thereby being literal containers and not their content.

Examples d), g) and h) have the same distribution of annotations—namely 2 literal, 2 metonymic and 1 underspecified—but d) has received the literal sense from MACE, whereas the

other two are metonymic. This difference is a result of having trained MACE independently for each dataset. The three examples receive the underspecified sense from the voting scheme, since neither the literal or metonymic sense is more present in the annotations.

On the other hand, e) and i) are skewed towards literality and receive the literal sense by plurality without having to resort to any backoff, but they are marked as underspecified by MACE.

## 8 Conclusions

We have described the annotation process of a regular-polysemy corpus in English, Danish and Spanish which deals with five different dot types. After annotating the examples for their literal, metonymic or underspecified reading, we have determined that this scheme can provide reliable ( $\alpha$  over 0.60) annotations for one dot type. Not all the dot types are equally easy to annotate. The main source of variation in agreement, and thus annotation reliability, is the dot type itself. While eng:animeat and eng:locorg appear the easiest, eng:artinfo and eng:procres obtain very low  $\alpha$  scores.

## 9 Further work

After collecting annotated data, the natural next step is to attempt class-based word-sense disambiguation (WSD) to predict the senses in Tables 3 and 4 using a state-of-the-art system like Nastase et al. (2012). We will consider a sense-assignment method (voting or MACE) as more appropriate if it provides the sense tags that are easiest to learn by our WSD system.

However, learnability is only one possible parameter for quality, and we also want to develop an expert-annotated gold standard to compare our data against. We also consider the possibility of developing a sense-assignment method that relies both on the theoretical assumption behind the voting scheme and the latent-variable approach used by MACE.

## Acknowledgments

The research leading to these results has been funded by the European Commission’s 7th Framework Program under grant agreement 238405 (CLARA).

## References

- Mette Skovgaard Andersen, Helle Asmussen, and Jørg Asmussen. 2002. The project of korpus 2000 going public. In *The Tenth EURALEX International Congress: EURALEX 2002*.
- J. D. Apresjan. 1974. Regular polysemy. *Linguistics*.
- R. Artstein and M. Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.
- Gemma Boleda, Sabine Schulte im Walde, and Toni Badia. 2012. Modeling regular polysemy: A study on the semantic classification of catalan adjectives. *Computational Linguistics*, 38(3):575–616.
- Ted Briscoe, Ann Copestake, and Alex Lascarides. 1995. Blocking. In *Computational Lexical Semantics*. Citeseer.
- Aitor Gonzalez-Agirre, Egoitz Laparra, and German Rigau. 2012. Multilingual central repository version 3.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2525–2529.
- Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of NAACL-HLT 2013*.
- N. Ide and C. Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics 2001*, pages 274–280. Citeseer.
- J. B. Johannessen, K. Haagen, K. Haaland, A. B. Jónsdóttir, A. Nøklestad, D. Kokkinakis, P. Meurer, E. Bick, and D. Haltrup. 2005. Named entity recognition for the mainland scandinavian languages. *Literary and Linguistic Computing*, 20(1):91.
- M. Lapata and A. Lascarides. 2003. A probabilistic account of logical metonymy. *Computational Linguistics*, 29(2):261–315.
- K. Markert and M. Nissim. 2002. Towards a corpus annotated for metonymies: the case of location names. In *Proc. of LREC*. Citeseer.
- K. Markert and M. Nissim. 2009. Data and models for metonymy resolution. *Language Resources and Evaluation*, 43(2):123–138.
- Vivi Nastase, Alex Judea, Katja Markert, and Michael Strube. 2012. Local and global context for supervised and unsupervised metonymy resolution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 183–193. Association for Computational Linguistics.
- Malvina Nissim and Katja Markert. 2005. Learning to buy a renault and talk to bmw: A supervised approach to conventional metonymy. In *Proceedings of the 6th International Workshop on Computational Semantics, Tilburg*.
- Geoffrey Nunberg. 1995. Transfers of meaning. *Journal of semantics*, 12(2):109–132.
- B. S. Pedersen, S. Nimb, J. Asmussen, N. H. Sørensen, L. Trap-Jensen, and H. Lorentzen. 2009. Dan-net: the challenge of compiling a wordnet for danish by reusing a monolingual dictionary. *Language resources and evaluation*, 43(3):269–299.
- Thierry Poibeau. 2006. Dealing with metonymic readings of named entities. *arXiv preprint cs/0607052*.
- J. Pustejovsky, A. Rumshisky, J. Moszkowicz, and O. Batiukova. 2009. Gml: Annotating argument selection and coercion. In *IWCS-8: Eighth International Conference on Computational Semantics*.
- J. Pustejovsky. 1995. The generative lexicon: a theory of computational lexical semantics.
- A. Rumshisky, VA Grinberg, and J. Pustejovsky. 2007. Detecting selectional behavior of complex types in text. In *Fourth International Workshop on Generative Approaches to the Lexicon, Paris, France*. Citeseer.
- Ekaterina Shutova. 2009. Sense-based interpretation of logical metonymy using a statistical method. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 1–9. Association for Computational Linguistics.
- R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.
- Jorge Vivaldi. 2009. Corpus and exploitation tool: Iulact and bwananet. In *I International Conference on Corpus Linguistics (CICL 2009), A survey on corpus-based research, Universidad de Murcia*, pages 224–239.

# Derivational Smoothing for Syntactic Distributional Semantics

Sebastian Padó\*    Jan Šnajder†    Britta Zeller\*

\*Heidelberg University, Institut für Computerlinguistik  
69120 Heidelberg, Germany

†University of Zagreb, Faculty of Electrical Engineering and Computing  
Unska 3, 10000 Zagreb, Croatia

{pado, zeller}@cl.uni-heidelberg.de    jan.snajder@fer.hr

## Abstract

Syntax-based vector spaces are used widely in lexical semantics and are more versatile than word-based spaces (Baroni and Lenci, 2010). However, they are also sparse, with resulting reliability and coverage problems. We address this problem by *derivational smoothing*, which uses knowledge about derivationally related words (*oldish* → *old*) to improve semantic similarity estimates. We develop a set of derivational smoothing methods and evaluate them on two lexical semantics tasks in German. Even for models built from very large corpora, simple derivational smoothing can improve coverage considerably.

## 1 Introduction

Distributional semantics (Turney and Pantel, 2010) builds on the assumption that the semantic similarity of words is strongly correlated to the overlap between their linguistic contexts. This hypothesis can be used to construct context vectors for words directly from large text corpora in an unsupervised manner. Such vector spaces have been applied successfully to many problems in NLP (see Turney and Pantel (2010) or Erk (2012) for current overviews).

Most distributional models in computational lexical semantics are either (a) *bag-of-words* models, where the context features are words within a surface window around the target word, or (b) *syntactic* models, where context features are typically pairs of dependency relations and context words.

The advantage of syntactic models is that they incorporate a richer, structured notion of context. This makes them more versatile; the Distributional Memory framework by Baroni and Lenci (2010) is applicable to a wide range of tasks. It is also able – at least in principle – to capture more fine-grained

types of semantic similarity such as predicate-argument plausibility (Erk et al., 2010). At the same time, syntactic spaces are much more prone to sparsity problems, as their contexts are sparser. This leads to reliability and coverage problems.

In this paper, we propose a novel strategy for combating sparsity in syntactic vector spaces, *derivational smoothing*. It follows the intuition that derivationally related words (*feed* – *feeder*, *blocked* – *blockage*) are, as a rule, semantically highly similar. Consequently, knowledge about derivationally related words can be used as a “back off” for sparse vectors in syntactic spaces. For example, the pair *oldish* – *ancient* should receive a high semantic similarity, but in practice, the vector for *oldish* will be very sparse, which makes this result uncertain. Knowing that *oldish* is derivationally related to *old* allows us to use the much less sparse vector for *old* as a proxy for *oldish*.

We present a set of general methods for smoothing vector similarity computations given a resource that groups words into derivational families (equivalence classes) and evaluate these methods on German for two distributional tasks (similarity prediction and synonym choice). We find that even for syntactic models built from very large corpora, a simple derivational resource that groups words on morphological grounds can improve the results.

## 2 Related Work

Smoothing techniques – either statistical, distributional, or knowledge-based – are widely applied in all areas of NLP. Many of the methods were first applied in Language Modeling to deal with unseen *n*-grams (Chen and Goodman, 1999; Dagan et al., 1999). Query expansion methods in Information Retrieval are also prominent cases of smoothing that addresses the lexical mismatch between query and document (Voorhees, 1994; Gonzalo et al., 1998; Navigli and Velardi, 2003). In lexical semantics, smoothing is often achieved by backing

off from words to semantic classes, either adopted from a resource such as WordNet (Resnik, 1996) or induced from data (Pantel and Lin, 2002; Wang et al., 2005; Erk et al., 2010). Similarly, distributional features support generalization in Named Entity Recognition (Finkel et al., 2005).

Although distributional information is often used for smoothing, to our knowledge there is little work on smoothing distributional models themselves. We see two main precursor studies for our work. Bergsma et al. (2008) build models of selectional preferences that include morphological features such as capitalization and the presence of digits. However, their approach is task-specific and requires a (semi-)supervised setting. Allan and Kumar (2003) make use of morphology by building language models for stemming-based equivalence classes. Our approach also uses morphological processing, albeit more precise than stemming.

### 3 A Resource for German Derivation

Derivational morphology describes the process of building new (derived) words from other (basis) words. Derived words can, but do not have to, share the part-of-speech (POS) with their basis (*old<sub>A</sub>* → *oldish<sub>A</sub>* vs. *warm<sub>A</sub>* → *warm<sub>V</sub>*, *warmth<sub>N</sub>*). Words can be grouped into *derivational families* by forming the transitive closure over individual derivation relations. The words in these families are typically semantically similar, although the exact degree depends on the type of relation and idiosyncratic factors (*book<sub>N</sub>* → *bookish<sub>A</sub>*, Lieber (2009)).

For German, there are several resources with derivational information. We use version 1.3 of DERIVBASE (Zeller et al., 2013),<sup>1</sup> a freely available resource that groups over 280,000 verbs, nouns, and adjectives into more than 17,000 non-singleton derivational families. It has a precision of 84% and a recall of 71%. Its higher coverage compared to CELEX (Baayen et al., 1996) and IMSLEX (Fitschen, 2004) makes it particularly suitable for the use in smoothing, where the resource should include low-frequency lemmas.

The following example illustrates a family that covers three POSes as well as a word with a predominant metaphorical reading (*to kneel* → *to beg*):

knieen<sub>V</sub> (*to kneel<sub>V</sub>*), beknieen<sub>V</sub> (*to beg<sub>V</sub>*), Kniende<sub>N</sub> (*kneeling\_person<sub>N</sub>*), kniend<sub>A</sub> (*kneeling<sub>A</sub>*), Knie<sub>Nn</sub> (*knee<sub>N</sub>*)

<sup>1</sup>Downloadable from: <http://goo.gl/7KG2U>

Using derivational knowledge for smoothing raises the question of how semantically similar the lemmas within a family really are. Fortunately, DERIVBASE provides information that can be used in this manner. It was constructed with hand-written derivation rules, employing string transformation functions that map basis lemmas onto derived lemmas. For example, a suffixation rule using the affix “heit” generates the derivation *dunkel* – *Dunkelheit* (*dark<sub>A</sub>* – *darkness<sub>N</sub>*). Since derivational families are defined as transitive closures, each pair of words in a family is connected by a derivation path. Because the rules do not have a perfect precision, our confidence in pairs of words decreases the longer the derivation path between them. To reflect this, we assign each pair a *confidence* of  $1/n$ , where  $n$  is the length of the shortest path between the lemmas. For example, *bekleiden* (*enrobe<sub>V</sub>*) is connected to *Verkleidung* (*disguise<sub>N</sub>*) through three steps via the lemmas *kleiden* (*dress<sub>V</sub>*) and *verkleiden* (*disguise<sub>V</sub>*) and is assigned the confidence  $1/3$ .

### 4 Models for Derivational Smoothing

Derivational smoothing exploits the fact that derivationally related words are also semantically related, by combining and/or comparing distributional representations of derivationally related words. The definition of a derivational smoothing algorithm consists of two parts: a *trigger* and a *scheme*.

**Notation.** Given a word  $w$ , we use  $\mathbf{w}$  to denote its distributional vector and  $\mathcal{D}(w)$  to denote the set of vectors for the derivational family of  $w$ . We assume that  $\mathbf{w} \in \mathcal{D}(w)$ . For words that have no derivations in DERIVBASE,  $\mathcal{D}(w)$  is a singleton set,  $\mathcal{D}(w) = \{\mathbf{w}\}$ . Let  $\alpha(w, w')$  denote the confidence of the pair  $(w, w')$ , as explained in Section 3.

**Smoothing trigger.** As discussed above, there is no guarantee for high semantic similarity within a derivational family. For this reason, smoothing may also drown out information. In this paper, we report on two triggers: *smooth always* always performs smoothing; *smooth if sim=0* smooths only when the unsmoothed similarity  $\text{sim}(\mathbf{w}_1, \mathbf{w}_2)$  is zero or unknown (due to  $w_1$  or  $w_2$  not being in the model).

**Smoothing scheme.** We present three smoothing schemes, all of which apply to the level of complete families. The first two schemes are *exemplar-based* schemes, which define the smoothed similarity for a word pair as a function of the pairwise similarities between all words of the two derivational families.

The first one, *maxSim*, checks for particularly similar words in the families:

$$\text{maxSim}(w_1, w_2) = \max_{\substack{\mathbf{w}'_1 \in \mathcal{D}(w_1) \\ \mathbf{w}'_2 \in \mathcal{D}(w_2)}} \text{sim}(\mathbf{w}'_1, \mathbf{w}'_2)$$

The second one, *avgSim*, computes the average pairwise similarity ( $N$  is the number of pairs):

$$\text{avgSim}(w_1, w_2) = \frac{1}{N} \sum_{\substack{\mathbf{w}'_1 \in \mathcal{D}(w_1) \\ \mathbf{w}'_2 \in \mathcal{D}(w_2)}} \text{sim}(\mathbf{w}'_1, \mathbf{w}'_2)$$

The third scheme, *centSim*, is *prototype-based*. It computes a centroid vector for each derivational family, which can be thought of as a representation for the concept(s) that it expresses:

$$\text{centSim}(w_1, w_2) = \text{sim}(\mathbf{c}(\mathcal{D}(w_1)), \mathbf{c}(\mathcal{D}(w_2)))$$

where  $\mathbf{c}(\mathcal{D}(w)) = \sum_{\mathbf{w}' \in \mathcal{D}(w)} \alpha(w, \mathbf{w}') \cdot \mathbf{w}'$  is the confidence-weighted centroid vector. *centSim* is similar to *avgSim*. It is more efficient to calculate and effectively introduces a kind of regularization, where outliers in either family have less impact on the overall result.

These models only represent a sample of possible derivational smoothing methods. We performed a number of additional experiments (POS-restricted smoothing, word-based, and pair-based smoothing triggers), but they did not yield any improvements over the simpler models we present here.

## 5 Experimental Evaluation

**Syntactic Distributional Model.** The syntactic distributional model that we use represents target words by pairs of dependency relations and context words. More specifically, we use the  $W \times LW$  matricization of DM.DE, the German version (Padó and Utt, 2012) of Distributional Memory (Baroni and Lenci, 2010). DM.DE was created on the basis of the 884M-token SDEWAC web corpus (Faaß et al., 2010), lemmatized, tagged, and parsed with the German MATE toolkit (Bohnet, 2010).

**Experiments.** We evaluate the impact of smoothing on two standard tasks from lexical semantics. The first task is predicting semantic similarity. We lemmatized and POS-tagged the German GUR350 dataset (Zesch et al., 2007), a set of 350 word pairs with human similarity judgments, created analogously to the well-known Rubenstein and Goodenough (1965) dataset for English.<sup>2</sup> We predict

semantic similarity as cosine similarity. We make a prediction for a word pair if both words are represented in the semantic space and their vectors have a non-zero similarity.

The second task is synonym choice on the German version of the Reader’s Digest WordPower dataset (Wallace and Wallace, 2005).<sup>2</sup> This dataset, which we also lemmatized and POS-tagged, consists of 984 target words with four synonym candidates each (including phrases), one of which is correct. Again, we compute semantic similarity as the cosine between target and a candidate vector and pick the highest-similarity candidate as synonym. For phrases, we compute the maximum pairwise word similarity. We make a prediction for an item if the target as well as at least one candidate are represented in the semantic space and their vectors have a non-zero similarity.

We expect differences between the two tasks with regard to derivational smoothing, since the words within derivational families are generally related but often not synonymous (cf. the example in Section 3). Thus, semantic similarity judgments should profit more easily from derivational smoothing than synonym choice.

**Baseline.** Our baseline is a standard bag-of-words vector space (BOW), which represents target words by the words occurring in their context. We use standard parameters ( $\pm 10$  word window, 8,000 most frequent verb, noun, and adjective lemmas). The model was created from the same corpus as DM.DE. We also applied derivational smoothing to this model, but did not obtain improvements.

**Evaluation.** To analyze the impact of smoothing, we evaluate the coverage of models and the quality of their predictions separately. In both tasks, coverage is the percentage of items for which we make a prediction. We measure quality of the semantic similarity task as the Pearson correlation between the model predictions and the human judgments for covered items (Zesch et al., 2007). For synonym choice, we follow the method established by Mohammad et al. (2007), measuring accuracy over covered items, with partial credit for ties.

**Results for Semantic Similarity.** Table 1 shows the results for the first task. The unsmoothed DM.DE model attains a correlation of  $r = 0.44$  and a coverage of 58.9%. Smoothing increases the coverage substantially to 88%. Additionally, conservative, prototype-based smoothing (if  $\text{sim} = 0$ )

<sup>2</sup>Downloadable from: <http://goo.gl/bFokI>

| Smoothing trigger          | Smoothing scheme | $r$ | Cov % |
|----------------------------|------------------|-----|-------|
| DM.DE, unsmoothed          |                  | .44 | 58.9  |
| DM.DE, smooth always       | avgSim           | .30 | 88.0  |
|                            | maxSim           | .43 | 88.0  |
|                            | centSim          | .44 | 88.0  |
| DM.DE, smooth if $sim = 0$ | avgSim           | .43 | 88.0  |
|                            | maxSim           | .42 | 88.0  |
|                            | centSim          | .47 | 88.0  |
| BoW baseline               |                  | .36 | 94.9  |

Table 1: Results on the semantic similarity task ( $r$ : Pearson correlation, Cov: Coverage)

increases correlation somewhat to  $r = 0.47$ . The difference to the unsmoothed model is not significant at  $p = 0.05$  according to Fisher’s (1925) method of comparing correlation coefficients.

The bag-of-words baseline (BOW) has a greater coverage than DM.DE models, but at the cost of lower correlation across the board. The only DM.DE model that performs worse than the BOW baseline is the non-conservative avgSim (average similarity) scheme. We attribute this weak performance to the presence of many pairwise zero similarities in the data, which makes the avgSim predictions unreliable.

To our knowledge, there are no previous published papers on distributional approaches to modeling this dataset. The best previous result is a GermaNet/Wikipedia-based model by Zesch et al. (2007). It reports a higher correlation ( $r = 0.59$ ) but a very low coverage at 33.1%.

**Results for Synonym Choice.** The results for the second task are shown in Table 2. The unsmoothed model achieves an accuracy of 53.7% and a coverage of 80.8%, as reported by Padó and Utt (2012). Smoothing increases the coverage by almost 6% to 86.6% (for example, a question item for *inferior* becomes covered after backing off from the target to *Inferiorität* (*inferiority*)). All smoothed models show a loss in accuracy, albeit small. The best model is again a conservative smoothing model ( $sim = 0$ ) with a loss of 1.1% accuracy. Using bootstrap resampling (Efron and Tibshirani, 1993), we established that the difference to the unsmoothed DM.DE model is not significant at  $p < 0.05$ . This time, the avgSim (average similarity) smoothing scheme performs best, with the prototype-based scheme in second place. Thus, the results for synonym choice are less clear-cut: derivational smoothing can trade accuracy against

| Smoothing trigger                   | Smoothing scheme | Acc %       | Cov % |
|-------------------------------------|------------------|-------------|-------|
| DM.DE, unsmoothed (Padó & Utt 2012) |                  | 53.7        | 80.8  |
| DM.DE, smooth always                | avgSim           | 46.0        | 86.6  |
|                                     | maxSim           | 50.3        | 86.6  |
|                                     | centSim          | 49.1        | 86.6  |
| DM.DE, smooth if $sim = 0$          | avgSim           | 52.6        | 86.6  |
|                                     | maxSim           | 51.2        | 86.6  |
|                                     | centSim          | 51.3        | 86.6  |
| BoW “baseline”                      |                  | <b>56.9</b> | 98.5  |

Table 2: Results on the synonym choice task (Acc: Accuracy, Cov: Coverage)

coverage but does not lead to a clear improvement. What is more, the BOW “baseline” significantly outperforms all syntactic models, smoothed and unsmoothed, with an almost perfect coverage combined with a higher accuracy.

## 6 Conclusions and Outlook

In this paper, we have introduced derivational smoothing, a novel strategy to combating sparsity in syntactic vector spaces by comparing and combining the vectors of morphologically related lemmas. The only information strictly necessary for the methods we propose is a grouping of lemmas into derivationally related classes. We have demonstrated that derivational smoothing improves two tasks, increasing coverage substantially and also leading to a numerically higher correlation in the semantic similarity task, even for vectors created from a very large corpus. We obtained the best results for a conservative approach, smoothing only zero similarities. This also explains our failure to improve less sparse word-based models, where very few pairs are assigned a similarity of zero. A comparison of prototype- and exemplar-based schemes did not yield a clear winner. The estimation of generic semantic similarity can profit more from derivational smoothing than the induction of specific lexical relations.

In future work, we plan to work on other evaluation tasks, application to other languages, and more sophisticated smoothing schemes.

**Acknowledgments.** Authors 1 and 3 were supported by the EC project EXCITEMENT (FP7 ICT-287923). Author 2 was supported by the Croatian Science Foundation (project 02.03/162: “Derivational Semantic Models for Information Retrieval”). We thank Jason Utt for his support and expertise.

## References

- James Allan and Giridhar Kumaran. 2003. Stemming in the Language Modeling Framework. In *Proceedings of SIGIR*, pages 455–456.
- Harald R. Baayen, Richard Piepenbrock, and Leon Gullikers. 1996. *The CELEX Lexical Database. Release 2. LDC96L14*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.
- Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A General Framework for Corpus-based Semantics. *Computational Linguistics*, 36(4):673–721.
- Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative Learning of Selectional Preference from Unlabeled Text. In *Proceedings of EMNLP*, pages 59–68, Honolulu, Hawaii.
- 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.
- Stanley F. Chen and Joshua Goodman. 1999. An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language*, 13(4):359–394.
- Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning*, 34(1–3):43–69.
- Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A Flexible, Corpus-driven Model of Regular and Inverse Selectional Preferences. *Computational Linguistics*, 36(4):723–763.
- 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-2010*, pages 803–810.
- 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 of the ACL*, pages 363–370.
- Ronald Aylmer Fisher. 1925. *Statistical methods for research workers*. Oliver and Boyd, Edinburgh.
- Arne Fitschen. 2004. *Ein computerlinguistisches Lexikon als komplexes System*. Ph.D. thesis, IMS, Universität Stuttgart.
- Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan M. Cigarrán. 1998. Indexing with WordNet Synsets Can Improve Text Retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, Montréal, Canada.
- Rochelle Lieber. 2009. *Morphology and Lexical Semantics*. Cambridge University Press.
- Saif Mohammad, Iryna Gurevych, Graeme Hirst, and Torsten Zesch. 2007. Cross-Lingual Distributional Profiles of Concepts for Measuring Semantic Distance. In *Proceedings of the 2007 Joint Conference on EMNLP and CoNLL*, pages 571–580, Prague, Czech Republic.
- Roberto Navigli and Paola Velardi. 2003. An Analysis of Ontology-based Query Expansion Strategies. In *Workshop on Adaptive Text Extraction and Mining*, Dubrovnik, Croatia.
- Sebastian Padó and Jason Utt. 2012. A Distributional Memory for German. In *Proceedings of KONVENS 2012 workshop on lexical-semantic resources and applications*, pages 462–470, Vienna, Austria.
- Patrick Pantel and Dekang Lin. 2002. Discovering Word Senses from Text. In *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619.
- Philip Resnik. 1996. Selectional Constraints: An Information-theoretic Model and its Computational Realization. *Cognition*, 61(1–2):127–159.
- Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- Ellen M. Voorhees. 1994. Query Expansion Using Lexical-semantic Relations. In *Proceedings of SIGIR*, pages 61–69.
- DeWitt Wallace and Lila Acheson Wallace. 2005. *Reader's Digest, das Beste für Deutschland*. Verlag Das Beste, Stuttgart.
- Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2005. Strictly Lexical Dependency Parsing. In *Proceedings of IWPT*, pages 152–159.
- Britta Zeller, Jan Šnajder, and Sebastian Padó. 2013. DERivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of ACL*, Sofia, Bulgaria.
- Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. 2007. Comparing Wikipedia and German Wordnet by Evaluating Semantic Relatedness on Multiple Datasets. In *Proceedings of NAACL/HLT*, pages 205–208, Rochester, NY.

# Diathesis alternation approximation for verb clustering

Lin Sun

Greedy Intelligence Ltd  
Hangzhou, China  
lin.sun@greedyint.com

Diana McCarthy and Anna Korhonen

DTAL and Computer Laboratory  
University of Cambridge  
Cambridge, UK  
diana@dianamccarthy.co.uk  
alk23@cam.ac.uk

## Abstract

Although diathesis alternations have been used as features for manual verb classification, and there is recent work on incorporating such features in computational models of human language acquisition, work on large scale verb classification has yet to examine the potential for using diathesis alternations as input features to the clustering process. This paper proposes a method for approximating diathesis alternation behaviour in corpus data and shows, using a state-of-the-art verb clustering system, that features based on alternation approximation outperform those based on independent subcategorization frames. Our alternation-based approach is particularly adept at leveraging information from less frequent data.

## 1 Introduction

Diathesis alternations (DAs) are regular alternations of the syntactic expression of verbal arguments, sometimes accompanied by a change in meaning. For example, *The man broke the window*  $\leftrightarrow$  *The window broke*. The syntactic phenomena are triggered by the underlying semantics of the participating verbs. Levin (1993)'s seminal book provides a manual inventory both of DAs and verb classes where membership is determined according to participation in these alternations. For example, most of the COOK verbs (e.g. bake, cook, fry ...) can all take various DAs, such as the causative alternation, middle alternation and instrument subject alternation.

In computational linguistics, work inspired by Levin's classification has exploited the link between syntax and semantics for producing classifications of verbs. Such classifications are useful for a wide variety of purposes such as semantic role labelling (Gildea and Jurafsky, 2002),

predicting unseen syntax (Parisien and Stevenson, 2010), argument zoning (Guo *et al.*, 2011) and metaphor identification (Shutova *et al.*, 2010). While Levin's classification can be extended manually (Kipper-Schuler, 2005), a large body of research has developed methods for automatic verb classification since such methods can be applied easily to other domains and languages.

Existing work on automatic classification relies largely on syntactic features such as subcategorization frames (SCF)s (Schulte im Walde, 2006; Sun and Korhonen, 2011; Vlachos *et al.*, 2009; Brew and Schulte im Walde, 2002). There has also been some success incorporating selectional preferences (Sun and Korhonen, 2009).

Few have attempted to use, or approximate, diathesis features directly for verb classification although manual classifications have relied on them heavily, and there has been related work on identifying the DAs themselves automatically using SCF and semantic information (Resnik, 1993; McCarthy and Korhonen, 1998; Lapata, 1999; McCarthy, 2000; Tsang and Stevenson, 2004). Exceptions to this include Merlo and Stevenson (2001), Joanis *et al.* (2008) and Parisien and Stevenson (2010, 2011). Merlo and Stevenson (2001) used cues such as passive voice, animacy and syntactic frames coupled with the overlap of lexical fillers between the alternating slots to predict a 3-way classification (unergative, unaccusative and object-drop). Joanis *et al.* (2008) used similar features to classify verbs on a much larger scale. They classify up to 496 verbs using 11 different classifications each having between 2 and 14 classes. Parisien and Stevenson (2010, 2011) used hierarchical Bayesian models on slot frequency data obtained from child-directed speech parsed with a dependency parser to model acquisition of SCF, alternations and ultimately verb classes which provided predictions for unseen syntactic behaviour of class members.



| Frame     | Example sentence                    | Freq |
|-----------|-------------------------------------|------|
| NP+PPon   | Jessica sprayed paint on the wall   | 40   |
| NP+PPwith | Jessica sprayed the wall with paint | 30   |
| PPwith    | *The wall sprayed with paint        | 0    |
| PPon      | Jessica sprayed paint on the wall   | 30   |

Table 1: Example frames for verb spray

In this paper, like Sun and Korhonen (2009); Joanis *et al.* (2008) we seek to automatically classify verbs into a broad range of classes. Like Joanis *et al.*, we include evidence of DA, but we do not manually select features attributed to specific alternations but rather experiment with syntactic evidence for alternation approximation. We use the verb clustering system presented in Sun and Korhonen (2009) because it achieves state-of-the-art results on several datasets, including those of Joanis *et al.*, even without the additional boost in performance from the selectional preference data. We are interested in the improvement that can be achieved to verb clustering using approximations for DAs, rather than the DA per se. As such we make the simple assumption that if a pair of SCFs tends to occur with the same verbs, we have a potential occurrence of DA. Although this approximation can give rise to false positives (pairs of frames that co-occur frequently but are not DA) we are nevertheless interested in investigating its potential usefulness for verb classification. One attractive aspect of this method is that it does not require a pre-defined list of possible alternations.

## 2 Diathesis Alternation Approximation

A DA can be approximated by a pair of SCFs. We parameterize frames involving prepositional phrases with the preposition. Example SCFs for the verb “spray” are shown in Table 1. The feature value of a single frame feature is the frequency of the SCF. Given two frames  $f_v(i), f_v(j)$  of a verb  $v$ , they can be transformed into a feature pair  $(f_v(i), f_v(j))$  as an approximation to a DA. The feature value of the DA feature  $(f_v(i), f_v(j))$  is approximated by the joint probability of the pair of frames  $p(f_v(i), f_v(j)|v)$ , obtained by integrating all the possible DAs. The key assumption is that the joint probability of two SCFs has a strong correlation with a DA on the grounds that the DA gives rise to both SCFs in the pair. We use the DA feature  $(f_v(i), f_v(j))$  with its value  $p(f_v(i), f_v(j)|v)$  as a new feature for verb clustering. As a comparison point, we can ignore the DA and make a frame independence assumption. The joint probability is

decomposed as:

$$p(f_v(i), f_v(j)|v)' \triangleq p(f_v(i)|v) \cdot p(f_v(j)|v) \quad (1)$$

We assume that SCFs are dependent as they are generated by the underlying meaning components (Levin and Hovav, 2006). The frame dependency is represented by a simple graphical model in figure 1.

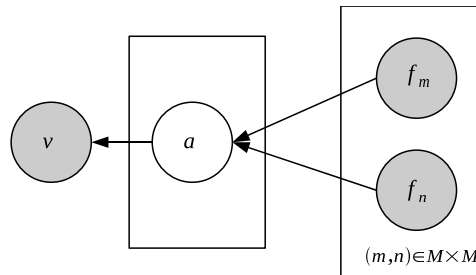


Figure 1: Graphical model for the joint probability of pairs of frames.  $v$  represents a verb,  $a$  represents a DA and  $f$  represents a specific frame in total of  $M$  possible frames

In the data, the verb ( $v$ ) and frames ( $f$ ) are observed, and any underlying alternation ( $a$ ) is hidden. The aim is to approximate but not to detect a DA, so  $a$  is summed out:

$$p(f_v(i), f_v(j)|v) = \sum_a p(f_v(i), f_v(j)|a) \cdot p(a|v) \quad (2)$$

In order to evaluate this sum, we use a relaxation<sup>1</sup>: the *sum* in equation 1 is replaced with the maximum (*max*). This is a reasonable relaxation, as a pair of frames rarely participates in more than one type of a DA.

$$p(f_v(i), f_v(j)|v) \approx \max(p(f_v(i), f_v(j)|a) \cdot p(a|v)) \quad (3)$$

The second relaxation further relaxes the first one by replacing the *max* with the least upper bound (*sup*): If  $f_v(i)$  occurs  $a$  times,  $f_v(j)$  occurs  $b$  times and  $b < a$ , the number of times that a DA occurs between  $f_v(i)$  and  $f_v(j)$  must be smaller or equal to  $b$ .

$$p(f_v(i), f_v(j)|v) \approx \sup\{p(f_v(i), f_v(j)|a)\} \cdot \sup\{p(a|v)\} \quad (4)$$

$$\sup\{p(f_v(i), f_v(j)|a)\} = Z^{-1} \cdot \min(f_v(i), f_v(j))$$

$$\sup\{p(a|v)\} = 1$$

$$Z = \sum_m \sum_n \min(f_v(m), f_v(n))$$

<sup>1</sup>A relaxation is used in mathematical optimization for relaxing the strict requirement, by either substituting it with an easier requirement or dropping it completely.

| Frame pair          | Possible DA     | Frequency |
|---------------------|-----------------|-----------|
| NP+PPon NP+PPwith   | Locative        | 30        |
| NP+PPon PPwith      | Causative(with) | 0         |
| NP+PPon PPon        | Causative(on)   | 30        |
| NP+PPwith PPwith    | ?               | 0         |
| NP+PPwith PPon      | ?               | 30        |
| PPwith PPon         | ?               | 0         |
| NP+PPon NP+PPon     | -               | 40        |
| NP+PPwith NP+PPwith | -               | 30        |
| PPwith PPwith       | -               | 0         |
| PPon PPon           | -               | 30        |

Table 2: Example frame pair features for *spray*

So we end up with a simple form:

$$p(f_v(i), f_v(j)|v) \approx Z^{-1} \cdot \min(f_v(i), f_v(j)) \quad (5)$$

The equation is intuitive: If  $f_v(i)$  occurs 40 times and  $f_v(j)$  30 times, the DA between  $f_v(i)$  and  $f_v(j) \leq 30$  times. This upper bound value is used as the feature value of the DA feature. The original feature vector  $\mathbf{f}$  of dimension  $M$  is transformed into  $M^2$  dimensions feature vector  $\tilde{\mathbf{f}}$ . Table 2 shows the transformed feature space for *spray*. The feature space matches our expectation well: valid DAs have a value greater than 0 and invalid DAs have a value of 0.

### 3 Experiments

We evaluated this model by performing verb clustering experiments using three feature sets:

**F1:** SCF parameterized with preposition. Examples are shown in Table 1.

**F2:** The frame pair features built from F1 with the frame independence assumption (equation 1). This feature is not a DA feature as it ignores the inter-dependency of the frames.

**F3:** The frame pair features (DAs) built from F1 with the frame dependency assumption (equation 4). This is the DA feature which considers the correlation of the two frames which are generated from the alternation.

F3 implicitly includes F1, as a frame can pair with itself.<sup>2</sup> In the example in Table 2, the frame pair ‘‘PP(on) PP(on)’’ will always have the same value as the ‘‘PP(on)’’ frame in F1.

We extracted the SCFs using the system of Preiss *et al.* (2007) which classifies each corpus

<sup>2</sup>We did this so that F3 included the SCF features as well as the DA approximation features. It would be possible in future work to exclude the pairs involving identical frames, thereby relying solely on the DA approximations, and compare performance with the results obtained here.

occurrence of a verb as a member of one of the 168 SCFs on the basis of grammatical relations identified by the RASP (Briscoe *et al.*, 2006) parser. We experimented with two datasets that have been used in prior work on verb clustering: the test sets 7-11 (3-14 classes) in Joanis *et al.* (2008), and the 17 classes set in Sun *et al.* (2008).

We used the spectral clustering (SPEC) method and settings as in Sun and Korhonen (2009) but adopted the Bhattacharyya kernel (Jebara and Kondor, 2003) to improve the computational efficiency of the approach given the high dimensionality of the quadratic feature space.

$$w_b(v, v') = \sum_{d=1}^D (v_d v'_d)^{1/2} \quad (6)$$

The mean-filed bound of the Bhattacharyya kernel is very similar to the KL divergence kernel (Jebara *et al.*, 2004) which is frequently used in verb clustering experiments (Korhonen *et al.*, 2003; Sun and Korhonen, 2009).

To further reduce computational complexity, we restricted our scope to the more frequent features. In the experiment described in this section we used the 50 most frequent features for the 3-6 way classifications (Joaanis *et al.*’s test set 7-9) and 100 features for the 7-17 way classifications. In the next section, we will demonstrate that F3 outperforms F1 regardless of the feature number setting. The features are normalized to sum 1.

The clustering results are evaluated using F-Measure as in Sun and Korhonen (2009) which provides the harmonic mean of precision ( $P$ ) and recall ( $R$ )

$P$  is calculated using modified purity – a global measure which evaluates the mean precision of clusters. Each cluster ( $k_i \in K$ ) is associated with the gold-standard class to which the majority of its members belong. The number of verbs in a cluster ( $k_i$ ) that take this class is denoted by  $n_{prevalent}(k_i)$ .

$$P = \frac{\sum_{k_i \in K: n_{prevalent}(k_i) > 2} n_{prevalent}(k_i)}{|\text{verbs}|}$$

$R$  is calculated using weighted class accuracy: the proportion of members of the dominant cluster  $\text{DOM-CLUST}_i$  within each of the gold-standard classes  $c_i \in C$ .

|    | Datasets      |              |              |              |              |              |
|----|---------------|--------------|--------------|--------------|--------------|--------------|
|    | Joanis et al. |              |              |              |              | Sun et al.   |
|    | 7             | 8            | 9            | 10           | 11           |              |
| F1 | 54.54         | 49.97        | 35.77        | 46.61        | 38.81        | 60.03        |
| F2 | 50.00         | 49.50        | 32.79        | 54.13        | 40.61        | 64.00        |
| F3 | <b>56.36</b>  | <b>53.79</b> | <b>52.90</b> | <b>66.32</b> | <b>50.97</b> | <b>69.62</b> |

Table 3: Results when using F3 (DA), F2 (pair of independent frames) and F1 (single frame) features with Bhattacharyya kernel on Joanis et al. and Sun et al. datasets

$$R = \frac{\sum_{i=1}^{|C|} |\text{verbs in DOM-CLUST}_i|}{|\text{verbs}|}$$

The results are shown in Table 3. The result of F2 is lower than that of F3, and even lower than that of F1 for 3-6 way classification. This indicates that the frame independence assumption is a poor assumption. F3 yields substantially better result than F2 and F1. The result of F3 is 6.4% higher than the result (F=63.28) reported in Sun and Korhonen (2009) using the F1 feature.

This experiment shows, on two datasets, that DA features are clearly more effective than the frame features for verb clustering, even when relaxations are used.

#### 4 Analysis of Feature Frequency

A further experiment was carried out using F1 and F3 on Joanis *et al.* (2008)'s test sets 10 and 11. The frequency ranked features were added to the clustering one at a time, starting from the most frequent one. The results are shown in figure 2. F3 outperforms F1 clearly on all the feature number settings. After adding some highly frequent frames (22 for test set 10 and 67 for test set 11), the performance for F1 is not further improved. The performance of F3, in contrast, is improved for almost all (including the mid-range frequency) frames, although to a lesser degree for low frequency frames.

#### 5 Related work

Parisien and Stevenson (2010) introduced a hierarchical Bayesian model capable of learning verb alternations and constructions from syntactic input. The focus was on modelling and explaining the child alternation acquisition rather than on automatic verb classification. Therefore, no quantitative evaluation of the clustering is reported, and the number of verbs under the novel verb generalization test is relatively small. Parisien and

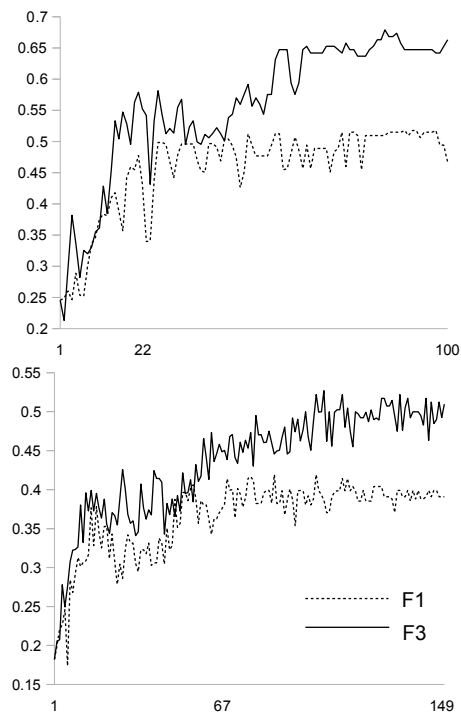


Figure 2: Comparison between frame features (F1) and DA features (F3) with different feature number settings. DA features clearly outperform frame features. The top figure is the result on test set 10 (8 ways). The bottom figure is the result on test set 11 (14 ways). The x axis is the number of features. The y axis is the F-Measure result.

Stevenson (2011) extended this work by adding semantic features.

Parisien and Stevenson's (2010) model 2 has a similar structure to the graphic model in figure 1. A fundamental difference is that we explicitly use a probability distribution over alternations (pair of frames) to represent a verb, whereas they represent a verb by a distribution over the observed frames similar to Vlachos *et al.* (2009)'s approach. Also the parameters in their model were inferred by Gibbs sampling whereas we avoided this inference step by using relaxation.

#### 6 Conclusion and Future work

We have demonstrated the merits of using DAs for verb clustering compared to the SCF data from which they are derived on standard verb classification datasets and when integrated in a state-of-the-art verb clustering system. We have also demonstrated that the performance of frame features is dominated by the high frequency frames. In contrast, the DA features enable the mid-range frequency frames to further improve the performance.

In the future, we plan to evaluate the performance of DA features in a larger scale experiment. Due to the high dimensionality of the transformed feature space (quadratic of the original feature space), we will need to improve the computational efficiency further, e.g. via use of an unsupervised dimensionality reduction technique Zhao and Liu (2007). Moreover, we plan to use Bayesian inference as in Vlachos *et al.* (2009); Parisien and Stevenson (2010, 2011) to infer the actual parameter values and avoid the relaxation.

Finally, we plan to supplement the DA feature with evidence from the slot fillers of the alternating slots, in the spirit of earlier work (McCarthy, 2000; Merlo and Stevenson, 2001; Joanis *et al.*, 2008). Unlike these previous works, we will use selectional preferences to generalize the argument heads but will do so using preferences from distributional data (Sun and Korhonen, 2009) rather than WordNet, and use *all argument head data* in *all* frames. We envisage using maximum average distributional similarity of the argument heads in any potentially alternating slots in a pair of co-occurring frames as a feature, just as we currently use the frequency of the less frequent co-occurring frame.

### Acknowledgement

Our work was funded by the Royal Society University Research Fellowship (AK) and the Dorothy Hodgkin Postgraduate Award (LS).

### References

- C. Brew and S. Schulte im Walde. Spectral clustering for German verbs. In *Proceedings of EMNLP*, 2002.
- E. Briscoe, J. Carroll, and R. Watson. The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 77–80, 2006.
- D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
- Y. Guo, A. Korhonen, and T. Poibeau. A weakly-supervised approach to argumentative zoning of scientific documents. In *Proceedings of EMNLP*, pages 273–283, Stroudsburg, PA, USA, 2011. ACL.
- T. Jebara and R. Kondor. Bhattacharyya and expected likelihood kernels. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, page 57. Springer, 2003.
- T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.
- E. Joanis, S. Stevenson, and D. James. A general feature space for automatic verb classification. *Natural Language Engineering*, 2008.
- K. Kipper-Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA, June 2005.
- A. Korhonen, Y. Krymolowski, and Z. Marx. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of ACL*, pages 64–71, Morristown, NJ, USA, 2003. ACL.
- M. Lapata. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of ACL*, pages 397–404. ACL Morristown, NJ, USA, 1999.
- B. Levin and M. Hovav. Argument realization. *Computational Linguistics*, 32(3):447–450, 2006.
- B. Levin. *English Verb Classes and Alternations: a preliminary investigation*. University of Chicago Press, Chicago and London, 1993.
- D. McCarthy and A. Korhonen. Detecting verbal participation in diathesis alternations. In *Proceedings of ACL*, volume 36, pages 1493–1495. ACL, 1998.
- D. McCarthy. Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of NAACL*, pages 256–263. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2000.
- P. Merlo and S. Stevenson. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408, 2001.
- C. Parisien and S. Stevenson. Learning verb alternations in a usage-based Bayesian model. In *Proceedings of the 32nd annual meeting of the Cognitive Science Society*, 2010.
- C. Parisien and S. Stevenson. Generalizing between form and meaning using learned verb classes. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, 2011.

- J. Preiss, T. Briscoe, and A. Korhonen. A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora. In *Proceedings of ACL*, volume 45, page 912, 2007.
- P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania, 1993.
- S. Schulte im Walde. Experiments on the automatic induction of German semantic verb classes. *Computational Linguistics*, 32(2):159–194, 2006.
- E. Shutova, L. Sun, and A. Korhonen. Metaphor identification using verb and noun clustering. In *Proceedings of COLING*, pages 1002–1010. ACL, 2010.
- L. Sun and A. Korhonen. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of EMNLP*, pages 638–647, 2009.
- L. Sun and A. Korhonen. Hierarchical verb clustering using graph factorization. In *Proceedings of EMNLP*, pages 1023–1033, Edinburgh, Scotland, UK., July 2011. ACL.
- L. Sun, A. Korhonen, and Y. Krymolowski. Verb class discovery from rich syntactic data. *Lecture Notes in Computer Science*, 4919:16, 2008.
- V. Tsang and S. Stevenson. Using selectional profile distance to detect verb alternations. In *HLT/NAACL 2004 Workshop on Computational Lexical Semantics*, 2004.
- A. Vlachos, A. Korhonen, and Z. Ghahramani. Unsupervised and constrained dirichlet process mixture models for verb clustering. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 74–82, 2009.
- Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of ICML*, pages 1151–1157, New York, NY, USA, 2007. ACM.

# Outsourcing FrameNet to the Crowd

Marco Fossati, Claudio Giuliano, and Sara Tonelli

Fondazione Bruno Kessler

Trento, Italy

{fossati, giuliano, satonelli}@fbk.eu

## Abstract

We present the first attempt to perform full FrameNet annotation with crowdsourcing techniques. We compare two approaches: the first one is the standard annotation methodology of lexical units and frame elements in two steps, while the second is a novel approach aimed at acquiring frames in a bottom-up fashion, starting from frame element annotation. We show that our methodology, relying on a single annotation step and on simplified role definitions, outperforms the standard one both in terms of accuracy and time.

## 1 Introduction

Annotating frame information is a complex task, usually modeled in two steps: first annotators are asked to choose the situation (or *frame*) evoked by a given predicate (the *lexical unit*, *LU*) in a sentence, and then they assign the semantic roles (or *frame elements*, *FEs*) that describe the participants typically involved in the chosen frame. Existing frame annotation tools, such as Salto (Burchardt et al., 2006) and the Berkeley system (Fillmore et al., 2002) foresee this two-step approach, in which annotators first select a frame from a large repository of possible frames (1,162 frames are currently listed in the online version of the resource), and then assign the FE labels constrained by the chosen frame to LU dependents.

In this paper, we argue that such workflow shows some redundancy which can be addressed by radically changing the annotation methodology and performing it in one single step. Our novel annotation approach is also more compliant with the definition of *frames* proposed in Fillmore (1976): in his seminal work, Fillmore postulated that the meanings of words can be understood on the basis of a semantic frame, i.e. a description of a type

of event or entity and the participants in it. This implies that frames can be distinguished one from another on the basis of the participants involved, thus it seems more cognitively plausible to start from the FE annotation to identify the frame expressed in a sentence, and not the contrary.

The goal of our methodology is to provide full frame annotation in a single step and in a bottom-up fashion. Instead of choosing the frame first, we focus on FEs and let the frame emerge based on the chosen FEs. We believe this approach complies better with the cognitive activity performed by annotators, while the 2-step methodology is more artificial and introduces some redundancy because part of the annotators' choices are replicated in the two steps (i.e. in order to assign a frame, annotators implicitly identify the participants also in the first step, even if they are annotated later).

Another issue we investigate in this work is how semantic roles should be annotated in a crowdsourcing framework. This task is particularly complex, therefore it is usually performed by expert annotators under the supervision of linguistic experts and lexicographers, as in the case of FrameNet. In NLP, different annotation efforts for encoding semantic roles have been carried out, each applying its own methodology and annotation guidelines (see for instance Ruppenhofer et al. (2006) for FrameNet and Palmer et al. (2005) for PropBank). In this work, we present a pilot study in which we assess to what extent role descriptions meant for 'linguistics experts' are also suitable for annotators from the crowd. Moreover, we show how a simplified version of these descriptions, less bounded to a specific linguistic theory, improve the annotation quality.

## 2 Related work

The construction of annotation datasets for NLP tasks via non-expert contributors has been ap-

proached in different ways, the most prominent being games with a purpose (GWAP) and micro-tasks. Verbosity (Von Ahn et al., 2006) was one of the first attempts in gathering annotations with a GWAP. Phrase Detectives (Chamberlain et al., 2008; Chamberlain et al., 2009) was meant to gather a corpus with coreference resolution annotations. Snow et al. (2008) described design and evaluation guidelines for five natural language micro-tasks. However, they explicitly chose a set of tasks that could be easily understood by non-expert contributors, thus leaving the recruitment and training issues open. Negri et al. (2011) built a multilingual textual entailment dataset for statistical machine translation systems.

The semantic role labeling problem has been recently addressed via crowdsourcing by Hong and Baker (2011). Furthermore, Baker (2012) highlighted the crucial role of recruiting people from the crowd in order to bypass the need for linguistics expert annotations. Nevertheless, Hong and Baker (2011) focused on the frame discrimination task, namely selecting the correct frame evoked by a given lemma. Such task is comparable to the word sense disambiguation one as per (Snow et al., 2008), although the complexity increased, due to lower inter-annotator agreement values.

### 3 Experiments

In this section, we describe the anatomy and discuss the results of the tasks we outsourced to the crowd via the CrowdFlower<sup>1</sup> platform.

**Golden data** Quality control of the collected judgements is a key factor for the success of the experiments. Cheating risk is minimized by adding *gold* units, namely data for which the requester already knows the answer. If a worker misses too many gold answers within a given threshold, he or she will be flagged as untrusted and his or her judgments will be automatically discarded.

**Worker switching effect** Depending on their accuracy in providing answers to gold units, workers may switch from a trusted to an untrusted status and vice versa. In practice, a worker submits his or her responses via a web page. Each page contains one gold unit and a variable number of regular units that can be set by the requester during the calibration phase. If a worker becomes un-

trusted, the platform collects another judgment to fill the gap. If a worker moves back to the trusted status, his or her previous contribution is added to the results as free extra judgments. Such phenomenon typically occurs when the complexity of gold units is high enough to induce low agreement in workers' answers. Thus, the requester is constrained to review gold units and to eventually forgive workers who missed them. This has massively happened in our experiments and is one of the main causes of the overall cost decrease and time increase.

**Cost calibration** The total cost of a generic crowdsourcing task is naturally bound to a data unit. This represents an issue in most of our experiments, as the number of questions per unit (i.e. a sentence) varies according to the number of frames and FEs evoked by the LU contained in a sentence. In order to enable cost comparison, for each experiment we need to use the average number of questions per sentence as a multiplier to a constant cost per sentence. We set the payment per working page to 5 \$ cents and the number of sentences per page to 3, resulting in 1.83 \$ cent per sentence.

#### 3.1 Assessing task reproducibility and worker behavior change

Since our overall goal is to compare the performance of FrameNet annotation using our novel workflow to the performance of the standard, 2-step approach, we first take into account past related works and try to reproduce them.

To our knowledge, the only attempt to annotate frame information through crowdsourcing is the one presented in Hong and Baker (2011), which however did not include FE annotation.

**Modeling** The task is designed as follows. (a) Workers are invited to read a sentence where a LU is bolded. (b) The question `Which is the correct sense?` is combined with the set of frames evoked by the given LU, as well as the `None` choice. Finally, (c) workers must select the correct frame. A set of example sentences corresponding to each possible frame is provided in the instructions to facilitate workers.

As a preliminary study, we wanted to assess to what extent the proposed task could be reproduced and if workers reacted in a comparable way over time. Hong and Baker (2011) did not publish the input datasets, thus we ignore which sen-

<sup>1</sup><https://crowdfLOWER.com>

| LU               | 2013                |          | 2011<br>Accuracy |
|------------------|---------------------|----------|------------------|
|                  | Sentences<br>(Gold) | Accuracy |                  |
| <i>high.a</i>    | 68 (9)              | 91.8     | 92               |
| <i>history.n</i> | 72 (9)              | 84.6     | 86               |
| <i>range.n</i>   | 65 (8)              | 95       | 93               |
| <i>rip.v</i>     | 88 (12)             | 81.9     | 92               |
| <i>thirst.n</i>  | 29 (4)              | 90.4     | 95               |
| <i>top.a</i>     | 36 (5)              | 98.7     | 96               |

Table 1: Comparison of the reproduced frame discrimination task as per (Hong and Baker, 2011)

tences were used. Besides, the authors computed accuracy values directly from the results upon a majority vote ground truth. Therefore, we decided to consider the same LUs used in Hong and Baker’s experiments, i.e. *high.a*, *history.n*, *range.n*, *rip.v*, *thirst.n* and *top.a*, but we leveraged the complete sets of FrameNet 1.5 expert-annotated sentences as gold-standard data for immediate accuracy computation.

**Discussion** Table 1 displays the results we achieved, jointly with the experiments by Hong and Baker (2011). For the latter, we only show accuracy values, as the number of sentences was set to a constant value of 18, 2 of which were gold. If we assume that the crowd-based ground truth in 2011 experiments is approximately equivalent to the expert one, workers seem to have reacted in a similar manner compared to Hong and Baker’s values, except for *rip.v*.

### 3.2 General task setting

We randomly chose the following LUs among the set of all verbal LUs in FrameNet evoking 2 frames each: *disappear.v* [CEASING\_TO\_BE, DEPARTING], *guide.v* [COTHEME, INFLUENCE\_OF\_EVENT\_ON\_COGNIZER], *heap.v* [FILLING, PLACING], *throw.v* [BODY\_MOVEMENT, CAUSE\_MOTION]. We considered verbal LUs as they usually have more overt arguments in a sentence, so that we were sure to provide workers with enough candidate FEs to annotate. Linguistic tasks in crowdsourcing frameworks are usually decomposed to make them accessible to the crowd. Hence, we set the polysemy of LUs to 2 to ensure that all experiments are executed using the smallest-scale subtask. More frames can then be handled by just replicating the experiments.

### 3.3 2-step approach

After observing that we were able to achieve similar results on the frame discrimination task as in previous work, we focused on the comparison between the 2-step and the 1-step frame annotation approaches.

We first set up experiments that emulate the former approach both in frame discrimination and FEs annotation. This will serve as the baseline against our methodology. Given the pipeline nature of the approach, errors in the frame discrimination step will affect FE recognition, thus impacting on the final accuracy. The magnitude of such effect strictly depends on the number of FEs associated with the wrongly detected frame.

#### 3.3.1 Frame discrimination

Frame discrimination is the first phase of the 2-step annotation procedure. Hence, we need to leverage its output as the input for the next step.

**Modeling** The task is modeled as per Section 3.1.

**Discussion** Table 2 gives an insight into the results, which confirm the overall good accuracy as per the experiments discussed in Section 3.1.

#### 3.3.2 Frame elements recognition

We consider all sentences annotated in the previous subtask with the frame assigned by the workers, even if it is not correct.

**Modeling** The task is presented as follows. (a) Workers are invited to read a sentence where a LU is bolded and the frame that was identified in the first step is provided as a title. (b) A list of FE definitions is then shown together with the FEs text chunks. Finally, (c) workers must match each definition with the proper FE.

**Simplification** Since FEs annotation is a very challenging task, and FE definitions are usually meant for experts in linguistics, we experimented with three different types of FE definitions: the original ones from FrameNet, a manually simplified version, and an automatically simplified one, using the tool by Heilman and Smith (2010). The latter simplifies complex sentences at the syntactic level and generates a question for each of the extracted clauses. As an example, we report below three versions obtained for the *Agent* definition in the DAMAGING frame:



| Approach<br>Task            | 2-STEP |      | 1-STEP      |
|-----------------------------|--------|------|-------------|
|                             | FD     | FER  |             |
| Accuracy                    | .900   | .687 | <b>.792</b> |
| Answers                     | 100    | 160  | 416         |
| Trusted                     | 100    | 100  | 84          |
| Untrusted                   | 21     | 36   | 217         |
| Time (h)                    | 102    | 69   | <b>130</b>  |
| Cost/question<br>(\$ cents) | 1.83   | 2.74 | 8.41        |

Table 2: Overview of the experimental results. FD stands for Frame Discrimination, FER for FEs Recognition

*Original*: The conscious entity, generally a person, that performs the intentional action that results in the damage to the Patient.

*Manually simplified*: This element describes the person that performs the intentional action resulting in the damage to another person or object.

*Automatic system*: What that performs the intentional action that results in the damage to the Patient?

Simplification was performed by a linguistic expert, and followed a set of straightforward guidelines, which can be summarized as follows:

- When the semantic type associated with the FE is a common concept (e.g. `Location`), replace the FE name with the semantic type.
- Make syntactically complex definitions as simple as possible.
- Avoid variability in FE definitions, try to make them homogeneous (e.g. they should all start with “This element describes...” or similar).
- Replace technical concepts such as `Artifact` or `Sentient` with common words such as `Object` and `Person` respectively.

Although these changes (especially the last item) may make FE definitions less precise from a lexicographic point of view (for instance, sentient entities are not necessarily persons), annotation became more intuitive and had a positive impact on the overall quality.

After few pilot annotations with the three types of FE definitions, we noticed that the simplified

one achieved a better accuracy and a lower number of untrusted annotators compared to the others. Therefore, we use the simplified definitions in both the 2-step and the 1-step approach (Section 3.4).

**Discussion** Table 2 provides an overview of the results we gathered. The total number of answers differs from the total number of trusted judgments, since the average value of questions per sentence amounts to 1.5.<sup>2</sup> First of all, we notice an increase in the number of untrusted judgments. This is caused by a generally low inter-worker agreement on gold sentences due to FE definitions, which still present a certain degree of complexity, even after simplification. We inspected the full reports sentence by sentence and observed a propagation of incorrect judgments when a sentence involves an unclear FE definition. As FE definitions may mutually include mentions of other FEs from the same frame, we believe this circularity generated confusion.

### 3.4 1-step approach

Having set the LU polysemy to 2, in our case a sentence  $S$  always contains a LU with 2 possible frames ( $f_1, f_2$ ), but only conveys one, e.g.  $f_1$ . We formulate the approach as follows.  $S$  is replicated in 2 data units ( $S_a, S_b$ ). Then,  $S_a$  is associated to the set  $E_1$  of  $f_1$  FE definitions, namely the correct ones for that sentence. Instead,  $S_b$  is associated to the set  $E_2$  of  $f_2$  FE definitions. We call  $S_b$  a *cross-frame* unit. Furthermore, we allow workers to select the `None` answer. In practice, we ask a total amount of  $|E_1 \cup E_2| + 2$  questions per sentence  $S$ . In this way, we let the frame directly emerge from the FEs. If workers correctly answer `None` to a FE definition  $d \in E_2$ , the probability that  $S$  evokes  $f_1$  increases.

**Modeling** Figure 1 displays a screenshot of the worker interface. The task is designed as per Section 3.3.2, but with major differences with respect to its content. This is better described by an example. The sentence `Karen threw her arms round my neck, spilling champagne everywhere` contains the LU `throw.v` evoking the frame `BODY_MOVEMENT`. However, `throw.v` is ambiguous and may also evoke `CAUSE_MOTION`. We ask to annotate both the `BODY_MOVEMENT` and the `CAUSE_MOTION`

<sup>2</sup>Cf. Section 3 for more details

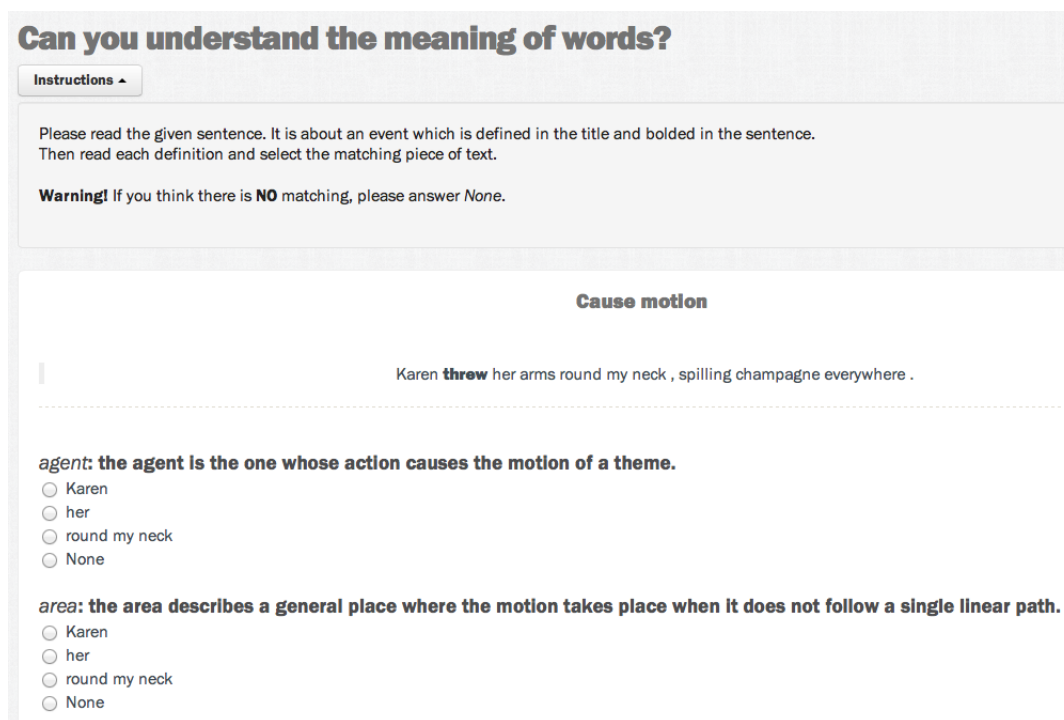


Figure 1: 1-step approach worker interface

core FEs, respectively as regular and cross-frame units.

**Discussion** We do not interpret the `None` choice as an abstention from judgment, since it is a correct answer for cross-frame units. Instead of precision and recall, we are thus able to directly compute workers' accuracy upon a majority vote. We envision an improvement with respect to the 2-step methodology, as we avoid the proven risk of error propagation originating from wrongly annotated frames in the first step. Table 2 illustrates the results we collected. As expected, accuracy reached a consistent enhancement. This demonstrates the hypothesis we stated in Section 1 on the cognitive plausibility of a bottom-up approach for frame annotation. Furthermore, the execution time decreases compared to the sum of the 2 steps, namely 130 hours against 171. Nevertheless, the cost is sensibly higher due to the higher number of questions that need to be addressed, in average 4.6 against 1.5. Untrusted judgments seriously grow, mainly because of the cross-frame gold complexity. Workers seem puzzled by the presence of `None`, which is a required answer for such units. If we consider the English FrameNet annotation agreement values between experts reported by Padó and Lapata (2009) as the upper bound (i.e., .897 for frame discrimination and .949

for FEs recognition), we believe our experimental setting can be reused as a valid alternative.

## 4 Conclusion

In this work, we presented an approach to perform frame annotation with crowdsourcing techniques, based on a single annotation step and on manually simplified FE definitions. Since the results seem promising, we are currently running larger scale experiments with the full set of FrameNet 1.5 annotated sentences. Input data, interface screenshots and full results are available and regularly updated at <http://db.tt/gu2Mj98i>.

Future work will include the investigation of a frame assignment strategy. In fact, we do not take into account the case of conflicting FE annotations in cross-frame units. Hence, we need a confidence score to determine which frame emerges if workers selected contradictory answers in a subset of cross-frame FE definitions.

## Acknowledgements

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

## References

- [Baker2012] Collin F Baker. 2012. Framenet, current collaborations and future goals. *Language Resources and Evaluation*, pages 1–18.
- [Burchardt et al.2006] Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Pado, and Manfred Pinkal. 2006. Salto—a versatile multi-level annotation tool. In *Proceedings of LREC 2006*, pages 517–520. Citeseer.
- [Chamberlain et al.2008] Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2008. Phrase detectors: A web-based collaborative annotation game. *Proceedings of I-Semantics, Graz*.
- [Chamberlain et al.2009] Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. 2009. Constructing an anaphorically annotated corpus with non-experts: Assessing the quality of collaborative annotations. In *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 57–62. Association for Computational Linguistics.
- [Fillmore et al.2002] Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. 2002. The FrameNet Database and Software Tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1157–1160, Las Palmas, Spain.
- [Fillmore1976] Charles J. Fillmore. 1976. Frame Semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language*, pages 20–32. Blackwell Publishing.
- [Heilman and Smith2010] Michael Heilman and Noah A. Smith. 2010. Extracting Simplified Statements for Factual Question Generation. In *Proceedings of QG2010: The Third Workshop on Question Generation*, Pittsburgh, PA, USA.
- [Hong and Baker2011] Jisup Hong and Collin F Baker. 2011. How good is the crowd at “real” wsd? *ACL HLT 2011*, page 30.
- [Negri et al.2011] Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ’11*, pages 670–679, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Padó and Lapata2009] Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36(1):307–340.
- [Palmer et al.2005] Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics*, 31(1).
- [Ruppenhofer et al.2006] Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. Available at <http://framenet.icsi.berkeley.edu/book/book.html>.
- [Snow et al.2008] Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.
- [Von Ahn et al.2006] Luis Von Ahn, Mihir Kedia, and Manuel Blum. 2006. Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78. ACM.

# Smatch: an Evaluation Metric for Semantic Feature Structures

**Shu Cai**

USC Information Sciences Institute  
4676 Admiralty Way, Suite 1001  
Marina del Rey, CA 90292  
shuca@isi.edu

**Kevin Knight**

USC Information Sciences Institute  
4676 Admiralty Way, Suite 1001  
Marina del Rey, CA 90292  
knight@isi.edu

## Abstract

The evaluation of whole-sentence semantic structures plays an important role in semantic parsing and large-scale semantic structure annotation. However, there is no widely-used metric to evaluate whole-sentence semantic structures. In this paper, we present *smatch*, a metric that calculates the degree of overlap between two semantic feature structures. We give an efficient algorithm to compute the metric and show the results of an inter-annotator agreement study.

## 1 Introduction

The goal of semantic parsing is to generate all semantic relationships in a text. Its output is often represented by whole-sentence semantic structures. Evaluating such structures is necessary for semantic parsing tasks, as well as semantic annotation tasks which create linguistic resources for semantic parsing.

However, there is no widely-used evaluation method for whole-sentence semantic structures. Current whole-sentence semantic parsing is mainly evaluated in two ways: 1. task correctness (Tang and Mooney, 2001), which evaluates on an NLP task that uses the parsing results; 2. whole-sentence accuracy (Zettlemoyer and Collins, 2005), which counts the number of sentences parsed completely correctly.

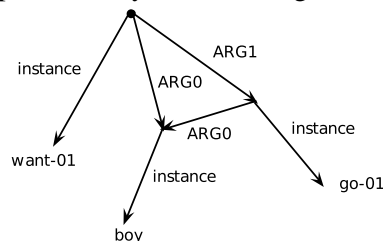
Nevertheless, it is worthwhile to explore evaluation methods that use scores which range from 0 to 1 (“partial credit”) to measure whole-sentence semantic structures. By using such methods, we are able to differentiate between two similar whole-sentence semantic structures regardless of specific

tasks or domains. In this work, we provide an evaluation metric that uses the degree of overlap between two whole-sentence semantic structures as the partial credit.

In this paper, we observe that the difficulty of computing the degree of overlap between two whole-sentence semantic feature structures comes from determining an optimal variable alignment between them, and further prove that finding such alignment is NP-complete. We investigate how to compute this metric and provide several practical and replicable computing methods by using Integer Linear Programming (ILP) and hill-climbing method. We show that our metric can be used for measuring the annotator agreement in large-scale linguistic annotation, and evaluating semantic parsing.

## 2 Semantic Overlap

We work on a semantic feature structure representation in a standard neo-Davidsonian (Davidson, 1969; Parsons, 1990) framework. For example, semantics of the sentence “the boy wants to go” is represented by the following directed graph:



In this graph, there are three concepts: want-01, boy, and go-01. Both want-01 and go-01 are frames from PropBank framesets (Kingsbury and Palmer, 2002). The frame want-01 has two arguments connected with ARG0 and ARG1, and go-01 has an argument (which is also the same boy instance) connected with ARG0.

Following (Langkilde and Knight, 1998) and (Langkilde-Geary, 2002), we refer to this semantic representation as AMR (Abstract Meaning Representation).

Semantic relationships encoded in the AMR graph can also be viewed as a conjunction of logical propositions, or triples:

```
instance(a, want-01)  ^
instance(b, boy)     ^
instance(c, go-01)   ^
ARG0(a, b)          ^
ARG1(a, c)          ^
ARG0(c, b)
```

Each AMR triple takes one of these forms: *relation(variable, concept)* (the first three triples above), or *relation(variable1, variable2)* (the last three triples above).

Suppose we take a second AMR (for “the boy wants the football”) and its associated propositional triples:

```
instance(x, want-01)  ^
instance(y, boy)     ^
instance(z, football) ^
ARG0(x, y)          ^
ARG1(x, z)
```

Our evaluation metric measures precision, recall, and f-score of the triples in the second AMR against the triples in the first AMR, i.e., the amount of propositional overlap.

The difficulty is that variable names are not shared between the two AMRs, so there are multiple ways to compute the propositional overlap based on different variable mappings. We therefore define the *smatch* score (for semantic match) as the *maximum f-score obtainable via a one-to-one matching of variables between the two AMRs*.

In the example above, there are six ways to match up variables between the two AMRs:

|                | M | P   | R   | F    |
|----------------|---|-----|-----|------|
| x=a, y=b, z=c: | 4 | 4/5 | 4/6 | 0.73 |
| x=a, y=c, z=b: | 1 | 1/5 | 1/6 | 0.18 |
| x=b, y=a, z=c: | 0 | 0/5 | 0/6 | 0.00 |
| x=b, y=c, z=a: | 0 | 0/5 | 0/6 | 0.00 |
| x=c, y=a, z=b: | 0 | 0/5 | 0/6 | 0.00 |
| x=c, y=b, z=a: | 2 | 2/5 | 2/6 | 0.36 |
| -----          |   |     |     |      |
| smatch score:  |   |     |     | 0.73 |

Here, M is the number of propositional triples that agree given a variable mapping, P is the precision

of the second AMR against the first, R is its recall, and F is its f-score. The smatch score is the maximum of the f-scores.

However, for AMRs that contain large number of variables, it is not efficient to get the f-score by simply using the method above. Exhaustively enumerating all variable mappings requires computing the f-score for  $n!/(n-m)!$  variable mappings (assuming one AMR has  $n$  variables and the other has  $m$  variables, and  $m \leq n$ ). This algorithm is too slow for all but the shortest AMR pairs.

### 3 Computing the Metric

This section describes how to compute the smatch score. As input, we are given AMR1 (with  $m$  variables) and AMR2 (with  $n$  variables). Without loss of generality,  $m \leq n$ .

**Baseline.** Our baseline first matches variables that share concepts. For example, it would match  $a$  in the first AMR example with  $x$  in the second AMR example of Section 2, because both are instances of *want-01*. If there are two or more variables to choose from, we pick the first available one. The rest of the variables are mapped randomly.

**ILP method.** We can get an optimal solution using integer linear programming (ILP). We create two types of variables:

- (Variable mapping)  $v_{ij} = 1$  iff the  $i$ th variable in AMR1 is mapped to the  $j$ th variable in AMR2 (otherwise  $v_{ij} = 0$ )
- (Triple match)  $t_{kl} = 1$  iff AMR1 triple  $k$  matches AMR2 triple  $l$ , otherwise  $t_{kl} = 0$ . A triple  $relation1(xy)$  matches  $relation2(wz)$  iff  $relation1 = relation2$ ,  $v_{xw} = 1$ , and  $v_{yz} = 1$  or  $y$  and  $z$  are the same concept.

Our constraints ensure a one-to-one mapping of variables, and they ensure that the chosen  $t$  values are consistent with the chosen  $v$  values:

$$\text{For all } i, \sum_j v_{ij} \leq 1$$

$$\text{For all } j, \sum_i v_{ij} \leq 1$$

For all triple pairs  $r(xy)r(wz)$  ( $r$  for relation),

$$t_{r(xy)r(wz)} \leq v_{xw}$$

$$t_{r(xy)r(wz)} \leq v_{yz}$$

when  $y$  and  $z$  are variables.

Finally, we ask the ILP solver to maximize:

$$\sum_{kl} t_{kl}$$

which denotes the maximum number of matching triples which lead to the smatch score.

**Hill-climbing method.** Finally, we develop a portable heuristic algorithm that does not require an ILP solver<sup>1</sup>. This method works in a greedy style. We begin with  $m$  random one-to-one mappings between the  $m$  variables of AMR1 and the  $n$  variables of AMR2. Each variable mapping is a pair  $(i, \text{map}(i))$  with  $1 \leq i \leq m$  and  $1 \leq \text{map}(i) \leq n$ . We refer to the  $m$  mappings as a variable mapping state.

We first generate a random initial variable mapping state, compute its triple match number, then hill-climb via two types of small changes:

1. Move one of the  $m$  mappings to a currently-unmapped variable from the  $n$ .
2. Swap two of the  $m$  mappings.

Any variable mapping state has  $m(n - m) + m(m - 1) = m(n - 1)$  neighbors during the hill-climbing search. We greedily choose the best neighbor, repeating until no neighbor improves the number of triple matches.

We experiment with two modifications to the greedy search: (1) executing multiple random restarts to avoid local optima, and (2) using our Baseline concept matching (“smart initialization”) instead of random initialization.

**NP-completeness.** There is unlikely to be an exact polynomial-time algorithm for computing smatch. We can reduce the 0-1 Maximum Quadratic Assignment Problem (0-1-Max-QAP) (Nagarajan and Sviridenko, 2009) and the subgraph isomorphism problem directly to the full smatch problem on graphs.<sup>2</sup>

We note that other widely-used metrics, such as TER (Snover et al., 2006), are also NP-complete. Fortunately, the next section shows that the smatch methods above are efficient and effective.

<sup>1</sup>The tool can be downloaded at <http://amr.isi.edu/evaluation.html>.

<sup>2</sup>Thanks to David Chiang for observing the subgraph isomorphism reduction.

## 4 Using Smatch

We report an AMR inter-annotator agreement study using smatch.

1. Our study has 4 annotators (A, B, C, D), who then converge on a consensus annotation E. We thus have 10 pairs of annotations: A-B, A-C, . . . , D-E.
2. The study is carried out 5 times. Each time annotators build AMRs for 4 sentences from the Wall Street Journal corpus. Sentence lengths range from 12 to 54 words, and AMRs range from 6 to 29 variables.
3. We use 7 smatch calculation methods in our experiments:
  - Base: Baseline matching method
  - ILP: Integer Linear Programming
  - R: Hill-climbing with random initialization
  - 10R: Hill-climbing with random initialization plus 9 random restarts
  - S: Hill-climbing with smart initialization
  - S+4R: Hill-climbing with smart initialization plus 4 random restarts
  - S+9R: Hill-climbing with smart initialization plus 9 random restarts

Table 1 shows smatch scores provided by the methods. Columns labeled 1-5 indicate sentence groups. Each individual smatch score is a document-level score of 4 AMR pairs.<sup>3</sup> ILP scores are optimal, so lower scores (in bold) indicate search errors.

Table 2 summarizes search accuracy as a percentage of smatch scores that equal that of ILP. Results show that the restarts are essential for hill-climbing, and that 9 restarts are sufficient to obtain good quality. The table also shows total runtimes over 200 AMR pairs (10 annotator pairs, 5 sentence groups, 4 AMR pairs per group). Heuristic search with smart initialization and 4 restarts (S+4R) gives the best trade-off between accuracy and speed, so this is the setting we use in practice.

Figure 1 shows smatch scores of each annotator (A-D) against the consensus annotation (E). The

<sup>3</sup>For documents containing multiple AMRs, we use the sum of matched triples over all AMR pairs to compute precision, recall, and f-score, much like corpus-level Bleu (Papineni et al., 2002).

|       | B           |             |      |             |             | C           |             |             |             |             | D           |             |      |             |             | E           |             |      |             |             |
|-------|-------------|-------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|-------------|-------------|-------------|-------------|------|-------------|-------------|
|       | 1           | 2           | 3    | 4           | 5           | 1           | 2           | 3           | 4           | 5           | 1           | 2           | 3    | 4           | 5           | 1           | 2           | 3    | 4           | 5           |
| Base  | <b>0.68</b> | <b>0.74</b> | 0.84 | <b>0.71</b> | <b>0.83</b> | <b>0.69</b> | <b>0.70</b> | 0.80        | <b>0.69</b> | <b>0.78</b> | <b>0.77</b> | <b>0.72</b> | 0.75 | <b>0.68</b> | <b>0.63</b> | <b>0.79</b> | <b>0.86</b> | 0.92 | <b>0.85</b> | <b>0.89</b> |
| ILP   | 0.74        | 0.80        | 0.84 | 0.76        | 0.88        | 0.75        | 0.78        | 0.80        | 0.77        | 0.88        | 0.83        | 0.77        | 0.75 | 0.72        | 0.76        | 0.85        | 0.92        | 0.92 | 0.89        | 0.92        |
| R     | 0.74        | <b>0.79</b> | 0.84 | <b>0.75</b> | <b>0.86</b> | <b>0.74</b> | <b>0.75</b> | 0.80        | 0.77        | 0.88        | <b>0.83</b> | <b>0.76</b> | 0.75 | 0.72        | <b>0.75</b> | 0.85        | 0.92        | 0.92 | <b>0.89</b> | <b>0.89</b> |
| A 10R | 0.74        | 0.80        | 0.84 | 0.76        | 0.88        | 0.75        | 0.78        | 0.80        | 0.77        | 0.88        | 0.83        | 0.77        | 0.75 | 0.72        | 0.76        | 0.85        | 0.92        | 0.92 | 0.89        | 0.92        |
| S     | 0.74        | 0.80        | 0.84 | <b>0.75</b> | 0.88        | 0.75        | 0.78        | 0.80        | <b>0.76</b> | 0.88        | 0.83        | 0.77        | 0.75 | 0.72        | 0.76        | 0.85        | 0.92        | 0.92 | 0.89        | 0.92        |
| S+4R  | 0.74        | 0.80        | 0.84 | 0.76        | 0.88        | 0.75        | 0.78        | 0.80        | 0.77        | 0.88        | 0.83        | 0.77        | 0.75 | 0.72        | 0.76        | 0.85        | 0.92        | 0.92 | 0.89        | 0.92        |
| S+9R  | 0.74        | 0.80        | 0.84 | 0.76        | 0.88        | 0.75        | 0.78        | 0.80        | 0.77        | 0.88        | 0.83        | 0.77        | 0.75 | 0.72        | 0.76        | 0.85        | 0.92        | 0.92 | 0.89        | 0.92        |
| Base  | -           | -           | -    | -           | -           | <b>0.72</b> | <b>0.68</b> | 0.74        | <b>0.69</b> | <b>0.79</b> | <b>0.71</b> | <b>0.72</b> | 0.76 | <b>0.65</b> | <b>0.57</b> | <b>0.68</b> | <b>0.71</b> | 0.83 | <b>0.79</b> | <b>0.86</b> |
| ILP   | -           | -           | -    | -           | -           | 0.74        | 0.83        | 0.74        | 0.75        | 0.85        | 0.78        | 0.83        | 0.76 | 0.68        | 0.73        | 0.76        | 0.81        | 0.83 | 0.83        | 0.89        |
| R     | -           | -           | -    | -           | -           | 0.74        | 0.83        | <b>0.72</b> | <b>0.72</b> | <b>0.83</b> | 0.78        | 0.83        | 0.76 | 0.68        | <b>0.68</b> | <b>0.74</b> | 0.81        | 0.83 | 0.83        | 0.89        |
| B 10R | -           | -           | -    | -           | -           | 0.74        | 0.83        | 0.74        | 0.75        | 0.85        | 0.78        | 0.83        | 0.76 | 0.68        | 0.73        | 0.76        | 0.81        | 0.83 | 0.83        | 0.89        |
| S     | -           | -           | -    | -           | -           | <b>0.73</b> | 0.83        | 0.74        | 0.75        | 0.85        | 0.78        | 0.83        | 0.76 | 0.68        | 0.73        | 0.76        | 0.81        | 0.83 | 0.83        | 0.89        |
| S+4R  | -           | -           | -    | -           | -           | 0.74        | 0.83        | 0.74        | 0.75        | 0.85        | 0.78        | 0.83        | 0.76 | 0.68        | 0.73        | 0.76        | 0.81        | 0.83 | 0.83        | 0.89        |
| S+9R  | -           | -           | -    | -           | -           | 0.74        | 0.83        | 0.74        | 0.75        | 0.85        | 0.78        | 0.83        | 0.76 | 0.68        | 0.73        | 0.76        | 0.81        | 0.83 | 0.83        | 0.89        |
| Base  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | <b>0.68</b> | <b>0.68</b> | 0.74 | <b>0.69</b> | <b>0.65</b> | <b>0.64</b> | <b>0.64</b> | 0.87 | <b>0.79</b> | <b>0.83</b> |
| ILP   | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | 0.78        | 0.81        | 0.74        | 0.76        | 0.87 | 0.85        | 0.89        |
| R     | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | <b>0.75</b> | <b>0.78</b> | <b>0.71</b> | 0.76        | 0.87 | 0.85        | 0.89        |
| C 10R | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | 0.78        | 0.81        | 0.74        | 0.76        | 0.87 | 0.85        | 0.89        |
| S     | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | <b>0.77</b> | 0.81        | 0.74        | 0.76        | 0.87 | 0.85        | 0.89        |
| S+4R  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | 0.78        | 0.81        | 0.74        | 0.76        | 0.87 | 0.85        | 0.89        |
| S+9R  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | 0.74        | 0.79        | 0.74 | 0.78        | 0.81        | 0.74        | 0.76        | 0.87 | 0.85        | 0.89        |
| Base  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | <b>0.68</b> | <b>0.69</b> | 0.81 | <b>0.74</b> | <b>0.64</b> |
| ILP   | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | 0.78        | 0.81 | 0.78        | 0.79        |
| R     | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | <b>0.73</b> | 0.81 | 0.78        | 0.79        |
| D 10R | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | 0.78        | 0.81 | 0.78        | 0.79        |
| S     | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | 0.78        | 0.81 | 0.78        | 0.79        |
| S+4R  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | 0.78        | 0.81 | 0.78        | 0.79        |
| S+9R  | -           | -           | -    | -           | -           | -           | -           | -           | -           | -           | -           | -           | -    | -           | -           | 0.77        | 0.78        | 0.81 | 0.78        | 0.79        |

Table 1: Inter-annotator smatch agreement for 5 groups of sentences, as computed with seven different methods (Base, ILP, R, 10R, S, S+4R, S+9R). The number 1-5 indicate the sentence group number. Bold scores are search errors.

|            | Base | ILP   | R     | 10R   | S    | S+4R  | S+9R  |
|------------|------|-------|-------|-------|------|-------|-------|
| Accuracy   | 20%  | 100%  | 66.5% | 100%  | 92%  | 100%  | 100%  |
| Time (sec) | 0.86 | 49.67 | 5.85  | 64.78 | 2.31 | 28.36 | 59.69 |

Table 2: Accuracy and running time (seconds) of various computing methods of smatch over 200 AMR pairs.

plot demonstrates that, as time goes by, annotators reach better agreement with the consensus.

We also note that smatch is used to measure the accuracy of machine-generated AMRs. (Jones et al., 2012) use it to evaluate automatic semantic parsing in a narrow domain, while Ulf Hermjakob<sup>4</sup> has developed a heuristic algorithm that exploits and supplements Ontonotes annotations (Pradhan et al., 2007) in order to automatically create AMRs for Ontonotes sentences, with a smatch score of 0.74 against human consensus AMRs.

## 5 Related Work

Related work on directly measuring the semantic representation includes the method in (Dridan and Oepen, 2011), which evaluates semantic parser output directly by comparing semantic substructures, though they require an alignment between sentence spans and semantic sub-structures. In contrast, our metric does not require the align-

<sup>4</sup>personal communication

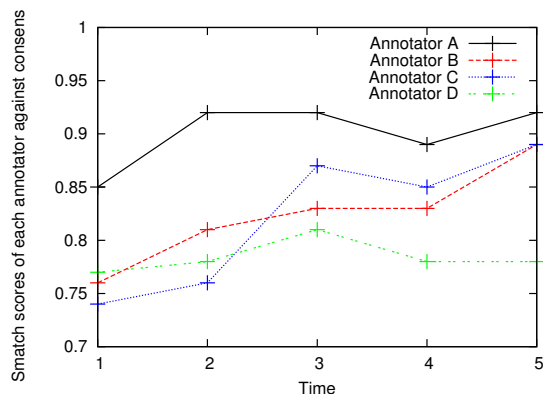


Figure 1: Smatch scores of annotators (A-D) against the consensus annotation (E) over time.

ment between an input sentence and its semantic analysis. (Allen et al., 2008) propose a metric which computes the maximum score by any alignment between LF graphs, but they do not address how to determine the alignments.

## 6 Conclusion and Future Work

We present an evaluation metric for whole-sentence semantic analysis, and show that it can be computed efficiently. We use the metric to measure semantic annotation agreement rates and parsing accuracy. In the future, we plan to investigate how to adapt smatch to other semantic representations.

## 7 Acknowledgements

We would like to thank David Chiang, Hui Zhang, other ISI colleagues and our anonymous reviewers for their thoughtful comments. This work was supported by NSF grant IIS-0908532.

## References

- J.F. Allen, M. Swift, and W. Beaumont. 2008. Deep Semantic Analysis of Text. In *Proceedings of the 2008 Conference on Semantics in Text Processing*.
- D. Davidson. 1969. The Individuation of Events. In *Nicholas Rescher (ed.) Essays in Honor of Carl G. Hempel Dordrecht: D. Reidel*.
- R. Dridan and S. Oepen. 2011. Parser Evaluation using Elementary Dependency Matching. In *Proceedings of the 12th International Conference on Parsing Technologies*.
- B. Jones, J. Andreas, D. Bauer, K. M. Hermann, and K. Knight. 2012. Semantics-Based Machine Translation with Hyperedge Replacement Grammars. In *Proceedings of COLING*.
- P. Kingsbury and M. Palmer. 2002. From Treebank to Propbank. In *Proceedings of LREC*.
- I. Langkilde and K. Knight. 1998. Generation that Exploits Corpus-based Statistical Knowledge. In *Proceedings of COLING-ACL*.
- I. Langkilde-Geary. 2002. An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator. In *Proceedings of International Natural Language Generation Conference (INLG'02)*.
- V. Nagarajan and M. Sviridenko. 2009. On the Maximum Quadratic Assignment Problem. *Mathematics of Operations Research*, 34.
- K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.
- T. Parsons. 1990. *Events in the Semantics of English*. The MIT Press.
- S. S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2007. Ontonotes: A Unified Relational Semantic Representation. In *Proceedings of the International Conference on Semantic Computing (ICSC '07)*.
- M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA-2006)*.
- L. R. Tang and R. J. Mooney. 2001. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In *Proceedings of the 12th European Conference on Machine Learning*.
- L. S. Zettlemoyer and M. Collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorical Grammars. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*.



# Variable Bit Quantisation for LSH

**Sean Moran**

School of Informatics  
The University of Edinburgh  
EH8 9AB, Edinburgh, UK  
sean.moran@ed.ac.uk

**Victor Lavrenko**

School of Informatics  
The University of Edinburgh  
EH8 9AB, Edinburgh, UK  
vlavrenk@inf.ed.ac.uk

**Miles Osborne**

School of Informatics  
The University of Edinburgh  
EH8 9AB, Edinburgh, UK  
miles@inf.ed.ac.uk

## Abstract

We introduce a scheme for optimally allocating a variable number of bits per LSH hyperplane. Previous approaches assign a constant number of bits per hyperplane. This neglects the fact that a subset of hyperplanes may be more informative than others. Our method, dubbed Variable Bit Quantisation (VBQ), provides a data-driven non-uniform bit allocation across hyperplanes. Despite only using a fraction of the available hyperplanes, VBQ outperforms uniform quantisation by up to 168% for retrieval across standard text and image datasets.

## 1 Introduction

The task of retrieving the nearest neighbours to a given query document permeates the field of Natural Language Processing (NLP). Nearest neighbour search has been used for applications as diverse as automatically detecting document translation pairs for the purposes of training a statistical machine translation system (SMT) (Krstovski and Smith, 2011), the large-scale generation of noun similarity lists (Ravichandran et al., 2005) to an unsupervised method for extracting domain specific lexical variants (Stephan Gouws and Metzle, 2011).

There are two broad approaches to nearest neighbour based search: *exact* and *approximate* techniques, which are differentiated by their ability to return completely correct nearest neighbours (the exact approach) or have some possibility of returning points that are not true nearest neighbours (the approximate approach). Approximate nearest neighbour (ANN) search using hashing techniques has recently gained prominence within NLP. The hashing-based approach maps the data into a substantially more compact representation

referred to as a *fingerprint*, that is more efficient for performing similarity computations. The resulting compact binary representation radically reduces memory requirements while also permitting fast sub-linear time retrieval of approximate nearest neighbours.

Hashing-based ANN techniques generally comprise two main steps: a *projection* stage followed by a *quantisation* stage. The projection stage performs a neighbourhood preserving embedding, mapping the input data into a lower-dimensional representation. The quantisation stage subsequently reduces the cardinality of this representation by converting the real-valued projections to binary. Quantisation is a lossy transformation which can have a significant impact on the resulting quality of the binary encoding.

Previous work has quantised each projected dimension into a uniform number of bits (Indyk and Motwani, 1998) (Kong and Li, 2012) (Kong et al., 2012) (Moran et al., 2013). We demonstrate that uniform allocation of bits is sub-optimal and propose a data-driven scheme for variable bit allocation. Our approach is distinct from previous work in that it provides a general objective function for bit allocation. VBQ makes no assumptions on the data and, in addition to LSH, it applies to a broad range of other projection functions.

## 2 Related Work

Locality sensitive hashing (LSH) (Indyk and Motwani, 1998) is an example of an approximate nearest neighbour search technique that has been widely used within the field of NLP to preserve the Cosine distances between documents (Charikar, 2002). LSH for cosine distance draws a large number of random hyperplanes within the input feature space, effectively dividing the space into non-overlapping regions (or buckets). Each hyperplane contributes one bit to the encoding, the value (0 or 1) of which is determined by comput-

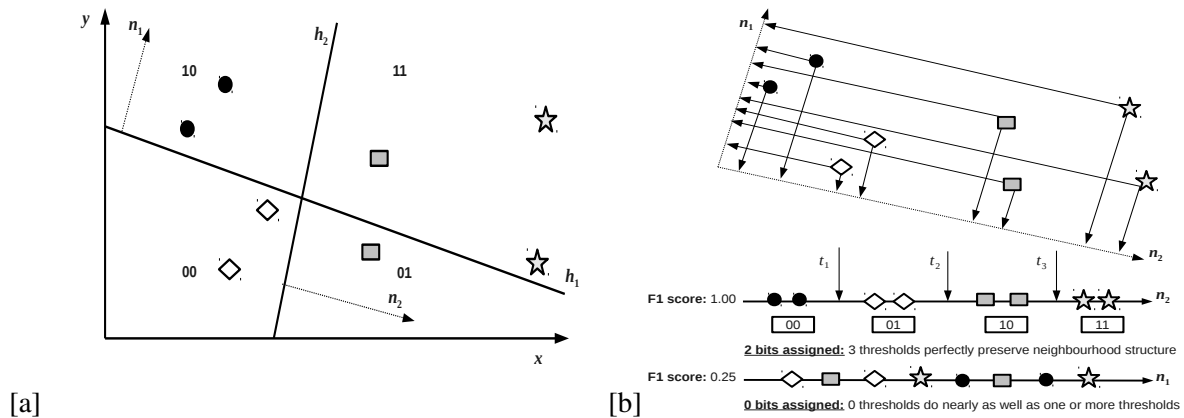


Figure 1: **Left:** Data points with identical shapes are 1-NN. Two hyperplanes  $h_1$ ,  $h_2$  are shown alongside their associated normal vectors ( $n_1$ ,  $n_2$ ). **Right top:** Projection of points onto the normal vectors  $n_1$  and  $n_2$  of the hyperplanes (arrows denote projections). **Right middle:** Positioning of the points along normal vector  $n_2$ . Three quantisation thresholds ( $t_1$ ,  $t_2$ ,  $t_3$ , and consequently 2 bits) can maintain the neighbourhood structure. **Right bottom:** the high degree of mixing between the 1-NN means that this hyperplane ( $h_1$ ) is likely to have 0 bits assigned (and therefore be discarded entirely).

ing the dot product of a data-point ( $\mathbf{x}$ ) with the normal vector to the hyperplane ( $\mathbf{n}_i$ ): that is, if  $\mathbf{x} \cdot \mathbf{n}_i < 0$ ,  $i \in \{1 \dots k\}$ , then the  $i$ -th bit is set to 0, and 1 otherwise. This encoding scheme is known as single bit quantisation (SBQ). More recent hashing work has sought to inject a degree of data-dependency into the positioning of the hyperplanes, for example, by using the principal directions of the data (Wang et al., 2012) (Weiss et al., 2008) or by training a stack of restricted Boltzmann machines (Salakhutdinov and Hinton, 2009).

Existing quantisation schemes for LSH allocate either one bit per hyperplane (Indyk and Motwani, 1998) or multiple bits per hyperplane (Kong et al., 2012) (Kong and Li, 2012) (Moran et al., 2013). For example, (Kong et al., 2012) recently proposed the Manhattan Hashing (MQ) quantisation technique where each projected dimension is encoded with multiple bits of natural binary code (NBC). The Manhattan distance between the NBC encoded data points is then used for nearest neighbour search. The authors demonstrated that MQ could better preserve the neighbourhood structure between the data points as compared to SBQ with Hamming distance.

Other recent quantisation work has focused on the setting of the quantisation thresholds: for example (Kong and Li, 2012) suggested encoding each dimension into two bits and using an adaptive thresholding scheme to set the threshold positions. Their technique dubbed, Double Bit Quantisation

(DBQ), attempts to avoid placing thresholds between data points with similar projected values. In other work (Moran et al., 2013) demonstrated that retrieval accuracy could be enhanced by using a topological quantisation matrix to guide the quantisation threshold placement along the projected dimensions. This topological quantisation matrix specified pairs of  $\epsilon$ -nearest neighbours in the original feature space. Their approach, Neighbourhood Preserving Quantisation (NPQ), was shown to achieve significant increases in retrieval accuracy over SBQ, MQ and DBQ for the task of image retrieval. In all of these cases the bit allocation is *uniform*: each hyperplane is assigned an identical number of bits.

### 3 Variable Bit Quantisation

Our proposed quantisation scheme, Variable Bit Quantisation (VBQ), assigns a variable number of bits to each hyperplane subject to a maximum upper limit on the total number of bits<sup>1</sup>. To do so, VBQ computes an F-measure based directly on the positioning of the quantisation thresholds along a projected dimension. The higher the F-measure for a given hyperplane, the better that hyperplane is at preserving the neighbourhood structure between the data points, and the more bits the hyperplane should be afforded from the bit budget  $B$ .

Figure 1(a) illustrates the original 2-dimensional feature space for a toy example.

<sup>1</sup>Referred to as the bit budget  $B$ , typically 32 or 64 bits.

The space is divided into 4 buckets by two random LSH hyperplanes. The circles, diamonds, squares and stars denote 1-nearest neighbours (1-NN). Quantisation for LSH is performed by projecting the data points onto the normal vectors ( $\mathbf{n}_1, \mathbf{n}_2$ ) to the hyperplanes ( $\mathbf{h}_1, \mathbf{h}_2$ ). This leads to two *projected dimensions*. Thresholding these projected dimensions at zero, and determining which side of zero a given data-point falls, yields the bit encoding for a given data-point.

Figure 1(b) demonstrates our proposed quantisation scheme. Similar to vanilla LSH, the data-points are projected onto the normal vectors, to yield two projected dimensions. This is illustrated on the topmost diagram in Figure 1(b). VBQ differs in how these projected dimensions are thresholded to yield the bit encoding: rather than one threshold situated at zero, VBQ employs one or more thresholds and positions these thresholds in an adaptive manner based upon maximisation of an F-measure. Using multiple thresholds enables more than one bit to be assigned per hyperplane<sup>2</sup>.

Figure 1(b) (middle, bottom) depicts the F-measure driven threshold optimisation along the projected dimensions. We define as a *positive pair*, those pairs of data points in the original feature space that are  $\epsilon$ -nearest neighbours ( $\epsilon$ -NN), and a *negative pair* otherwise. In our toy example, data points with the same shape symbol form a positive pair, while points with different symbols are negative pairs. Intuitively, the thresholds should be positioned in such a way as to maximize the number of positive pairs that fall within the same thresholded region, while also ensuring the negative pairs fall into different regions.

This intuition can be captured by an F-measure which counts the number of positive pairs that are found within the same thresholded regions (true positives, TP), the number of negative pairs found within the same regions (false positives, FP), and the number of positive pairs found in different regions of the threshold partitioned dimension (false negatives, FN). For  $\mathbf{n}_2$ , three thresholds are optimal, given they perfectly preserve the neighbourhood structure. For  $\mathbf{n}_1$ , no thresholds can provide a neighbourhood preserving quantisation and therefore it is better to discard the hyperplane  $\mathbf{h}_1$ . VBQ uses random restarts to optimise the F-measure<sup>3</sup>.

The computed F-measure scores per hyper-

<sup>2</sup> $b$  bits, requires  $2^b - 1$  thresholds.

<sup>3</sup>More details on the computation of the F-measure per hyperplane can be found in (Moran et al., 2013).

plane ( $h$ ), per bit count ( $b$ ) are an effective signal for bit allocation: more informative hyperplanes tend to have higher F-measure, for higher bit counts. VBQ applies a binary integer linear program (BILP) on top of the F-measure scores to obtain the bit allocation. To do so, the algorithm collates the scores in a matrix  $\mathbf{F}$  with elements  $F_{b,h}$ , where  $b \in \{0, \dots, k\}$ <sup>4</sup> indexes the rows, with  $k$  being the maximum number of bits allowable for any given hyperplane (set to 4 in this work), and  $h \in \{1 \dots, B\}$  indexes the columns. The BILP uses  $\mathbf{F}$  to find the bit allocation that maximises the cumulative F-measure across the  $B$  hyperplanes (Equation 1).

$$\begin{aligned} & \max \quad \|\mathbf{F} \circ \mathbf{Z}\| \\ & \text{subject to} \quad \|\mathbf{Z}_h\| = 1 \quad h \in \{1 \dots B\} \\ & \quad \|\mathbf{Z} \circ \mathbf{D}\| \leq B \\ & \quad \mathbf{Z} \text{ is binary} \end{aligned} \quad (1)$$

$\|\cdot\|$  denotes the Frobenius  $L_1$  norm,  $\circ$  the Hadamard product and  $\mathbf{D}$  is a constraint matrix, with  $D_{b,h} = b$ , ensuring that the bit allocation remains within the bit budget  $B$ . The BILP is solved using the standard branch and bound optimization algorithm (Land and Doig, 1960). The output from the BILP is an indicator matrix  $\mathbf{Z} \in \{0, 1\}^{(k+1) \times B}$  whose columns specify the optimal bit allocation for a given hyperplane i.e.  $Z_{b,h} = 1$  if the BILP decided to allocate  $b$  bits for hyperplane  $h$ , and zero otherwise. Example matrices for the toy problem in Figure 1 are given hereunder (in this example,  $k = 2$  and  $B = 2$ ).

$$\begin{array}{ccc} \mathbf{F} & h_1 & h_2 & \mathbf{D} & \mathbf{Z} \\ b_0 & \begin{pmatrix} 0.25 & 0.25 \end{pmatrix} & & \begin{pmatrix} 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \end{pmatrix} \\ b_1 & \begin{pmatrix} 0.35 & 0.50 \end{pmatrix} & & \begin{pmatrix} 1 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \end{pmatrix} \\ b_2 & \begin{pmatrix} 0.40 & 1.00 \end{pmatrix} & & \begin{pmatrix} 2 & 2 \end{pmatrix} & \begin{pmatrix} 0 & 1 \end{pmatrix} \end{array}$$

Notice how the indicator matrix  $\mathbf{Z}$  specifies an assignment of 0 bits for hyperplane  $h_1$  and 2 bits for hyperplane  $h_2$  as this yields the highest cumulative F-measure across hyperplanes while also meeting the bit budget. VBQ is therefore a principled method to select a discriminative subset of hyperplanes, and simultaneously allocate bits to the remaining hyperplanes, given a fixed overall bit budget  $B$ , while maximizing cumulative F-measure.

<sup>4</sup>For 0 bits, we compute the F-measure without any thresholds along the projected dimension.

| Dataset | CIFAR-10 |       |       |       |              | TDT-2 |       |       |              | Reuters-21578 |       |       |              |
|---------|----------|-------|-------|-------|--------------|-------|-------|-------|--------------|---------------|-------|-------|--------------|
|         | SBQ      | MQ    | DBQ   | NPQ   | VBQ          | SBQ   | MQ    | DBQ   | VBQ          | SBQ           | MQ    | DBQ   | VBQ          |
| SIKH    | 0.042    | 0.063 | 0.047 | 0.090 | <b>0.161</b> | 0.034 | 0.045 | 0.031 | <b>0.092</b> | 0.102         | 0.112 | 0.087 | <b>0.389</b> |
| LSH     | 0.119    | 0.093 | 0.066 | 0.153 | <b>0.207</b> | 0.189 | 0.097 | 0.089 | <b>0.229</b> | 0.276         | 0.201 | 0.175 | <b>0.538</b> |
| BLSI    | 0.038    | 0.135 | 0.111 | 0.155 | <b>0.231</b> | 0.283 | 0.210 | 0.087 | <b>0.396</b> | 0.100         | 0.030 | 0.030 | <b>0.156</b> |
| SH      | 0.051    | 0.135 | 0.111 | 0.167 | <b>0.202</b> | 0.146 | 0.212 | 0.167 | <b>0.370</b> | 0.033         | 0.028 | 0.030 | <b>0.154</b> |
| PCAH    | 0.036    | 0.137 | 0.107 | 0.153 | <b>0.219</b> | 0.281 | 0.208 | 0.094 | <b>0.374</b> | 0.095         | 0.034 | 0.027 | <b>0.154</b> |

Table 1: Area under the Precision Recall curve (AUPRC) for all five projection methods. Results are for 32 bits (images) and at 128 bits (text). The best overall score for each dataset is shown in bold face.

## 4 Experiments

### 4.1 Datasets

Our text datasets are *Reuters-21578* and *TDT-2*. The original Reuters-21578 corpus contains 21578 documents in 135 categories. We use the *ModApte* version and discard those documents with multiple category labels. This leaves 8,293 documents in 65 categories. The corpus contains 18,933 distinct terms. The TDT-2 corpus consists of 11,201 on-topic documents which are classified into 96 semantic categories. We remove those documents appearing in two or more categories and keep only the largest 30 categories. This leaves 9,394 documents in total with 36,771 distinct terms. Both text datasets are TF-IDF and L2 norm weighted. To demonstrate the generality of VBQ we also evaluate on the *CIFAR-10* image dataset (Krizhevsky, 2009), which consists of 60,000 images represented as 512 dimensional Gist descriptors (Oliva and Torralba, 2001). All of the datasets are identical to those that have been used in previous ANN hashing work (Zhang et al., 2010) (Kong and Li, 2012) and are publicly available on the Internet.

### 4.2 Projection Methods

VBQ is independent of the projection stage and therefore can be used to quantise the projections from a wide range of different projection functions, including LSH. In our evaluation we take a sample of the more popular data-independent (LSH, SIKH) and data-dependent (SH, PCAH, BLSI) projection functions used in recent hashing work:

- **SIKH:** Shift-Invariant Kernel Hashing (SIKH) uses random projections that approximate shift invariant kernels (Raginsky and Lazebnik, 2009). We follow previous work and use a Gaussian kernel with a bandwidth set to the average distance to the 50th nearest

neighbour (Kong et al., 2012) (Raginsky and Lazebnik, 2009).

- **LSH:** Locality Sensitive Hashing uses a Gaussian random matrix for projection (Indyk and Motwani, 1998) (Charikar, 2002).
- **BLSI:** Binarised Latent Semantic Indexing (BLSI) forms projections through Singular Value Decomposition (SVD) (Salakhutdinov and Hinton, 2009).
- **SH:** Spectral Hashing (SH) uses the eigenfunctions computed along the principal component directions of the data for projection (Weiss et al., 2008).
- **PCAH:** Principal Component Analysis Hashing (PCAH) employs the eigenvectors corresponding to the largest eigenvalues of the covariance matrix for projection (Wang et al., 2012).

### 4.3 Baselines

Single Bit Quantisation (SBQ) (Indyk and Motwani, 1998), Manhattan Hashing (MQ) (Kong et al., 2012), Double Bit Quantisation (DBQ) (Kong and Li, 2012) and Neighbourhood Preserving Quantisation (NPQ) (Moran et al., 2013). MQ, DBQ and NPQ all assign 2 bits per hyperplane, while SBQ assigns 1 bit per hyperplane. All methods, including VBQ, are constrained to be within the allocated bit budget  $B$ . If a method assigns more bits to one hyperplane, then it either discards, or assigns less bits to other hyperplanes.

### 4.4 Evaluation Protocol

We adopt the standard Hamming ranking evaluation paradigm (Kong et al., 2012). We randomly select 1000 query data points per run. Our results are averaged over 10 runs, and the average reported. The  $\epsilon$ -neighbours of each query point

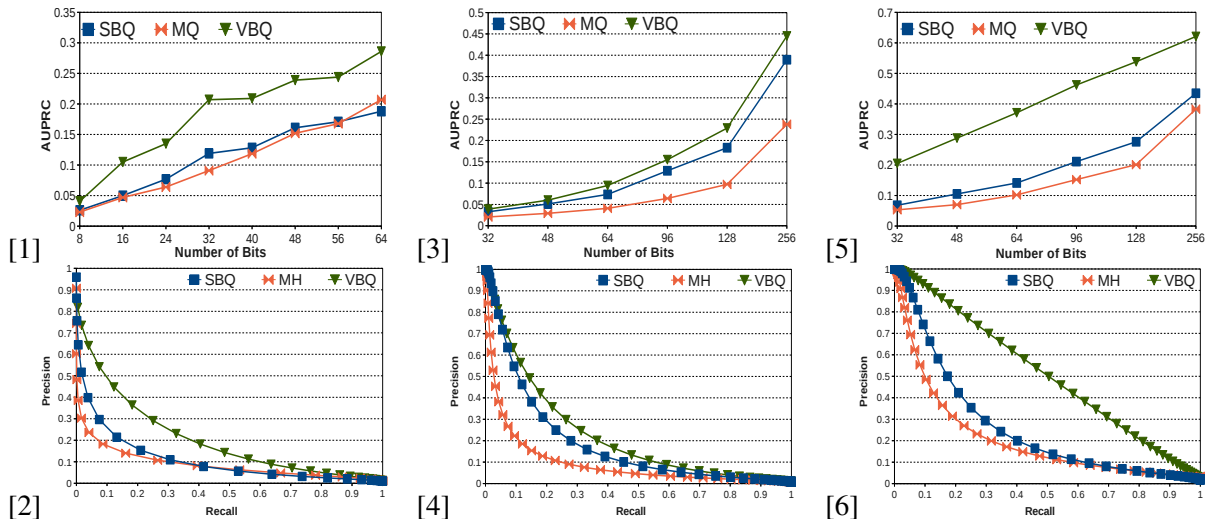


Figure 2: [1] LSH AUPRC vs bits for CIFAR-10 [2] LSH Precision-Recall curve for CIFAR-10 [3] LSH AUPRC vs bits for TDT-2 [4] LSH Precision-Recall curve for TDT-2 [5] LSH AUPRC vs bits for Reuters-21578 [6] LSH Precision-Recall curve for Reuters-21578

form the ground truth for evaluation. The threshold  $\epsilon$  is computed by sampling 100 training data-points at random from the training dataset and determining the distance at which these points have 50 nearest neighbours on average. Positive pairs and negative pairs for F-measure computation are computed by thresholding the *training dataset* Euclidean distance matrix by  $\epsilon$ . We adopt the Manhattan distance and multi-bit binary encoding method as suggested in (Kong et al., 2012). The F-measure we use for threshold optimisation is:  $F_\beta = (1 + \beta^2)TP / ((1 + \beta^2)TP + \beta^2FN + FP)$ . We select the parameter  $\beta$  on a *held-out validation* dataset. The area under the precision-recall curve (AUPRC) is used to evaluate the quality of retrieval.

#### 4.5 Results

Table 1 presents our results. For LSH on text (Reuters-21578) at 128 bits we find a substantial 95% gain in retrieval performance over uniformly assigning 1 bit per hyperplane (SBQ) and a 168% gain over uniformly assigning 2 bits per hyperplane (MQ). VBQ gain over SBQ at 128 bits is statistically significant based upon a paired Wilcoxon signed rank test across 10 random train/test partitions ( $p$ -value:  $\leq 0.0054$ ). This pattern is repeated on TDT-2 (for 128 bits, SBQ vs VBQ:  $p$ -value  $\leq 0.0054$ ) and CIFAR-10 (for 32 bits, SBQ vs VBQ:  $p$ -value:  $\leq 0.0054$ ). VBQ also reaps substantial gains for the Eigendecomposition based projections (PCA, SH, BLSI) effectively exploit-

ing the imbalanced variance across hyperplanes - that is, those hyperplanes capturing higher proportions of the variance in the data are allocated more bits from the fixed bit budget. Figure 2 (top row) illustrates that VBQ is effective across a range of bit budgets. Figure 2 (bottom row) presents the precision-recall (PR) curves at 32 bits (CIFAR-10) and 128 bits (TDT-2, Reuters-21578). We confirm our hypothesis that judicious allocation of variable bits is significantly more effective than uniform allocation.

## 5 Conclusions

Our proposed quantisation scheme computes a non-uniform bit assignment across LSH hyperplanes. The novelty of our approach is centred upon a binary integer linear program driven by a novel F-measure based objective function that determines the most appropriate bit allocation: hyperplanes that better preserve the neighbourhood structure of the input data points are awarded more bits from a fixed bit budget. Our evaluation on standard datasets demonstrated that VBQ can substantially enhance the retrieval accuracy of a selection of popular hashing techniques across two distinct modalities (text and images). In this paper we concentrated on the hamming ranking based scenario for hashing. In the future, we would like to examine the performance of VBQ in the lookup based hashing scenario where hash tables are used for fast retrieval.

## References

- Moses Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388.
- Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing, STOC '98*, pages 604–613, New York, NY, USA. ACM.
- Weihao Kong and Wu-Jun Li. 2012. Double-bit quantization for hashing. In *AAAI*.
- Weihao Kong, Wu-Jun Li, and Minyi Guo. 2012. Manhattan hashing for large-scale image retrieval. *SIGIR '12*, pages 45–54.
- Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. Master's thesis.
- Kriste Krstovski and David A. Smith. 2011. A Minimally Supervised Approach for Detecting and Ranking Document Translation Pairs. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland. Association for Computational Linguistics.
- A. H. Land and A. G. Doig. 1960. An automatic method of solving discrete programming problems. *Econometrica*, 28:pp. 497–520.
- Sean Moran, Victor Lavrenko, and Miles Osborne. 2013. Neighbourhood preserving quantisation for lsh. In *36th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Dublin, Ireland, 07/2013.
- Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.
- Maxim Raginsky and Svetlana Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS '09*, pages 1509–1517.
- Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized algorithms and nlp: using locality sensitive hash function for high speed noun clustering. *ACL '05*, pages 622–629. Association for Computational Linguistics.
- Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978.
- Dirk Hovy Stephan Gouws and Donald Metzle. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP, EMNLP '11*, page 8290. Association for Computational Linguistics.
- Jun Wang, S. Kumar, and Shih-Fu Chang. 2012. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2393–2406.
- Yair Weiss, Antonio B. Torralba, and Robert Fergus. 2008. Spectral hashing. In *NIPS*, pages 1753–1760.
- Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. 2010. Self-taught hashing for fast similarity search. In *SIGIR*, pages 18–25.

# Context Vector Disambiguation for Bilingual Lexicon Extraction from Comparable Corpora

**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 approach that extends the standard approach used for bilingual lexicon extraction from comparable corpora. We focus on the unresolved problem of *polysemous words* revealed by the bilingual dictionary and introduce a use of a Word Sense Disambiguation process that aims at improving the adequacy of context vectors. On two specialized French-English comparable corpora, empirical experimental results show that our method improves the results obtained by two state-of-the-art approaches.

## 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 basic assumption behind most studies is a *distributional* hypothesis (Harris, 1954), which states that words with a similar meaning are likely to appear in similar contexts across languages. The so-called *standard approach* to bilingual lexicon extraction from comparable corpora is based on the characterization and comparison of *context vectors* of source and target words. Each element in the context vector of a source or target word represents its association with a word which occurs within a window of  $N$  words. To enable the comparison of source and target vectors, words in the source vectors are translated into the target language using an existing bilingual dictionary.

The core of the standard approach is the bilingual dictionary. Its use 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 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 introduce noise in context vectors.

In this paper, we present a novel approach that addresses the word polysemy problem neglected in the standard approach. We introduce a Word Sense Disambiguation (WSD) process that identifies the translations of polysemous words that are more likely to give the best representation of context vectors in the target language. For this purpose, we employ five WordNet-based semantic *similarity* and *relatedness* measures and use a *data fusion* method that merges the results obtained by each measure. We test our approach on two specialized French-English comparable corpora (*financial and medical*) and report improved results compared to two state-of-the-art approaches.

## 2 Related Work

Most previous works addressing the task of bilingual lexicon extraction from comparable corpora are based on the standard approach. In order to improve the results of this approach, recent researches based on the assumption that more the context vectors are representative, better is the bilingual lexicon extraction were conducted. In these works, additional linguistic resources such as specialized dictionaries (Chiao and Zweigenbaum, 2002) or transliterated words (Prochasson et al., 2009) were combined with the bilingual dic-

tionary to translate context vectors. Few works have however focused on the ambiguity problem revealed by the seed bilingual dictionary. (Hazem and Morin, 2012) propose a method that filters the entries of the bilingual dictionary on the base of a POS-Tagging and a domain relevance measure criteria but no improvements have been demonstrated. Gaussier et al. (2004) attempted to solve the problem of 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 an improvement of the F-Measure (+0.02 at Top20) were reported for a mixed method. Recently, (Morin and Prochasson, 2011) proceed as the standard approach but weigh the different translations according to their frequency in the target corpus. Here, we propose a method that differs from Gaussier et al. (2004) in this way: If they focus on words ambiguities on source and target languages, we thought that it would be sufficient to disambiguate only translated source context vectors.

### 3 Context Vector Disambiguation

#### 3.1 Semantic similarity measures

A large number of WSD techniques were proposed in the literature. The most widely used ones are those that compute semantic similarity<sup>1</sup> with the help of WordNet. WordNet has been used in 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 it 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 bilingual lexicon extraction from comparable corpora.

Among semantic similarity measures using WordNet, we distinguish: (1) measures based on path length which simply counts the distance between two words in the WordNet taxonomy, (2) measures relying on information content in which a semantically annotated corpus is needed to compute frequencies of words to be compared and (3) the ones using gloss overlap which are designed to compute semantic relatedness. In this work, we use five similarity measures and compare their performances. These measures include three

<sup>1</sup>For consistence, we often use “semantic similarity” to refer collectively to both similarity and relatedness.

path-based semantic similarity measures denoted PATH,WUP (Wu and Palmer, 1994) and LEACOCK (Leacock and Chodorow, 1998). PATH is a baseline that is equal to the inverse of the shortest path between two words. WUP finds the depth of the least common subsumer of the words, and scales that by the sum of the depths of individual words. The depth of a word is its distance to the root node. LEACOCK finds the shortest path between two words, and scales that by the maximum path length found in the is-a hierarchy in which they occur. Path length measures have the advantage of being independent of corpus statistics, and therefor uninfluenced by sparse data.

Since semantic relatedness is considered to be more general than semantic similarity, we also use two relatedness measures: LESK (Banerjee and Pedersen, 2002) and VECTOR (Patwardhan, 2003). LESK finds overlaps between the glosses of word pairs, as well as words’ hyponyms. VECTOR creates a co-occurrence matrix for each gloss token. Each gloss is then represented as a vector that averages token co-occurrences.

#### 3.2 Disambiguation process

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. According to the above-described semantic similarity measures, a similarity value  $Sim_{Value}$  is derived between all the translations provided for each polysemous word by the bilingual dictionary and all monosemic words appearing within the same context vector. In practice, since a word can belong to more than one synset<sup>2</sup> in WordNet, the semantic similarity between two words  $w_1$  and  $w_2$  is defined as the *maximum* of  $Sim_{Value}$  between the synset or the synsets that include the  $synsets(w_1)$  and

<sup>2</sup>a group of a synonymous words in WordNet



$synsets(w_2)$  according to the following equation:

$$Sem_{Sim}(w_1, w_2) = \max\{Sim_{Value}(s_1, s_2); (s_1, s_2) \in synsets(w_1) \times synsets(w_2)\} \quad (1)$$

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  $w_p$ :

$$Ave\_Sim(w_p^j) = \frac{1}{N} \sum_{i=1}^N Sem_{Sim}(w_i, w_p^j) \quad (2)$$

where  $N$  is the total number of monosemic words in each context vector and  $Sem_{Sim}$  is the similarity value of  $w_p^j$  and the  $i^{th}$  monosemic word. Hence, according to average similarity values  $Ave\_Sim(w_p^j)$ , we obtain for each polysemous word  $w_p$  an ordered list of translations  $w_p^1 \dots w_p^n$ .

## 4 Experiments and Results

### 4.1 Resources and Experimental Setup

We conducted our experiments on two French-English comparable corpora specialized on the *corporate finance* and the *breast cancer* sub-domains. Both corpora were extracted from Wikipedia<sup>3</sup>. We consider the domain 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 *categories tree*. Then we collected all articles belonging to one of these categories and used inter-language links to build the comparable corpus. Both corpora have been normalized through the following linguistic preprocessing steps: tokenisation, part-of-speech tagging, lemmatisation and function words removal. The resulting corpora<sup>4</sup> sizes as well as their polysemy rate  $P_R$  are given in Table 1. The polysemy rate indicates how much words in the comparable corpora are associated to more than one translation in the seed bilingual dictionary. The dictionary consists of an in-house bilingual dictionary which contains about 120,000 entries belonging to the general language with an average of 7 translations per entry.

In bilingual terminology extraction from comparable corpora, a reference list is required to evaluate the performance of the alignment. Such lists are often composed of about 100 single

| Corpus                   | French  | English | $P_R$ |
|--------------------------|---------|---------|-------|
| <i>Corporate finance</i> | 402.486 | 756.840 | 41%   |
| <i>Breast cancer</i>     | 396.524 | 524.805 | 47%   |

Table 1: Comparable corpora sizes in term of words and polysemy rates ( $P_R$ ) associated to each corpus

terms (Hazem and Morin, 2012; Chiao and Zweigenbaum, 2002). Here, we created two reference lists<sup>5</sup> for the *corporate finance* and the *breast cancer* sub-domains. The first list is composed of 125 single terms extracted from the glossary of bilingual micro-finance terms<sup>6</sup>. The second list contains 79 terms extracted from the French-English MESH and the UMLS thesauri<sup>7</sup>. Note that reference terms pairs appear more than five times in each part of both comparable corpora.

Three other parameters need to be set up, namely the window size, the association measure and the similarity measure. We followed (Laroche and Langlais, 2010) to define these parameters. They carried out a complete study of the influence of these parameters on the bilingual alignment. The context vectors were defined by computing the Discounted Log-Odds Ratio (equation 3) between words occurring in the same context window of size 7.

$$Odds-Ratio_{disc} = \log \frac{(O_{11} + \frac{1}{2})(O_{22} + \frac{1}{2})}{(O_{12} + \frac{1}{2})(O_{21} + \frac{1}{2})} \quad (3)$$

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. As similarity measure, we chose to use the cosine measure.

### 4.2 Results of bilingual lexicon extraction

To evaluate the performance of our approach, we used both the standard approach (SA) and the approach proposed by (Morin and Prochasson, 2011) (henceforth MP11) as baselines. The experiments were performed with respect to the five semantic similarity measures described in section 3.1. Each measure provides, for each polysemous word, a ranked list of translations. A question that arises here is whether we should introduce only the top-ranked translation into the context vector or consider a larger number of translations, mainly when a translation list contains synonyms. For this

<sup>3</sup><http://dumps.wikimedia.org/>

<sup>4</sup>Comparable corpora will be shared publicly

<sup>5</sup>Reference lists will be shared publicly

<sup>6</sup><http://www.microfinance.lu/en/>

<sup>7</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> |  |
|----------------------|----------------------------|------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--|
| a) Corporate Finance |                            | Standard Approach (SA) | 0.172             |                   |                   |                   |                   |                   |                   |  |
|                      |                            | MP11                   | 0.336             |                   |                   |                   |                   |                   |                   |  |
|                      | Single measure             | WUP                    | 0.241             | 0.284             | <i>0.301</i>      | 0.275             | 0.258             | 0.215             | 0.224             |  |
|                      |                            | PATH                   | 0.250             | 0.284             | <i>0.301</i>      | <i>0.284</i>      | 0.258             | 0.215             | 0.215             |  |
|                      |                            | LEACOCK                | 0.250             | <i>0.293</i>      | <i>0.301</i>      | 0.275             | <i>0.275</i>      | 0.241             | <i>0.232</i>      |  |
|                      |                            | LESK                   | <i>0.272</i>      | <i>0.293</i>      | 0.293             | 0.275             | 0.258             | <i>0.250</i>      | 0.215             |  |
|                      |                            | VECTOR                 | 0.267             | 0.310             | 0.284             | <i>0.284</i>      | 0.232             | 0.232             | <i>0.232</i>      |  |
|                      | CONDORCET <sub>Merge</sub> | <b>0.362</b>           | <b>0.379</b>      | <b>0.353</b>      | <b>0.362</b>      | <b>0.336</b>      | 0.275             | 0.267             |                   |  |
|                      |                            | 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) Breast Cancer     |                            | Standard Approach (SA) | 0.493             |                   |                   |                   |                   |                   |                   |  |
|                      |                            | MP11                   | 0.553             |                   |                   |                   |                   |                   |                   |  |
|                      | Single measure             | WUP                    | 0.481             | 0.566             | <i>0.566</i>      | 0.542             | 0.554             | 0.542             | <i>0.554</i>      |  |
|                      |                            | PATH                   | 0.542             | 0.542             | 0.554             | 0.566             | <i>0.578</i>      | 0.554             | <i>0.554</i>      |  |
|                      |                            | LEACOCK                | 0.506             | <i>0.578</i>      | 0.554             | 0.566             | 0.542             | 0.554             | 0.542             |  |
|                      |                            | LESK                   | 0.469             | 0.542             | 0.542             | <b>0.590</b>      | 0.554             | 0.554             | 0.542             |  |
|                      |                            | VECTOR                 | <i>0.518</i>      | 0.566             | 0.530             | 0.566             | 0.542             | <i>0.566</i>      | <i>0.554</i>      |  |
|                      | CONDORCET <sub>Merge</sub> | <b>0.566</b>           | <b>0.614</b>      | <b>0.600</b>      | <b>0.590</b>      | <b>0.600</b>      | <b>0.578</b>      | <b>0.578</b>      |                   |  |

Table 2: F-Measure at Top20 for the two domains; MP11 = (Morin and Prochasson, 2011). In each column, italics shows best single similarity measure, bold shows best result. Underline shows best result overall.

reason, we take into account in our experiments different numbers of translations, noted  $WN-T_i$ , ranging from the pivot translation ( $i = 1$ ) to the seventh word in the translation list. This choice is motivated by the fact that words in both corpora have on average 7 translations in the bilingual dictionary. Both baseline systems use all translations associated to each entry in the bilingual dictionary. The only difference is that in MP11 translations are weighted according to their frequency in the target corpus.

The results of different works focusing on bilingual lexicon extraction from comparable corpora are evaluated on the number of correct candidates found in the first  $N$  first candidates output by the alignment process (the Top $N$ ). Here, we use the Top20 F-measure as evaluation metric. The results obtained for the *corporate finance* corpus are presented in Table 2a. The first notable observation is that disambiguating context vectors using semantic similarity measures outperforms the SA. The highest F-measure is reported by VECTOR. Using the top two words ( $WN-T_2$ ) in context vectors increases the F-measure from 0.172 to 0.310. However, compared to MP11, no improvement is achieved. Concerning the *breast cancer* corpus, Table 2b shows improvements in most cases over both the SA and MP11. The maximum F-

measure was obtained by LESK when for each polysemous word up to four translations ( $WN-T_4$ ) are considered in context vectors. This method achieves an improvement of respectively +0.097 and +0.037% over SA and MP11.

Each of the tested 5 semantic similarity measures provides a different view of how to rank the translations of a given test word. Combining the obtained ranked lists should reinforce the confidence in consensus translations, while decreasing the confidence in non-consensus translations. We have therefore tested their combination. For this, we used a voting method, and chose one in the Condorcet family the *Condorcet data fusion method*. This method was widely used to combine document retrieval results from information retrieval systems (Montague and Aslam, 2002; Nurray and Can, 2006). It is a single-winner election method that ranks the candidates in order of preference. It is a *pairwise voting*, i.e. it compares every possible pair of candidates to decide the preference of them. A matrix can be used to present the competition process. Every candidate appears in the matrix as a row and a column as well. If there are  $m$  candidates, then we need  $m^2$  elements in the matrix in total. Initially 0 is written to all the elements. If  $d_i$  is preferred to  $d_j$ , then we add 1 to the element at row  $i$  and column  $j$  ( $a_{ij}$ ). The pro-

cess is repeated until all the ballots are processed. For every element  $a_{ij}$ , if  $a_{ij} > m/2$ , then  $d_i$  beats  $d_j$ ; if  $a_{ij} < m/2$ , then  $d_j$  beats  $d_i$ ; otherwise ( $a_{ij} = m/2$ ), there is a draw between  $d_i$  and  $d_j$ . The total score of each candidate is quantified by summing the raw scores it obtains in all pairwise competitions. Finally the ranking is achievable based on the total scores calculated.

Here, we view the ranking of the extraction results from different similarity measures as a special instance of the voting problem where the Top20 extraction results correspond to candidates and different semantic similarity measures are the voters. The combination method referred to as CONDORCET<sub>Merge</sub> outperformed all the others (see Tables 2a and 2b): (1) individual measures, (2) SA, and (3) MP11. Even though the two corpora are fairly different (subject and polysemy rate), the optimal results are obtained when considering up to two most similar translations in context vectors. This behavior shows that the fusion method is robust to domain change. The addition of supplementary translations, which are probably noisy in the given domain, degrades the overall results. The F-measure gains with respect to SA are +0.207 for corporate finance and +0.121 for the breast cancer corpus. More interestingly, our approach outperforms MP11, showing that the role of disambiguation is more important than that of feature weighting.

## 5 Conclusion

We presented in this paper a novel method that extends the standard approach used for bilingual lexicon extraction. This method disambiguates polysemous words in context vectors by selecting only the most relevant translations. Five semantic similarity and relatedness measures were used for this purpose. Experiments conducted on two specialized comparable corpora indicate that the combination of similarity metrics leads to a better performance than two state-of-the-art approaches. This shows that the ambiguity present in specialized comparable corpora hampers bilingual lexicon extraction, and that methods such as the one introduced here are needed. The obtained results are very encouraging and can be improved in a number of ways. First, we plan to mine much larger specialized comparable corpora and focus on their quality (Li and Gaussier, 2010). We also plan to test our method on bilingual lexicon extrac-

tion from general-domain corpora, where ambiguity is generally higher and disambiguation methods should be all the more needed.

## References

- Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '02*, pages 136–145, London, UK, UK. Springer-Verlag.
- 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, Myunggwon 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).
- 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.
- Myunggwon 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 Laroché 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.
- Claudia Leacock and Martin Chodorow, 1998. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.
- 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.
- Mark Montague and Javed A. Aslam. 2002. Concorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management, CIKM '02*, pages 538–548, New York, NY, USA. ACM.
- Emmanuel Morin and Emmanuel Prochasson. 2011. Bilingual lexicon extraction from comparable corpora enhanced with parallel corpora. In *Proceedings, 4th Workshop on Building and Using Comparable Corpora (BUCC)*, page 27–34, Portland, Oregon, USA.
- Rabia Nuray and Fazli Can. 2006. Automatic ranking of information retrieval systems using data fusion. *Inf. Process. Manage.*, 42(3):595–614, May.
- Siddharth Patwardhan. 2003. Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness. Master’s thesis, University of Minnesota, Duluth, August.
- 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.
- 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.

# The Effects of Lexical Resource Quality on Preference Violation Detection

**Jesse Dunietz**

Computer Science Department  
Carnegie Mellon University  
Pittsburgh, PA, 15213, USA  
jdunietz@cs.cmu.edu

**Lori Levin and Jaime Carbonell**

Language Technologies Institute  
Carnegie Mellon University  
Pittsburgh, PA, 15213, USA  
{lsl, jgc}@cs.cmu.edu

## Abstract

Lexical resources such as WordNet and VerbNet are widely used in a multitude of NLP tasks, as are annotated corpora such as treebanks. Often, the resources are used as-is, without question or examination. This practice risks missing significant performance gains and even entire techniques.

This paper addresses the importance of resource quality through the lens of a challenging NLP task: detecting selectional preference violations. We present DAVID, a simple, lexical resource-based preference violation detector. With as-is lexical resources, DAVID achieves an  $F_1$ -measure of just 28.27%. When the resource entries and parser outputs for a small sample are corrected, however, the  $F_1$ -measure on that sample jumps from 40% to 61.54%, and performance on other examples rises, suggesting that the algorithm becomes practical given refined resources. More broadly, this paper shows that resource quality matters tremendously, sometimes even more than algorithmic improvements.

## 1 Introduction

A variety of NLP tasks have been addressed using *selectional preferences* or *restrictions*, including word sense disambiguation (see Navigli (2009)), semantic parsing (e.g., Shi and Mihalcea (2005)), and metaphor processing (see Shutova (2010)). These semantic problems are quite challenging; metaphor analysis, for instance, has long been recognized as requiring considerable semantic knowledge (Wilks, 1978; Carbonell, 1980). The advent of extensive lexical resources, annotated corpora, and a spectrum of NLP tools

presents an opportunity to revisit such challenges from the perspective of selectional preference violations. Detecting these violations, however, constitutes a severe stress-test for resources designed for other tasks. As such, it can highlight shortcomings and allow quantifying the potential benefits of improving resources such as WordNet (Fellbaum, 1998) and VerbNet (Schuler, 2005).

In this paper, we present DAVID (Detector of Arguments of Verbs with Incompatible Denotations), a resource-based system for detecting preference violations. DAVID is one component of METAL (Metaphor Extraction via Targeted Analysis of Language), a new system for identifying, interpreting, and cataloguing metaphors. One purpose of DAVID was to explore how far lexical resource-based techniques can take us. Though our initial results suggested that the answer is “not very,” further analysis revealed that the problem lies less in the technique than in the state of existing resources and tools.

Often, it is assumed that the frontier of performance on NLP tasks is shaped entirely by algorithms. Manning (2011) showed that this may not hold for POS tagging – that further improvements may require resource cleanup. In the same spirit, we argue that for some semantic tasks, exemplified by preference violation detection, resource quality may be at least as essential as algorithmic enhancements.

## 2 The Preference Violation Detection Task

DAVID builds on the insight of Wilks (1978) that the strongest indicator of metaphoricity is the violation of selectional preferences. For example, only plants can literally be pruned. If *laws* is the object of *pruned*, the verb is likely metaphorical. Flagging such semantic mismatches between verbs and arguments is the task of preference violation detection.

We base our definition of preferences on the Pragglejaz guidelines (Pragglejaz Group, 2007) for identifying the most basic sense of a word as the most concrete, embodied, or precise one. Similarly, we define selectional preferences as the semantic constraints imposed by a verb’s most basic sense. Dictionaries may list figurative senses of *prune*, but we take the basic sense to be cutting plant growth.

Several types of verbs were excluded from the task because they have very lax preferences. These include verbs of becoming or seeming (e.g., *transform*, *appear*), light verbs, auxiliaries, and aspectual verbs. For the sake of simplifying implementation, phrasal verbs were also ignored.

### 3 Algorithm Design

To identify violations, DAVID employs a simple algorithm based on several existing tools and resources: SENNA (Collobert et al., 2011), a semantic role labeling (SRL) system; VerbNet, a computational verb lexicon; SemLink (Loper et al., 2007), which includes mappings between PropBank (Palmer et al., 2005) and VerbNet; and WordNet. As one metaphor detection component of METAL’s several, DAVID is designed to favor precision over recall. The algorithm is as follows:

1. Run the Stanford CoreNLP POS tagger (Toutanova et al., 2003) and the TurboParser dependency parser (Martins et al., 2011).
2. Run SENNA to identify the semantic arguments of each verb in the sentence using the PropBank argument annotation scheme (Arg0, Arg1, etc.). See Table 1 for example output.
3. For each verb *V*, find all VerbNet entries for *V*. Using SemLink, map each PropBank argument name to the corresponding VerbNet thematic roles in these entries (Agent, Patient, etc.). For example, the VerbNet class for *prune* is *carve-21.2-2*. SemLink maps Arg0 to the Agent of *carve-21.2-2* and Arg1 to the Patient.
4. Retrieve from VerbNet the selectional restrictions of each thematic role. In our running example, VerbNet specifies *+int\_control* and *+concrete* for the Agent and Patient of *carve-21.2-2*, respectively.
5. If the head of any argument cannot be interpreted to meet *V*’s preferences, flag *V* as a violation.

“The politician pruned laws regulating plastic bags, and created new fees for inspecting dairy farms.”

| Verb       | Arg0           | Arg1          |
|------------|----------------|---------------|
| pruned     | The politician | laws ... bags |
| regulating | laws           | plastic bags  |
| created    | The politician | new fees      |
| inspecting | - -            | dairy farms   |

Table 1: SENNA’s SRL output for the example sentence above. Though this example demonstrates only two arguments, SENNA is capable of labeling up to six.

| Restriction  | WordNet Synsets                                       |
|--------------|---|
| animate      | animate_being.n.01<br>people.n.01<br>person.n.01      |
| concrete     | physical_object.n.01<br>matter.n.01<br>substance.n.01 |
| organization | social_group.n.01<br>district.n.01                    |

Table 2: DAVID’s mappings between some common VerbNet restriction types and WordNet synsets.

Each VerbNet restriction is interpreted as mandating or forbidding a set of WordNet hypernyms, defined by a custom mapping (see Table 2). For example, VerbNet requires both the Patient of a verb in *carve-21.2-2* and the Theme of a verb in *wipe\_manner-10.4.1-1* to be concrete. By empirical inspection, concrete nouns are hyponyms of the WordNet synsets *physical\_object.n.01*, *matter.n.03*, or *substance.n.04*. *Laws* (the Patient of *prune*) is a hyponym of none of these, so *prune* would be flagged as a violation.

### 4 Corpus Annotation

To evaluate our system, we assembled a corpus of 715 sentences from the METAL project’s corpus of sentences with and without metaphors. The corpus was annotated by two annotators following an annotation manual. Each verb was marked for whether its arguments violated the selectional preferences of the most basic, literal meaning of the verb. The annotators resolved conflicts by dis-

| Error source                   | Frequency    |
|--------------------------------|--------------|
| Bad/missing VN entries         | 4.5 (14.1%)  |
| Bad/missing VN restrictions    | 6 (18.8%)    |
| Bad/missing SL mappings        | 2 (6.3%)     |
| Parsing/head-finding errors    | 3.5 (10.9%)  |
| SRL errors                     | 8.5 (26.6%)  |
| VN restriction system too weak | 4 (12.5%)    |
| Confounding WordNet senses     | 3.5 (10.9%)  |
| <b>Endemic errors:</b>         | 7.5 (23.4%)  |
| <b>Resource errors:</b>        | 12.5 (39.1%) |
| <b>Tool errors:</b>            | 12 (37.5%)   |
| <b>Total:</b>                  | 32 (100%)    |

Table 3: Sources of error in 90 randomly selected sentences. For errors that were due to a combination of sources, 1/2 point was awarded to each source. (VN stands for VerbNet and SL for SemLink.)

cussing until consensus.

## 5 Initial Results

As the first row of Table 4 shows, our initial evaluation left little hope for the technique. With such low precision and  $F_1$ , it seemed a lexical resource-based preference violation detector was out. When we analyzed the errors in 90 randomly selected sentences, however, we found that most were not due to systemic problems with the approach; rather, they stemmed from SRL and parsing errors and missing or incorrect resource entries (see Table 3). Armed with this information, we decided to explore how viable our algorithm would be absent these problems.

## 6 Refining The Data

To evaluate the effects of correcting DAVID’s inputs, we manually corrected the tool outputs and resource entries that affected the aforementioned 90 sentences. SRL output was corrected for every sentence, while SemLink and VerbNet entries were corrected only for each verb that produced an error.

### 6.1 Corrections to Tool Output (Parser/SRL)

Guided by the PropBank database and annotation guidelines, we corrected all errors in core role assignments from SENNA. These corrections included relabeling arguments, adding missed arguments, fixing argument spans, and deleting anno-

tations for non-verbs. The only parser-related error we corrected was a mislabeled noun.

### 6.2 Correcting Corrupted Data in VerbNet

The VerbNet download is missing several subclasses that are referred to by SemLink or that have been updated on the VerbNet website. Some roles also have not been updated to the latest version, and some subclasses are listed with incorrect IDs. These problems, which caused SemLink mappings to fail, were corrected before reviewing errors from the corpus.

Six subclasses needed to be fixed, all of which were easily detected by a simple script that did not depend on the 90-sentence subcorpus. We therefore expect that few further changes of this type would be needed for a more complete resource refinement effort.

### 6.3 Corpus-Based Updates to SemLink

Our modifications to SemLink’s mappings included adding missing verbs, adding missing roles to mappings, and correcting mappings to more appropriate classes or roles. We also added null mappings in cases where a PropBank argument had no corresponding role in VerbNet. This makes the system’s strategy for ruling out mappings more reliable.

No corrections were made purely based on the sample. Any time a verb’s mappings were edited, VerbNet was scoured for plausible mappings for every verb sense in PropBank, and any nonsensical mappings were deleted. For example, when the phrase *go dormant* caused an error, we inspected the mappings for *go*. Arguments of all but 2 of the 7 available mappings were edited, either to add missing arguments or to correct nonsensical ones. These changes actually had a net negative impact on test set performance because the bad mappings had masked parsing and selectional preference problems.

Based on the 90-sentence subcorpus, we modified 20 of the existing verb entries in SemLink. These changes included correcting 8 role mappings, adding 13 missing role mappings to existing senses, deleting 2 incorrect senses, adding 11 verb senses, correcting 2 senses, deleting 1 superfluous role mapping, and adding 46 null role mappings. (Note that although null mappings represented the largest set of changes, they also had the least impact on system behavior.) One entirely new verb was added, as well.

## 6.4 Corpus-Based Updates to VerbNet

Nineteen VerbNet classes were modified, and one class had to be added. The modifications generally involved adding, correcting, or deleting selectional restrictions, often by introducing or rearranging subclasses. Other changes amounted to fixing clerical errors, such as incorrect role names or restrictions that had been ANDed instead of ORed.

An especially difficult problem was an inconsistency in the semantics of VerbNet’s subclass system. In some cases, the restrictions specified on a verb in a subclass did not apply to subcategorization frames inherited from a superclass, but in other cases the restrictions clearly applied to all frames. The conflict was resolved by duplicating subclassed verbs in the top-level class whenever different selectional restrictions were needed for the two sets of frames.

As with SemLink, samples determined only *which* classes were modified, not what modifications were made. Any non-obvious changes to selectional restrictions were verified by examining dozens of verb instances from SketchEngine’s (Kilgarriff et al., 2004) corpus. For example, the Agent of *seek* was restricted to `+animate`, but the corpus confirmed that organizations are commonly described non-metaphorically as seeking, so the restriction was updated to `+animate | +organization`.

## 7 Results After Resource Refinement

After making corrections for each set of 10 sentences, we incrementally recomputed  $F_1$  and precision, both on the subcorpus corrected so far and on a test set of all 625 sentences that were never corrected. (The manual nature of the correction effort made testing  $k$ -fold subsets impractical.) The results for 30-sentence increments are shown in Table 4.

The most striking feature of these figures is how much performance improves on corrected sentences: for the full 90 sentences,  $F_1$  rose from 30.43% to 61.54%, and precision rose even more dramatically from 31.82% to 80.00%. Interestingly, resource corrections alone generally made a larger difference than tool corrections alone, suggesting that resources may be the dominant factor in resource-intensive tasks such as this one. Even more compellingly, the improvement from correcting both the tools and the resources was

nearly double the sum of the improvements from each alone: tool and resource improvements interact synergistically.

The effects on the test corpus are harder to interpret. Due to a combination of SRL problems and the small number of sentences corrected, the scores on the test set improved little with resource correction; in fact, they even dipped slightly between the 30- and 60-sentence increments. Nonetheless, we contend that our results testify to the generality of our corrections: after each iteration, every altered result was either an error fixed or an error that should have appeared before but had been masked by another. Note also that all results on the test set are without corrected tool output; presumably, these sentences would also have improved synergistically with more accurate SRL. How long corrections would continue to improve performance is a question that we did not have the resources to answer, but our results suggest that there is plenty of room to go.

Some errors, of course, are endemic to the approach and cannot be fixed either by improved resources or by better tools. For example, we consider every WordNet sense to be plausible, which produces false negatives. Additionally, the selectional restrictions specified by VerbNet are fairly loose; a more refined set of categories might capture the range of verbs’ restrictions more accurately.

## 8 Implications for Future Refinement Efforts

Although improving resources is infamously labor-intensive, we believe that similarly refining the remainder of VerbNet and SemLink would be doable. In our study, it took about 25-35 person-hours to examine about 150 verbs and to modify 20 VerbNet classes and 25 SemLink verb entries (excluding time for SENNA corrections, fixing corrupt VerbNet data, and analysis of DAVID’s errors). Extrapolating from our experience, we estimate that it would take roughly 6-8 person-weeks to systematically fix this particular set of issues with VerbNet.

Improving SemLink could be more complex, as its mappings are automatically generated from VerbNet annotations on top of the PropBank corpus. One possibility is to correct the generated mappings directly, as we did in our study, which we estimate would take about two person-months.



With the addition of some metadata from the generation process, it would then be possible to follow the corrected mappings back to annotations from which they were generated and fix those annotations. One downside of this approach is that if the mappings were ever regenerated from the annotated corpus, any mappings not encountered in the corpus would have to be added back afterwards.

Null role mappings would be particularly thorny to implement. To add a null mapping, we must know that a role definitely does not belong, and is not just incidentally missing from an example. For instance, VerbNet’s *defend-85* class truly has no equivalent to *Arg2* in PropBank’s *defend.01*, but *Arg0* or *Arg1* may be missing for other reasons (e.g., in a passive). It may be best to simply omit null mappings, as is currently done. Alternatively, full parses from the Penn Treebank, on which PropBank is based, might allow distinguishing phenomena such as passives where arguments are predictably omitted.

The maintainers of VerbNet and PropBank are aware of many of the issues we have raised, and we have been in contact with them about possible approaches to fixing them. They are particularly aware of the inconsistent semantics of selectional restrictions on VerbNet subclasses, and they hope to fix this issue within a larger attempt at retooling VerbNet’s selectional restrictions. In the meantime, we are sharing our VerbNet modifications with them for them to verify and incorporate. We are also sharing our SemLink changes so that they can, if they choose, continue manual correction efforts or trace SemLink problems back to the annotated corpus.

## 9 Conclusion

Our results argue for investing effort in developing and fixing resources, in addition to developing better NLP tools. Resource and tool improvements interact synergistically: better resources multiply the effect of algorithm enhancements. Gains from fixing resources may sometimes even exceed what the best possible algorithmic improvements can provide. We hope the NLP community will take up the challenge of investing in its resources to the extent that its tools demand.

## Acknowledgments

Thanks to Eric Nyberg for suggesting building a system like DAVID, to Spencer Onuffer for his an-

| Sent. | Tools | Rsrcs    | P      | F <sub>1</sub> |
|-------|-------|----------|--------|----------------|
| 715   | 0     | 0        | 27.14% | 28.27%         |
| 625   | 0     | 0        | 26.55% | 27.98%         |
| 625   | 0     | corr.    | 26.37% | 28.15%         |
| 30    | 0     | 0        | 50.00% | 40.00%         |
| 30    | 30    | 0        | 66.67% | 44.44%         |
| 30    | 0     | corr.+30 | 62.50% | 50.00%         |
| 30    | 30    | corr.+30 | 87.50% | 70.00%         |
| 625   | 0     | corr.+30 | 27.07% | 28.82%         |
| 60    | 0     | 0        | 35.71% | 31.25%         |
| 60    | 60    | 0        | 54.55% | 31.38%         |
| 60    | 0     | corr.+60 | 53.85% | 45.16%         |
| 60    | 60    | corr.+60 | 90.91% | 68.97%         |
| 625   | 0     | corr.+60 | 26.92% | 28.74%         |
| 90    | 0     | 0        | 31.82% | 30.43%         |
| 90    | 90    | 0        | 44.44% | 38.10%         |
| 90    | 0     | corr.+90 | 47.37% | 41.86%         |
| 90    | 90    | corr.+90 | 80.00% | 61.54%         |
| 625   | 0     | corr.+90 | 27.37% | 28.99%         |

Table 4: Performance on preference violation detection task. Column 1 shows the sentence count. Columns 2 and 3 show how many sentences’ SRL/parsing and resource errors, respectively, had been fixed (“corr.” indicates corrupted files).

notation efforts, and to Davida Fromm for curating METAL’s corpus of English sentences.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0020. 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, DoD/ARL, or the U.S. Government.

## References

Jaime G. Carbonell. 1980. Metaphor: a key to extensible semantic analysis. In *Proceedings of the 18th annual meeting on Association for Computational Linguistics*, ACL ’80, pages 17–21, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael

- Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In *Proceedings of EURALEX*.
- Edward Loper, Szu-ting Yi, and Martha Palmer. 2007. Combining lexical resources: Mapping between PropBank and VerbNet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.
- Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189. Springer.
- André F. T. Martins, Noah A. Smith, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. 2011. Dual decomposition with many overlapping components. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 238–249, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.
- Karin K. Schuler. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA. AAI3179808.
- Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 3406 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin Heidelberg.
- Ekaterina Shutova. 2010. Models of metaphor in NLP. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, pages 688–697, 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, NAACL '03*, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yorick Wilks. 1978. Making preferences more active. *Artificial Intelligence*, 11:197–223.

# Exploiting Qualitative Information from Automatic Word Alignment for Cross-lingual NLP Tasks

**José G.C. de Souza**  
FBK-irst,  
University of Trento  
Trento, Italy  
desouza@fbk.eu

**Miquel Esplà-Gomis**  
Universitat d'Alacant  
Alacant, Spain  
mespla@dlsi.ua.es

**Marco Turchi**  
FBK-irst  
Trento, Italy  
turchi@fbk.eu

**Matteo Negri**  
FBK-irst  
Trento, Italy  
negri@fbk.eu

## Abstract

The use of automatic word alignment to capture sentence-level semantic relations is common to a number of cross-lingual NLP applications. Despite its proved usefulness, however, word alignment information is typically considered from a quantitative point of view (*e.g.* the number of alignments), disregarding qualitative aspects (the importance of aligned terms). In this paper we demonstrate that integrating qualitative information can bring significant performance improvements with negligible impact on system complexity. Focusing on the cross-lingual textual entailment task, we contribute with a novel method that: *i)* significantly outperforms the state of the art, and *ii)* is portable, with limited loss in performance, to language pairs where training data are not available.

## 1 Introduction

Meaning representation, comparison and projection across sentences are major challenges for a variety of cross-lingual applications. So far, despite the relevance of the problem, research on multilingual applications has either circumvented the issue, or proposed partial solutions.

When possible, the typical approach builds on the reduction to a monolingual task, burdening the process with dependencies from machine translation (MT) components. For instance, in cross-lingual question answering and cross-lingual textual entailment (CLTE), intermediate MT steps are respectively performed to ease answer retrieval/presentation (Parton, 2012; Tanev et al., 2006) and semantic inference (Mehdad et al., 2010). Direct solutions that avoid such pivoting strategies typically exploit similarity measures that rely on bag-of-words representations. As an

example, most supervised approaches to MT quality estimation (Blatz et al., 2003; Callison-Burch et al., 2012) and CLTE (Wäschle and Fendrich, 2012) include features that consider the amount of equivalent terms that are found in the input sentence pairs. Such simplification, however, disregards the fact that semantic equivalence is not only proportional to the number of equivalent terms, but also to their importance. In other words, instead of checking *what* of a given sentence can be found in the other, current approaches limit the analysis to *the amount* of lexical elements they share, under the rough assumption that the more the better.

In this paper we argue that:

- (1) Considering qualitative aspects of word alignments to identify sentence-level semantic relations can bring significant performance improvements in cross-lingual NLP tasks.
- (2) Shallow linguistic processing techniques (often a constraint in real cross-lingual scenarios due to limited resources availability) can be leveraged to set up portable solutions that still outperform current bag-of-words methods.

To support our claims we experiment with the CLTE task, which allows us to perform exhaustive comparative experiments due to the availability of comparable benchmarks for different language pairs. In the remainder of the paper, we:

- (1) Prove the effectiveness of our method over datasets for four language combinations;
- (2) Assess the portability of our models across languages in different testing conditions.

## 2 Objectives and Method

We propose a supervised learning approach for identifying and classifying semantic relations between two sentences  $T_1$  and  $T_2$  written in different languages. Beyond semantic equivalence, which is relevant to applications such as MT quality es-

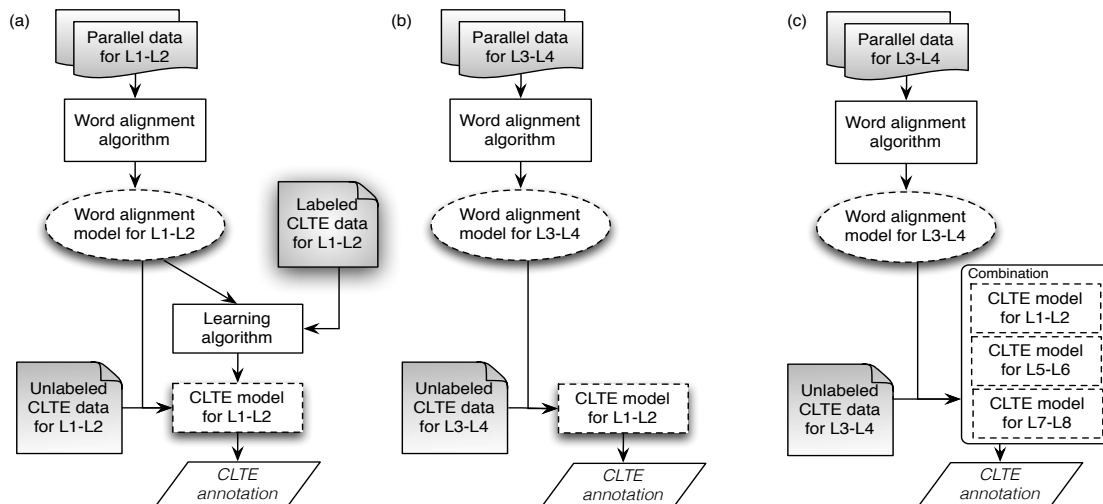


Figure 1: System architecture in different training/evaluation conditions. (a): parallel data and CLTE labeled data are available for language pair L1-L2. (b): the L1-L2 CLTE model is used to cope with the unavailability of labeled data for L3-L4. (c): the same problem is tackled by combining multiple models.

timization (Mehdad et al., 2012b),<sup>1</sup> we aim to capture a richer set of relations potentially relevant to other tasks. For instance, recognizing unrelatedness, forward and backward entailment relations, represents a core problem in cross-lingual document summarization (Lenci et al., 2002) and content synchronization (Monz et al., 2011; Mehdad et al., 2012a). CLTE, as proposed within the SemEval evaluation exercises (Negri et al., 2012; Negri et al., 2013), represents an ideal framework to evaluate such capabilities. Within this framework, our goal is to automatically identify the following entailment relations between  $T_1$  and  $T_2$ : *forward* ( $T_1 \rightarrow T_2$ ), *backward* ( $T_1 \leftarrow T_2$ ), *bidirectional* ( $T_1 \leftrightarrow T_2$ ) and *no\_entailment*.

Our approach (see Figure 1) involves two core components: *i*) a word alignment model, and *ii*) a CLTE classifier. The former is trained on a parallel corpus, and associates equivalent terms in  $T_1$  and  $T_2$ . The information about word alignments is used to extract quantitative (amount and distribution of the alignments) and qualitative features (importance of the aligned terms) to train the CLTE classifier. Although in principle both components need training data (respectively a parallel corpus and labeled CLTE data), our goal is to develop a method that is also portable across languages. To this aim, while the parallel corpus is necessary to train the word aligner for any language pair we want to deal with, the CLTE clas-

sifier can be designed to learn from features that capture language independent knowledge.<sup>2</sup> This allows us to experiment in different testing conditions, namely: *i*) when CLTE training data are available for a given language pair (Figure 1a), and *ii*) when CLTE training data are missing, and a model trained on other language pairs has to be reused (Figure 1b-c).

**Features.** Considering word alignment information, we extract three different groups of features: **AL**, **POS**, and **IDF**.

The **AL** group provides *quantitative* information about the aligned/unaligned words in each sentence  $T_*$  of the pair. These features are:

1. proportion of aligned words in  $T_*$ . We use this indicator as our baseline (**B** henceforth);
2. number of sequences of unaligned words, normalized by the length of  $T_*$ ;
3. length of the longest *a*) sequence of aligned words, and *b*) sequence of unaligned words, both normalized by the length of  $T_*$ ;
4. average length of *a*) the aligned word sequences, and *b*) the unaligned word sequences;
5. position of *a*) the first unaligned word, and *b*) the last unaligned word, both normalized by the length of  $T_*$ ;
6. proportion of word  $n$ -grams in  $T_*$  containing only aligned words (the feature was com-

<sup>1</sup>A translation has to be semantically equivalent to the source sentence.

<sup>2</sup>For instance, the fact that aligning all nouns and the most relevant terms in  $T_1$  and  $T_2$  is a good indicator of semantic equivalence.

puted separately for values of  $n = 1 \dots 5$ ).

The **POS** group considers the part of speech (PoS) of the words in  $T_*$  as a source of *qualitative* information about their importance. To compute these features we use the TreeTagger (Schmid, 1995), manually mapping the fine-grained set of assigned PoS labels into a more general set of tags ( $P$ ) based on the *universal PoS tag set* by Petrov et al. (2012). POS features differentiate between **aligned words** (words in  $T_1$  that are aligned to one or more words in  $T_2$ ) and **alignments** (the edges connecting words in  $T_1$  and  $T_2$ ). Features considering the aligned words in  $T_*$  are:

7. for each PoS tag  $p \in P$ , proportion of aligned words in  $T_*$  tagged with  $p$ ;
8. proportion of words in  $T_1$  aligned with words with the same PoS tag in  $T_2$  (and vice-versa);
9. for each PoS tag  $p \in P$ , proportion of words in  $T_1$  tagged as  $p$  which are aligned to words with the same tag in  $T_2$  (and vice-versa).

Features considering the alignments are:

10. proportion of alignments connecting words with the same PoS tag  $p$ ;
11. for each PoS tag  $p \in P$ , proportion of alignments connecting two words tagged as  $p$ .

**IDF**, the last feature, uses the inverse document frequency (Salton and Buckley, 1988) as another source of *qualitative* information under the assumption that rare words (and, therefore, with higher IDF) are more informative:

12. summation of all the IDF scores of the aligned words in  $T_*$  over the summation of the IDF scores of all words in  $T_*$ .

### 3 Experiments

Our experiments cover two different scenarios. First, the typical one, in which the CLTE model is trained on labeled data for the same pair of languages  $L_1$ – $L_2$  of the test set. Then, simulating the less favorable situation in which labeled training data for  $L_1$ – $L_2$  are missing, we investigate the possibility to use existing CLTE models trained on labeled data for a different language pair  $L_3$ – $L_4$ .

The SemEval 2012 CLTE datasets used in our experiments are available for four language pairs: Es–En, De–En, Fr–En, and It–En. Each dataset was created with the crowdsourcing-based method

described in Negri et al. (2011), and consists of 1000  $T_1$ – $T_2$  pairs (500 for training, 500 for test).

To train the word alignment models we used the Europarl parallel corpus (Koehn, 2005), concatenated with the News Commentary corpus<sup>3</sup> for three language pairs: De–En (2,079,049 sentences), Es–En (2,123,036 sentences), Fr–En (2,144,820 sentences). For It–En we only used the parallel data available in Europarl (1,909,115 sentences) since this language pair is not covered by the News Commentary corpus. IDF values for the words in each language were calculated on the monolingual part of these corpora, using the average IDF value of each language for unseen terms.

To build the word alignment models we used the MGIZA++ package (Gao and Vogel, 2008). Experiments have been carried out with the *hidden Markov model* (HMM) (Vogel et al., 1996) and *IBM models 3 and 4* (Brown et al., 1993).<sup>4</sup> We also explored three symmetrization techniques (Koehn et al., 2005): *union*, *intersection*, and *grow-diagonal-and*. A greedy feature selection process on training data, with different combinations of word alignment models and symmetrization methods, indicated *HMM/intersection* as the best performing combination. For this reason, all our experiments use this setting.

The SVM implementation of Weka (Hall et al., 2009) was used to build the CLTE model.<sup>5</sup> Two binary classifiers were trained to separately check  $T_1 \rightarrow T_2$  and  $T_1 \leftarrow T_2$ , merging their output to obtain the 4-class judgments (e.g. yes/yes=bidirectional, yes/no=forward).

#### 3.1 Evaluation with CLTE training data

Figure 2 shows the accuracy obtained by the different feature groups.<sup>6</sup> For the sake of comparison, state-of-the-art results achieved for each language combination at SemEval 2012 are also reported. As regards Es–En (63.2% accuracy) and De–En (55.8%), the top scores were obtained by the system described in (Wäschle and Fendrich, 2012), where a combination of binary classifiers for each entailment direction is trained with a mix-

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

<sup>4</sup>Five iterations of HMM, and three iterations of IBM models 3 and 4 have been performed on the training corpora.

<sup>5</sup>The polynomial kernel was used with parameters empirically estimated on the training set ( $C = 2.0$ , and  $d = 1$ )

<sup>6</sup>In Figures 2 and 3, the “\*” indicates statistically significant improvements over the state of the art at  $p \leq 0.05$ , calculated with approximate randomization (Padó, 2006).

ture of monolingual (*i.e.* with the input sentences translated in the same language using Google Translate<sup>7</sup>) and cross-lingual features. Although such system exploits word-alignment information to some extent, this is only done at quantitative level (*e.g.* number of unaligned words, percentage of aligned words, length of the longest unaligned subsequence). As regards It–En, the state of the art (56.6%) is represented by the system described in (Jimenez et al., 2012), which uses a pure pivoting method (using Google Translate) and adaptive similarity functions based on “soft” cardinality for flexible term comparisons. The two systems obtained the same result on Fr–En (57.0%).

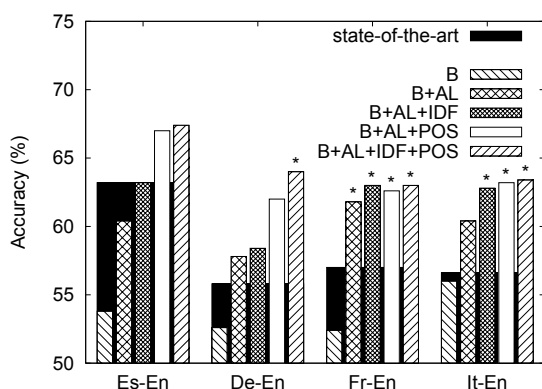


Figure 2: Accuracy obtained by each feature group on four language combinations.

As can be seen in Figure 2, the combination of *all* our features outperforms the state of the art for each language pair. The accuracy improvement ranges from 6.6% for Es–En (from 63.2% to 67.4%) to 14.6% for De–En (from 55.8% to 64%). Except for Es–En, that has very competitive state-of-the-art results, the combination of AL with POS or IDF feature groups always outperforms the best systems. Furthermore, the performance increase with qualitative features (POS and IDF) shows coherent trends across all language pairs. It is worth noting that, while we rely on a pure cross-lingual approach, both the state-of-the-art CLTE systems include features from the translation of  $T_1$  into the language of  $T_2$ . For De–En, quantitative features alone achieve lower results compared to the other languages. This can be motivated by the higher difficulty in aligning De–En pairs (this hypothesis is supported by the fact that the average number of alignments per sentence pair is 18 for De–En, and  $>22$  for the other combinations). Nevertheless, qualitative features lead to results comparable

<sup>7</sup><http://translate.google.com/>

with the other language pairs.

The selection of the best performing features for each language pair produces further improvements of varying degrees in Es–En (from 67.4% to 68%), De–En (64% – 64.8%) and It–En (63.4% – 66.8%), while performance remains stable for Fr–En (63%). All these configurations include the IDF feature (12) and the proportion of aligned words for each PoS category (7), proving the effectiveness of qualitative word alignment features.

The fact that HMM/intersection is the best combination of alignment model and symmetrization method is interesting, since it contradicts the general notion that IBM models 3 and 4 perform better than HMM (Och and Ney, 2003). A possible explanation is that, while word alignment models are usually trained on parallel corpora, the majority of CLTE sentence pairs are not parallel. In this setting, where producing reliable alignments is more difficult, IBM models are less effective for at least two reasons. First, including a word fertility model, IBM 3 and 4 limit (typically to the half of the source sentence length) the number of target words that can be aligned with the `null` word. Therefore, when such limit is reached, these models tend to force low probability, hence less reliable, word alignments. Second, in IBM model 4, the larger distortion limit makes it possible to align distant words. In the case of non-parallel sentences, this often results in wrong or noisy alignments that affect final results. For these reasons, CLTE data seem more suitable for the simpler and more conservative HMM model, and a precision-oriented symmetrization method like intersection.

### 3.2 Evaluation without CLTE training data

The goal of our second round of experiments is to investigate if, and to what extent, our approach can be considered as language-independent. Confirming this would allow to reuse models trained for a given language pair in situations where CLTE training data is missing. This is a rather realistic situation since, while bitexts to train word aligners are easier to find, the availability of labeled CLTE data is far from being guaranteed.

Our experiments have been carried out, over the same SemEval datasets, with two methods that do not use labeled data for the target language combination. The first one (method *b* in Figure 1) uses a CLTE model trained for a language pair  $L_1$ – $L_2$  for which labeled training data are avail-

able, and applies this model to a language pair  $L_3-L_4$  for which only parallel corpora are available. The second method ( $c$  in Figure 1) addresses the same problem, but exploits a combination of CLTE models trained for different language pairs. For each test set, the models trained for the other three language pairs are used in a voting scheme, in order to check whether they can complement each other to increase final results.

All the experiments have been performed using the best CLTE model for each language pair, comparing results with those presented in Section 3.1.

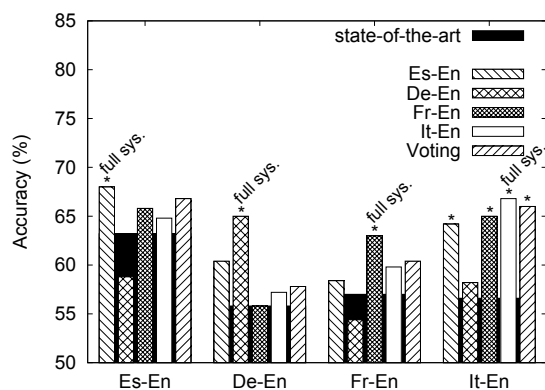


Figure 3: Accuracy obtained by reusing CLTE models (alone and in a voting scheme).

As shown in Figure 3, reusing models for a new language pair leads to results that still outperform the state of the art.<sup>6</sup> Remarkably, when used for other language combinations, the Es–En, It–En, and Fr–En models always lead to results above, or equal to the state of the art. For similar languages such as Spanish, French, and Italian, the accuracy increase over the state of the art is up to 14.8% (from 56.6% to 65.0%) and 13.4% (from 56.6% to 64.2%) when the Fr–En and Es–En models are respectively used to label the It–En dataset. Although not always statistically significant and below the performance obtained in the ideal scenario where CLTE training data are available (*full sys.*), such improvements suggest that our features can be re-used, at least to some extent, across different language settings. As expected, the major incompatibilities arise between German and the other languages due to the linguistic differences between this language and the others. However, it is interesting to note that: *i*) at least in one case (*i.e.* when tested on It–En) the De–En model still achieves results above the state of the art, and *ii*) on the De–En evaluation setting the worst model (Fr–En) still achieves state of the art results.

The results obtained with the voting scheme suggest that our models can complement each other when used on a new language pair. Although statistically significant only over It–En data, voting results both outperform the state of the art and the results achieved by single models.

## 4 Conclusion

We investigated the usefulness of qualitative information from automatic word alignment to identify semantic relations between sentences in different languages. With coherent results in CLTE, we demonstrated that features considering the importance of aligned terms can successfully integrate the quantitative evidence (number and proportion of aligned terms) used by previous supervised learning approaches. A study on the portability across languages of the learned models demonstrated that word alignment information can be exploited to train reusable models for new language combinations where bitexts are available but CLTE labeled data are not.

## Acknowledgments

This work has been partially supported by the EC-funded projects CoSyne (FP7-ICT-4-248531) and MateCat (ICT-2011.4.2–287688), and by Spanish Government through projects TIN2009-14009-C02-01 and TIN2012-32615.

## References

- John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation. Summer workshop final report, JHU/CLSP.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.
- Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT’12)*, pages 10–51, Montréal, Canada.
- Qin Gao and Stephan Vogel. 2008. Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, USA.

- 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.
- Sergio Jimenez, Claudia Bécerra, and Alexander Gelbukh. 2012. Soft Cardinality + ML: Learning Adaptive Similarity Functions for Cross-lingual Textual Entailment. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, pages 684–688, Montréal, Canada.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.
- Philip Koehn. 2005. Europarl: a Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand.
- Alessandro Lenci, Roberto Bartolini, Nicoletta Calzolari, Ana Agua, Stephan Busemann, Emmanuel Cartier, Karine Chevreau, and José Coch. 2002. Multilingual summarization by integrating linguistic resources in the MLIS-MUSI Project. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 1464–1471, Las Palmas de Gran Canaria, Spain.
- Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010. Towards Cross-Lingual Textual Entailment. In *Proceedings of the Eleventh Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, pages 321–324, Los Angeles, California, USA.
- Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012a. Detecting Semantic Equivalence and Information Disparity in Cross-lingual Documents. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 120–124, Jeju Island, Korea.
- Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012b. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the Machine Translation Workshop (WMT2012)*, Montréal, Canada.
- Christoph Monz, Vivi Nastase, Matteo Negri, Angela Fahrni, Yashar Mehdad, and Michael Strube. 2011. CoSyne: a Framework for Multilingual Content Synchronization of Wikis. In *Proceedings of WikiSym 2011, the International Symposium on Wikis and Open Collaboration*, pages 217–218, Mountain View, California, USA.
- Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and Conquer: Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, Edinburgh, Scotland.
- Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2012. Semeval-2012 Task 8: Cross-Lingual Textual Entailment for Content Synchronization. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, pages 399–407, Montréal, Canada.
- Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2013. Semeval-2013 Task 8: Cross-Lingual Textual Entailment for Content Synchronization. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, GA.
- Franz J. Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Sebastian Padó, 2006. *User's guide to sigf: Significance testing by approximate randomisation*.
- Kristen Parton. 2012. *Lost and Found in Translation: Cross-Lingual Question Answering with Result Translation*. Ph.D. thesis, Columbia University.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey.
- Gerard Salton and Christopher Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523.
- Helmut Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, Dublin, Ireland.
- Hristo Tanev, Milen Kouylekov, Bernardo Magnini, Matteo Negri, and Kiril Simov. 2006. Exploiting Linguistic Indices and Syntactic Structures for Multilingual Question Answering: ITC-irst at CLEF 2005. *Accessing Multilingual Information Repositories*, pages 390–399.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based Word Alignment in Statistical Translation. In *Proceedings of the 16th International Conference on Computational Linguistics (ACL'96)*, pages 836–841, Copenhagen, Denmark.
- Katharina Wäschle and Sascha Fendrich. 2012. HDU: Cross-lingual Textual Entailment with SMT Features. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, pages 467–471, Montréal, Canada.



# An Information Theoretic Approach to Bilingual Word Clustering

Manaal Faruqui and Chris Dyer

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA, 15213, USA

{mfaruqui, cdyer}@cs.cmu.edu

## Abstract

We present an information theoretic objective for bilingual word clustering that incorporates both monolingual distributional evidence as well as cross-lingual evidence from parallel corpora to learn high quality word clusters jointly in any number of languages. The monolingual component of our objective is the average mutual information of clusters of adjacent words in each language, while the bilingual component is the average mutual information of the aligned clusters. To evaluate our method, we use the word clusters in an NER system and demonstrate a statistically significant improvement in  $F_1$  score when using bilingual word clusters instead of monolingual clusters.

## 1 Introduction

A word cluster is a group of words which ideally captures syntactic, semantic, and distributional regularities among the words belonging to the group. Word clustering is widely used to reduce the number of parameters in statistical models which leads to improved generalization (Brown et al., 1992; Kneser and Ney, 1993; Clark, 2003; Koo et al., 2008; Turian et al., 2010), and multilingual clustering has been proposed as a means to improve modeling of translational correspondences and to facilitate projection of linguistic resource across languages (Och, 1999; Täckström et al., 2012). In this paper, we argue that generally more informative clusters can be learned when evidence from multiple languages is considered while creating the clusters.

We propose a novel bilingual word clustering objective (§2). The first term deals with each

language independently and ensures that the data is well-explained by the clustering in a sequence model (§2.1). The second term ensures that the cluster alignments induced by a word alignment have high mutual information across languages (§2.2). Since the objective consists of terms representing the entropy monolingual data (for each language) and parallel bilingual data, it is particularly attractive for the usual situation in which there is much more monolingual data available than parallel data. Because of its similarity to the variation of information metric (Meilă, 2003), we call this bilingual term in the objective the **aligned variation of information**.

## 2 Word Clustering

A word clustering  $\mathcal{C}$  is a partition of a vocabulary  $\Sigma = \{x_1, x_2, \dots, x_{|\Sigma|}\}$  into  $K$  disjoint subsets,  $C_1, C_2, \dots, C_K$ . That is,  $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ ;  $C_i \cap C_j = \emptyset$  for all  $i \neq j$  and  $\bigcup_{k=1}^K C_k = \Sigma$ .

### 2.1 Monolingual objective

We use the average surprisal in a probabilistic sequence model to define the monolingual clustering objective. Let  $c_i$  denote the word class of word  $w_i$ . Our objective assumes that the probability of a word sequence  $\mathbf{w} = \langle w_1, w_2, \dots, w_M \rangle$  is

$$p(\mathbf{w}) = \prod_{i=1}^M p(c_i | c_{i-1}) \times p(w_i | c_i), \quad (2.1)$$

where  $c_0$  is a special start symbol. The term  $p(c_i | c_{i-1})$  is the probability of class  $c_i$  following class  $c_{i-1}$ , and  $p(w_i | c_i)$  is the probability of class  $c_i$  emitting word  $w_i$ . Using the MLE estimates after taking the negative logarithm, this term reduces to

the following as shown in (Brown et al., 1992):

$$H(\mathcal{C}; \mathbf{w}) = 2 \sum_{k=1}^K \frac{\#(C_k)}{M} \log \frac{\#(C_k)}{M} - \sum_i \sum_{j \neq i} \frac{\#(C_i, C_j)}{M} \log \frac{\#(C_i, C_j)}{M}$$

where  $\#(C_k)$  is the count of  $C_k$  in the corpus  $\mathbf{w}$  under the clustering  $\mathcal{C}$ ,  $\#(C_i, C_j)$  is the count of the number of times that cluster  $C_i$  precedes  $C_j$  and  $M$  is the size of the corpus. Using the monolingual objective to cluster, we solve the following search problem:

$$\hat{\mathcal{C}} = \arg \min_{\mathcal{C}} H(\mathcal{C}; \mathbf{w}). \quad (2.2)$$

## 2.2 Bilingual objective

Now let us suppose we have a second language with vocabulary  $\Omega = \{y_1, y_2, \dots, y_{|\Omega|}\}$ , which is clustered into  $K$  disjoint subsets  $\mathcal{D} = \{D_1, D_2, \dots, D_K\}$ , and a corpus of text in the second language,  $\mathbf{v} = \langle v_1, v_2, \dots, v_N \rangle$ . Obviously we can cluster both languages using the monolingual objective above:

$$\hat{\mathcal{C}}, \hat{\mathcal{D}} = \arg \min_{\mathcal{C}, \mathcal{D}} H(\mathcal{C}; \mathbf{w}) + H(\mathcal{D}; \mathbf{v}).$$

This joint minimization for the clusterings for both languages clearly has no benefit since the two terms of the objective are independent. We must alter the object by further assuming that we have *a priori* beliefs that some of the words in  $\mathbf{w}$  and  $\mathbf{v}$  have the same meaning.

To encode this belief, we introduce the notion of a **weighted vocabulary alignment**  $\mathcal{A}$ , which is a function on pairs of words in vocabularies  $\Sigma$  and  $\Omega$  to a value greater than or equal to 0, i.e.,  $\mathcal{A} : \Sigma \times \Omega \mapsto \mathbb{R}_{\geq 0}$ . For concreteness,  $\mathcal{A}(x, y)$  will be the number of times that  $x$  is aligned to  $y$  in a word aligned parallel corpus. By abuse of notation, we write marginal weights  $\mathcal{A}(x) = \sum_{y \in \Omega} \mathcal{A}(x, y)$  and  $\mathcal{A}(y) = \sum_{x \in \Sigma} \mathcal{A}(x, y)$ . We also define the set marginals  $\mathcal{A}(C, D) = \sum_{x \in C} \sum_{y \in D} \mathcal{A}(x, y)$ .

Using this weighted vocabulary alignment, we state an objective that encourages clusterings to have high average mutual information when alignment links are followed; that is, on average how much information does knowing the cluster of a word  $x \in \Sigma$  impart about the clustering of  $y \in \Omega$ , and vice-versa?

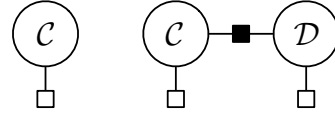


Figure 1: Factor graphs of the monolingual (left) & proposed bilingual clustering problem (right).

We call this quantity the **aligned variation of information** (AVI).

$$\text{AVI}(\mathcal{C}, \mathcal{D}; \mathcal{A}) = \mathbb{E}_{\mathcal{A}(x,y)} [-\log p(c_x | d_y) - \log p(d_y | c_x)]$$

Writing out the expectation and gathering terms, we obtain

$$\text{AVI}(\mathcal{C}, \mathcal{D}; \mathcal{A}) = - \sum_{x \in \Sigma} \sum_{y \in \Omega} \frac{\mathcal{A}(x, y)}{\mathcal{A}(\cdot, \cdot)} \times \left[ 2 \log \frac{\mathcal{A}(C, D)}{\mathcal{A}(\cdot, \cdot)} - \log p(C) - \log p(D) \right],$$

where it is assumed that  $0 \log x = 0$ .

Our bilingual clustering objective can therefore be stated as the following search problem over a linear combination of the monolingual and bilingual objectives:

$$\arg \min_{\mathcal{C}, \mathcal{D}} \underbrace{H(\mathcal{C}; \mathbf{w}) + H(\mathcal{D}; \mathbf{v})}_{\text{monolingual}} + \underbrace{\beta \text{AVI}(\mathcal{C}, \mathcal{D})}_{\beta \times \text{bilingual}}. \quad (2.3)$$

**Understanding AVI.** Intuitively, we can imagine sampling a random alignment from the distribution obtained by normalizing  $\mathcal{A}(\cdot, \cdot)$ . AVI gives us a measure of how much information do we obtain, on average, from knowing the cluster in one language about the clustering of a linked element chosen at random proportional to  $\mathcal{A}(x, \cdot)$  (or conditioned the other way around). In the following sections, we denote  $\text{AVI}(\mathcal{C}, \mathcal{D}; \mathcal{A})$  by  $\text{AVI}(\mathcal{C}, \mathcal{D})$ . To further understand AVI, we remark that AVI reduces to the VI metric when the alignment maps words to themselves in the same language. As a proper metric, VI has a number of attractive properties, and these can be generalized to AVI (without restriction on the alignment map), namely:

- *Non-negativity:*  $\text{AVI}(C, D) \geq 0$ ;
- *Symmetry:*  $\text{AVI}(C, D) = \text{AVI}(D, C)$ ;
- *Triangle inequality:*  $\text{AVI}(C, D) + \text{AVI}(D, E) \geq \text{AVI}(C, E)$ ;

- *Identity of indiscernibles:*  
 $AVI(C, D) = 0$  iff  $C \equiv D$ .<sup>1</sup>

### 2.3 Example

Figure 2 provides an example illustrating the difference between the bilingual vs. monolingual clustering objectives. We compare two different clusterings of a two-sentence Arabic-English parallel corpus (the English half of the corpus contains the same sentence, twice, while the Arabic half has two variants with the same meaning). While English has a relatively rigid SVO word order, Arabic can alternate between the traditional VSO order and an more modern SVO order. Since our monolingual clustering objective relies exclusively on the distribution of clusters before and after each token, flexible word order alternations like this can cause unintuitive results. To further complicate matters, verbs can inflect differently depending on whether their subject precedes or follows them (Haywood and Nahmad, 1999), so a monolingual model, which knows nothing about morphology and may only rely on distributional clues, has little chance of performing well without help. This is indeed what we observe in the monolingual objective optimal solution (center), in which  $Aw1Ad$  (*boys*) and  $yElbwn$  (*play+PRES + 3PL*) are grouped into a single class, while  $yElb$  (*play+PRES + 3SG*) is in its own class. However, the AVI term (which is of course not included) has a value of 1.0, reflecting the relatively disordered clustering relative to the given alignment. On the right, we see the optimal solution that includes the AVI term in the clustering objective. This has an AVI of 0, indicating that knowing the clustering of any word is completely informative about the words it is aligned to. By including this term, a slightly worse monolingual solution is chosen, but the clustering corresponds to the reasonable intuition that words with the same meaning (i.e., the two variants of *to play*) should be clustered together.

### 2.4 Inference

Figure 1 shows the factor graph representation of our clustering models. Finding the optimal clustering under both the monolingual and bilingual objectives is a computationally hard combinatorial optimization problem (Och, 1995). We use a greedy hill-climbing word exchange algorithm (Martin et al., 1995) to find a minimum

<sup>1</sup> $C \equiv D$  iff  $\forall i |\{D(y) | \forall (x, y) \in \mathcal{A}, C(x) = i\}| = 1$

value for our objective. We terminate the optimization procedure when the number of words exchanged at the end of one complete iteration through both the languages is less than 0.1% of the sum of vocabulary of the two languages and at least five complete iterations have been completed.<sup>2</sup> For every language the word clusters are initialised in a round robin order according to the token frequency.

## 3 Experiments

Evaluation of clustering is not a trivial problem. One branch of work seeks to recast the problem as the of part-of-speech (POS) induction and attempts to match linguistic intuitions. However, hard clusters are particularly useful for downstream tasks (Turian et al., 2010). We therefore chose to focus our evaluation on the latter problem. For our evaluation, we use our word clusters as an input to a named entity recognizer which uses these clusters as a source of features. Our evaluation task is the German corpus with NER annotation that was created for the shared task at CoNLL-2003<sup>3</sup>. The training set contains approximately 220,000 tokens and the development set and test set contains 55,000 tokens each. We use Stanford’s Named Entity Recognition system<sup>4</sup> which uses a linear-chain conditional random field to predict the most likely sequence of NE labels (Finkel and Manning, 2009).

**Corpora for Clustering:** We used parallel corpora for {Arabic, English, French, Korean & Turkish}-German pairs from WIT-3 corpus (Cetolo et al., 2012)<sup>5</sup>, which is a collection of translated transcriptions of TED talks. Each language pair contained around 1.5 million German words. The corpus was word aligned in two directions using an unsupervised word aligner (Dyer et al., 2013), then the intersected alignment points were taken.

**Monolingual Clustering:** For every language pair, we train German word clusters on the monolingual German data from the parallel data. Note that the parallel corpora are of different sizes and hence the monolingual German data from every parallel corpus is different. We treat the  $F_1$  score

<sup>2</sup>In practice, the number of exchanged words drops of exponentially, so this threshold is typically reached in not many iterations.

<sup>3</sup><http://www.cnts.ua.ac.be/conll2003/ner/>

<sup>4</sup><http://nlp.stanford.edu/ner/index.shtml>

<sup>5</sup><https://wit3.fbk.eu/mt.php?release=2012-03>

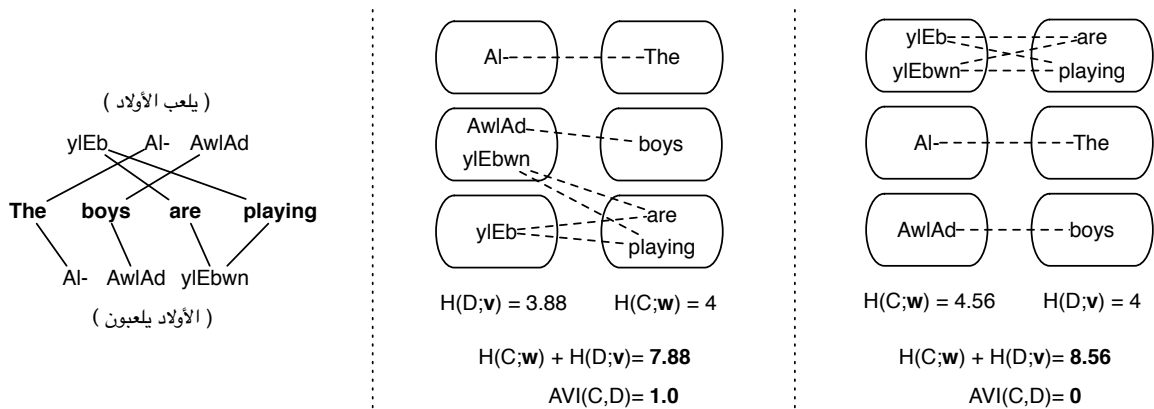


Figure 2: A two-sentence English-Arabic parallel corpus (left); a 3-class clustering that maximizes the monolingual objective ( $\beta = 0$ ; center); and a 3-class clustering that maximizes the joint monolingual and bilingual objective (any  $\beta > 0.68$ ; right).

obtained using monolingual word clusters ( $\beta = 0$ ) as the baseline. Table 1 shows the  $F_1$  score of NER<sup>6</sup> when trained on these monolingual German word clusters.

**Bilingual Clustering:** While we have formulated a joint objective that enables using both monolingual and bilingual evidence, it is possible to create word clusters using the bilingual signal only by removing the first term in Eq. 2.3. Table 1 shows the performance of NER when the word clusters are obtained using only the bilingual information for different language pairs. As can be seen, these clusters are helpful for all the language pairs. For *Turkish* the  $F_1$  score improves by 1.0 point over when there are no distributional clusters which clearly shows that the word alignment information improves the clustering quality. We now need to supplement the bilingual information with monolingual information to see if the improvement sustains.

We varied the weight of the bilingual objective ( $\beta$ ) from 0.05 to 0.9 and observed the effect in NER performance on English-German language pair. The  $F_1$  score is maximum for  $\beta = 0.1$  and decreases monotonically when  $\beta$  is either increased or decreased. This indicates that bilingual information is helpful, but less valuable than monolingual information. Preliminary experiments showed that the value of  $\beta = 0.1$  is fairly robust across other language pairs and hence we fix it to that for all the experiments.

We run our bilingual clustering model ( $\beta =$

<sup>6</sup>Faruqui and Padó (2010) show that for the size of our generalization data in German-NER,  $K = 100$  should give us the optimum value.

0.1) across all language pairs and note the  $F_1$  scores. Table 1 (unrefined) shows that except for Arabic-German & French-German, all other language pairs deliver a better  $F_1$  score than only using monolingual German data. In case of Arabic-German there is a drop in score by 0.25 points. Although, we have observed improvement in  $F_1$  score over the monolingual case, the gains do not reach significance according to McNemar’s test (Dietterich, 1998).

Thus we propose to further refine the quality of word alignment links as follows: Let  $x$  be a word in language  $\Sigma$  and  $y$  be a word in language  $\Omega$  and let there exists an alignment link between  $x$  and  $y$ . Recall that  $\mathcal{A}(x, y)$  is the count of the alignment links between  $x$  and  $y$  observed in the parallel data, and  $\mathcal{A}(x)$  and  $\mathcal{A}(y)$  are the respective marginal counts. Then we define an edge association weight  $e(x, y) = \frac{2 \times \mathcal{A}(x, y)}{\mathcal{A}(x) + \mathcal{A}(y)}$ . This quantity is an association of the strength of the relationship between  $x$  and  $y$ , and we use it to remove all alignment links whose  $e(x, y)$  is below a given threshold before running the bilingual clustering model. We vary  $e$  from 0.1 to 0.7 and observe the new  $F_1$  scores on the development data. Table 1 (refined) shows the results obtained by our refined model. The values shown in bold are the highest improvements over the monolingual model.

For English and Turkish we observe a statistically significant improvement over the monolingual model (cf. Table 1) with  $p < 0.007$  and  $p < 0.001$  according to McNemar’s test. Arabic improves least with just an improvement of 0.02  $F_1$  points over the monolingual baseline. We

| Language Pair | Dev            |                            |                              |                            | Test                       |                            |
|---------------|----------------|----------------------------|------------------------------|----------------------------|----------------------------|----------------------------|
|               | —<br>(only bi) | $\beta = 0$<br>(only mono) | $\beta = 0.1$<br>(unrefined) | $\beta = 0.1$<br>(refined) | $\beta = 0$<br>(only mono) | $\beta = 0.1$<br>(refined) |
| No clusters   |                |                            | 68.27                        |                            | 72.32                      |                            |
| En-De         | 68.95          | 70.04                      | <b>70.33</b>                 | <b>70.64</b> <sup>†</sup>  | 72.30                      | <b>72.98</b> <sup>†</sup>  |
| Fr-De         | 69.16          | 69.74                      | 69.69                        | <b>69.89</b>               | 72.66                      | <b>72.83</b>               |
| Ar-De         | 69.01          | 69.65                      | 69.40                        | <b>69.67</b>               | 72.90                      | 72.37                      |
| Tr-De         | 69.29          | 69.46                      | <b>69.64</b>                 | <b>70.05</b> <sup>†</sup>  | 72.41                      | <b>72.54</b>               |
| Ko-De         | 68.95          | 69.70                      | <b>69.78</b>                 | <b>69.95</b>               | 72.71                      | 72.54                      |
| Average       | 69.07          | 69.71                      | <b>69.76</b>                 | <b>70.04</b> <sup>†</sup>  | 72.59                      | <b>72.65</b>               |

Table 1: NER performance using different word clustering models. Bold indicates an improvement over the monolingual ( $\beta = 0$ ) baseline; † indicates a significant improvement (McNemar’s test,  $p < 0.01$ ).

see that the optimal value of  $e$  changes from one language pair to another. For French and English  $e = 0.1$  gives the best results whereas for Turkish and Arabic  $e = 0.5$  and for Korean  $e = 0.7$ . Are these thresholds correlated with anything? We suggest that higher values of  $e$  correspond to more intrinsically noisy alignments. Since alignment models are parameterized based on the vocabularies of the languages they are aligning, larger vocabularies are more prone to degenerate solutions resulting from overfitting. So we are not surprised to see that sparser alignments (resulting from higher values of  $e$ ) are required by languages like Korean, while languages like French and English make due with denser alignments.

**Evaluation on Test Set:** We now verify our results on the test set. We take the best bilingual word clustering model obtained for every language pair ( $e = 0.1$  for En, Fr.  $e = 0.5$  for Ar, Tr.  $e = 0.7$  for Ko) and train NER classifiers using these. Table 1 shows the performance of German NER classifiers on the test set. All the values shown in bold are better than the monolingual baselines. English again has a statistically significant improvement over the baseline. French and Turkish show the next best improvements. The English-German cluster model performs better than the `mkcls`<sup>7</sup> tool (72.83%).

## 4 Related Work

Our monolingual clustering model is purely distributional in nature. Other extensions to word clustering have incorporated morphological and orthographic information (Clark, 2003). The work of Snyder and Barzilay (2010), which focused on POS induction is very closely related. The earliest work on bilingual word clustering was proposed by (Och, 1999) which, like us, uses a lan-

guage modeling approach (Brown et al., 1992; Kneser and Ney, 1993) for monolingual optimization and a similarity function for bilingual similarity. Täckström et al. (2012) use cross-lingual word clusters to show transfer of linguistic structure. While their clustering method is superficially similar, the objective function is more heuristic in nature than our information-theoretic conception of the problem. Multilingual learning has been applied to a number of unsupervised and supervised learning problems, including word sense disambiguation (Diab, 2003; Guo and Diab, 2010), topic modeling (Mimno et al., 2009; Boyd-Graber and Blei, 2009), and morphological segmentation (Snyder and Barzilay, 2008).

Also closely related is the technique of cross-lingual annotation projection. This has been applied to bootstrapping syntactic parsers (Hwa et al., 2005; Smith and Smith, 2007; Cohen et al., 2011), morphology (Fraser, 2009), tense (Schiehlen, 1998) and T/V pronoun usage (Faruqui and Padó, 2012).

## 5 Conclusions

We presented a novel information theoretic model for bilingual word clustering which seeks a clustering with high average mutual information between clusters of adjacent words, and also high mutual information across observed word alignment links. We have shown that improvement in clustering can be obtained across a range of language pairs, evaluated in terms of their value as features in an extrinsic NER task. Our model can be extended for clustering any number of given languages together in a joint framework, and incorporate both monolingual and parallel data.

**Acknowledgement:** We would like to thank W. Ammar, V. Chahuneau and W. Ling for valuable discussions.

<sup>7</sup><http://www.statmt.org/moses/giza/mkcls.html>

## References

- J. Boyd-Graber and D. M. Blei. 2009. Multilingual topic models for unaligned text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 75–82, Arlington, Virginia, United States. AUAI Press.
- P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December.
- M. Cettolo, C. Girardi, and M. Federico. 2012. Wit<sup>3</sup>: Web inventory of transcribed and translated talks. In *Proceedings of the 16<sup>th</sup> Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.
- A. 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.
- S. B. Cohen, D. Das, and N. A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 50–61, Stroudsburg, PA, USA. Association for Computational Linguistics.
- M. T. Diab. 2003. *Word sense disambiguation within a multilingual framework*. Ph.D. thesis, University of Maryland at College Park, College Park, MD, USA. AAI3115805.
- T. G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.
- C. Dyer, V. Chahuneau, and N. A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. NAACL*.
- M. Faruqui and S. Padó. 2010. Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany.
- M. Faruqui and S. Padó. 2012. Towards a model of formal and informal address in english. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- J. R. Finkel and C. D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 141–150, Stroudsburg, PA, USA. Association for Computational Linguistics.
- A. Fraser. 2009. Experiments in morphosyntactic processing for translating to and from German. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 115–119, Athens, Greece, March. Association for Computational Linguistics.
- W. Guo and M. Diab. 2010. Combining orthogonal monolingual and multilingual sources of evidence for all words wsd. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1542–1551, Stroudsburg, PA, USA. Association for Computational Linguistics.
- J. A. Haywood and H. M. Nahmad. 1999. *A new Arabic grammar of the written language*. Lund Humphries Publishers.
- R. Hwa, P. Resnik, A. Weinberg, C. I. Cabezas, and O. Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, pages 311–325.
- R. Kneser and H. Ney. 1993. Forming word classes by statistical clustering for statistical language modelling. In R. Khler and B. Rieger, editors, *Contributions to Quantitative Linguistics*, pages 221–226. Springer Netherlands.
- T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL*.
- S. Martin, J. Liermann, and H. Ney. 1995. Algorithms for bigram and trigram word clustering. In *Speech Communication*, pages 1253–1256.
- M. Meilă. 2003. Comparing Clusterings by the Variation of Information. In *Learning Theory and Kernel Machines*, pages 173–187.
- D. Mimno, H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. 2009. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 880–889, Stroudsburg, PA, USA. Association for Computational Linguistics.
- F. J. Och. 1995. Maximum-Likelihood-Schätzung von Wortkategorien mit Verfahren der kombinatorischen Optimierung. Studienarbeit, University of Erlangen.
- F. J. Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, EACL '99, pages 71–76, Stroudsburg, PA, USA. Association for Computational Linguistics.
- M. Schiehlen. 1998. Learning tense translation from bilingual corpora.
- D. A. Smith and N. A. Smith. 2007. Probabilistic Models of Nonprojective Dependency Trees. In *Proceedings of the 2007 Joint Conference on*

*Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 132–140, Prague, Czech Republic, June. Association for Computational Linguistics.

- B. Snyder and R. Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *In The Annual Conference of the Association for Computational Linguistics*.
- B. Snyder and R. Barzilay. 2010. Climbing the tower of babel: Unsupervised multilingual learning. In J. Frnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 29–36. Omnipress.
- O. Täckström, R. McDonald, and J. Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, page 11. Association for Computational Linguistics.
- J. Turian, L. Ratinov, and Y. 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 '10*, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Building and Evaluating a Distributional Memory for Croatian

Jan Šnajder\* Sebastian Padó† Željko Agić‡

\*University of Zagreb, Faculty of Electrical Engineering and Computing  
Unska 3, 10000 Zagreb, Croatia

†Heidelberg University, Institut für Computerlinguistik  
69120 Heidelberg, Germany

‡University of Zagreb, Faculty of Humanities and Social Sciences  
Ivana Lučića 3, 10000 Zagreb, Croatia

jan.snajder@fer.hr pado@cl.uni-heidelberg.de zagic@ffzg.hr

## Abstract

We report on the first structured distributional semantic model for Croatian, DM.HR. It is constructed after the model of the English Distributional Memory (Baroni and Lenci, 2010), from a dependency-parsed Croatian web corpus, and covers about 2M lemmas. We give details on the linguistic processing and the design principles. An evaluation shows state-of-the-art performance on a semantic similarity task with particularly good performance on nouns. The resource is freely available.

## 1 Introduction

Most current work in lexical semantics is based on the *Distributional Hypothesis* (Harris, 1954), which posits a correlation between the degree of words' semantic similarity and the similarity of the contexts in which they occur. Using this hypothesis, word meaning representations can be extracted from large corpora. Words are typically represented as vectors whose dimensions correspond to context features. The vector similarities, which are interpreted as semantic similarities, are used in numerous applications (Turney and Pantel, 2010).

Most vector spaces in current use are either *word-based* (co-occurrence defined by surface window, context words as dimensions) or *syntax-based* (co-occurrence defined syntactically, syntactic objects as dimensions). Syntax-based models have several desirable properties. First, they are model to fine-grained types of semantic similarity such as predicate-argument plausibility (Erk et al., 2010). Second, they are more versatile – Baroni and Lenci (2010) have presented a generic framework, the Distributional Memory (DM), which is applicable

to a wide range of tasks beyond word similarity. Third, they avoid the “syntactic assumption” inherent in word-based models, namely that context words are relevant iff they are in an  $n$ -word window around the target. This property is particularly relevant for free word order languages with many long distance dependencies and non-projective structure (Kübler et al., 2009). Their obvious problem, of course, is that they require a large parsed corpus.

In this paper, we describe the construction of a Distributional Memory for Croatian (DM.HR), a free word order language. To do so, we parse hrWaC (Ljubešić and Erjavec, 2011), a 1.2B-token Croatian web corpus. We evaluate DM.HR on a synonym choice task, where it outperforms the standard bag-of-words model for nouns and verbs.

## 2 Related Work

Vector space semantic models have been applied to a number of Slavic languages, including Bulgarian (Nakov, 2001a), Czech (Smrž and Rychlý, 2001), Polish (Piasecki, 2009; Broda et al., 2008; Broda and Piasecki, 2008), and Russian (Nakov, 2001b; Mitrofanova et al., 2007). Previous work on distributional semantic models for Croatian dealt with similarity prediction (Ljubešić et al., 2008; Janković et al., 2011) and synonym detection (Karan et al., 2012), however using only word-based and not syntactic-based models.

So far the only DM for a language other than English is the German DM.DE by Padó and Utt (2012), who describe the process of building DM.DE and the evaluation on a synonym choice task. Our work is similar, though each language has its own challenges. Croatian, like other Slavic languages, has rich inflectional morphology and free word order, which lead to errors in linguistic processing and affect the quality of the DM.



### 3 Distributional Memory

DM represents co-occurrence information in a general, non-task-specific manner, as a tensor, i.e., a three-dimensional matrix, of weighted *word-link-word* tuples. Each tuple is mapped onto a number by scoring function  $\sigma: W \times L \times W \rightarrow \mathbb{R}^+$ , that reflects the strength of the association. When a particular task is selected, a vector space for this task can be generated from the tensor by matricization. Regarding the examples from Section 1, synonym discovery would use a *word by link-word* space ( $W \times LW$ ), which contains vectors for words  $w$  represented by pairs  $\langle l, w \rangle$  of a link and a context word. Analogy discovery would use a *word-word by link* space ( $WW \times L$ ), which represents word pairs  $\langle w_1, w_2 \rangle$  by vectors over links  $l$ .

The links can be chosen to model any relation of interest between words. However, as noted by Padó and Utt (2012), dependency relations are the most obvious choice. Baroni and Lenci (2010) introduce three dependency-based DM variants: DepDM, LexDM, and TypeDM. DepDM uses links that correspond to dependency relations, with sub-categorization for subject (*subj\_tr* and *subj\_intr*) and object (*obj* and *iobj*). Furthermore, all prepositions are lexicalized into links (e.g.,  $\langle \text{sun, on, Sunday} \rangle$ ). Finally, the tensor is symmetrized: for each tuple  $\langle w_1, l, w_2 \rangle$ , its inverse  $\langle w_2, l^{-1}, w_1 \rangle$  is included. The other two variants are more complex: LexDM uses more lexicalized links, encoding, e.g., lexical material between the words, while TypeDM extends LexDM with a scoring function based on lexical variability.

Following the work of Padó and Utt (2012), we build a DepDM variant for DM.HR. Although Baroni and Lenci (2010) show that TypeDM can outperform the other two variants, DepDM often performs at a comparable level, while being much simpler to build and more efficient to compute.

### 4 Building DM.HR

To build DM.HR, we need to collect co-occurrence counts from a corpus. Since no sufficiently large suitable corpus exists for Croatian, we first explain how we preprocessed, tagged, and parsed the data.

**Corpus and preprocessing.** We adopted hrWaC, the 1.2B-token Croatian web corpus (Ljubešić and Erjavec, 2011), as starting point. hrWaC was built with the aim of obtaining a cleaner-than-usual web corpus. To this end, a conservative boilerplate re-

moval procedure was used; Ljubešić and Erjavec (2011) report a precision of 97.9% and a recall of 70.7%. Nonetheless, our inspection revealed that, apart from the unavoidable spelling and grammatical errors, hrWaC still contains non-textual content (e.g., code snippets and formatting structure), encoding errors, and foreign-language content. As this severely affects linguistic processing, we additionally filtered the corpus.

First, we removed from hrWaC the content crawled from main discussion forum and blog websites. This content is highly ungrammatical and contains a lot of non-diacriticized text, typical for user-generated content. This step alone removed one third of the data. We processed the remaining content with a tokenizer and a sentence segmenter based on regular expressions, obtaining 66M sentences. Next, we applied a series of heuristic filters at the document- and sentence-level. At the document level, we discard all documents (1) whose length is below a specified threshold, (2) contain no diacritics, (3) contain no words from a list of frequent Croatian words, or (4) contain a single word from lists of distinctive foreign-language words (for Serbian). The last two steps serve to eliminate foreign-language content. In particular, the last step serves to filter out the text in Serbian, which at the sentence-level is difficult to automatically discriminate from Croatian. At the sentence-level, we discard sentences that are (1) shorter than a specified threshold, (2) contain non-standard symbols, (3) contain non-diacriticized Croatian words, or (4) contain too many foreign words from a list of foreign-language words (for English and Slovene). The last step filters out specifically the sentences in English and Slovene, as we found that these often occur mixed with text in Croatian. The final filtered version of hrWaC contains 51M sentences and 1.2B tokens. The corpus is freely available for download, along with a more detailed description of the preprocessing steps.<sup>1</sup>

**Tagging, lemmatization, and parsing.** For morphosyntactic (MSD) tagging, lemmatization, and dependency parsing of hrWaC, we use freely available tools with models trained on the new SETimes Corpus of Croatian (SETIMES.HR), based on the Croatian part of the SETimes parallel corpus.<sup>2</sup> SETIMES.HR and the derived tools are prototypes

<sup>1</sup><http://takelab.fer.hr/data>

<sup>2</sup><http://www.nljubesic.net/resources/corpora/setimes/>

|                   | SETIMES.HR | Wikipedia |
|-------------------|------------|-----------|
| HunPos (POS only) | 97.1       | 94.1      |
| HunPos (full MSD) | 87.7       | 81.5      |
| CST lemmatizer    | 97.7       | 96.5      |
| MSTParser         | 77.5       | 68.8      |

Table 1: Tagging, lemmatization, and parsing accuracy

that are about to be released as parts of another work. Here we give a general description and a re-evaluation that we consider relevant for building DM.HR.

SETIMES.HR consists of 90K tokens and 4K sentences, manually lemmatized and MSD-tagged according to Multext East v4 tagset (Erjavec, 2012), with the help of the Croatian Lemmatization Server (Tadić, 2005). It is used also as a basis for a novel formalism for syntactic annotation and dependency parsing of Croatian (Agić and Merkler, 2013).

On the basis of previous evaluation for Croatian (Agić et al., 2008; Agić et al., 2009; Agić, 2012) and availability and licensing considerations, we chose HunPos tagger (Halácsy et al., 2007), CST lemmatizer (Ingason et al., 2008), and MSTParser (McDonald et al., 2006) to process hrWaC. We evaluated the tools on 100-sentence test sets from SETIMES.HR and Wikipedia; performance on Wikipedia should be indicative of the performance on a cross-domain dataset, such as hrWaC. In Table 1 we show lemmatization and tagging accuracy, as well as dependency parsing accuracy in terms of labeled attachment score (LAS). The results show that lemmatization, tagging and parsing accuracy improves on the state of the art for Croatian. The SETIMES.HR dependency parsing models are publicly available.<sup>3</sup>

**Syntactic patterns.** We collect the co-occurrence counts of tuples using a set of syntactic patterns. The patterns effectively define the link types, and hence the dimensions of the semantic space. Similar to previous work, we use two sorts of links: unlexicalized and lexicalized.

For unlexicalized links, we use ten syntactic patterns. These correspond to the main dependency relations produced by our parser: *Pred* for predicates, *Atr* for attributes, *Adv* for adverbs, *Atv* for verbal complements, *Obj* for objects, *Prep* for prepositions, and *Pnom* for nominal predicates. We subcategorized the subject relation into *Sub\_tr* (sub-

<sup>3</sup><http://zeljko.agic.me/resources/>

| Link                 | P (%)       | R (%)       | F <sub>1</sub> (%) |
|----------------------|-------------|-------------|--------------------|
| <b>Unlexicalized</b> |             |             |                    |
| <i>Adv</i>           | 57.3        | 52.7        | 54.9               |
| <i>Atr</i>           | 85.0        | 89.3        | 87.1               |
| <i>Atv</i>           | 75.3        | 70.9        | 73.1               |
| <i>Obj</i>           | 71.4        | 71.7        | 71.5               |
| <i>Pnom</i>          | 55.7        | 50.8        | 53.1               |
| <i>Pred</i>          | 81.8        | 70.6        | 75.8               |
| <i>Prep</i>          | 50.0        | 28.6        | 36.4               |
| <i>Sb_tr</i>         | 67.8        | 73.8        | 70.7               |
| <i>Sb_intr</i>       | 64.5        | 64.8        | 64.7               |
| <i>Verb</i>          | 61.6        | 73.6        | 67.1               |
| <b>Lexicalized</b>   |             |             |                    |
| Prepositions         | 67.2        | 67.9        | 67.5               |
| Verbs                | 61.6        | 73.6        | 67.1               |
| <b>All links</b>     | <b>73.7</b> | <b>75.5</b> | <b>74.6</b>        |

Table 2: Tuple extraction performance on SETIMES.HR

jects of transitive verbs) and *Sub\_intr* (subject of intransitive verbs). The motivation for this is better modeling of verb semantics by capturing diathesis alternations. In particular, for many Croatian verbs reflexivization introduces a meaning shift, e.g., *predati* (*to hand in/out*) vs. *predati se* (*to surrender*). With subject subcategorization, reflexive and irreflexive readings will have different tensor representations; e.g.,  $\langle student, Subj\_tr, zadaća \rangle$  ( $\langle student, Subj\_tr, homework \rangle$ ) vs.  $\langle trupe, Subj\_intr, napadač \rangle$  ( $\langle troops, Subj\_intr, invaders \rangle$ ). Finally, similar to Padó and Utt (2012), we use *Verb* as an underspecified link between subjects and objects linked by non-auxiliary verbs.

For lexicalized links, we use two more extraction patterns for prepositions and verbs. Prepositions are directly lexicalized as links; e.g.,  $\langle mjesto, na, sunce \rangle$  ( $\langle place, on, sun \rangle$ ). The same holds for non-auxiliary verbs linking subjects to objects; e.g.,  $\langle država, kupiti, količina \rangle$  ( $\langle state, buy, amount \rangle$ ).

**Tuple extraction and scoring.** The overall quality of the DM.HR depends on the accuracy of extracted tuples, which is affected by all preprocessing steps. We computed the performance of tuple extraction by evaluating a sample of tuples extracted from a parsed version of SETIMES.HR against the tuples extracted from the SETIMES.HR gold annotations (we use the same sample as for tagging and parsing performance evaluation). Table 2 shows Precision, Recall, and F<sub>1</sub> score. Overall, we achieve the best performance on the *Atr* links, followed by *Pred* links. The performance is generally higher on unlexicalized links than on lexicalized links (note that performance on unlexical-

| Link | Word     | LMI    | Link  | Word        | LMI  |
|------|----------|--------|-------|-------------|------|
| Atv  | moći     | 225107 | Adv   | moguće      | 9669 |
| Atv  | željeti  | 22049  | Atv   | namjeravati | 9095 |
| Obj  | stan     | 19997  | Obj   | karta       | 8936 |
| po   | cijena   | 18534  | prije | godina      | 8584 |
| Pred | kada     | 14408  | Adv   | nedavno     | 7842 |
| Obj  | dionica  | 13720  | Atv   | odlučiti    | 7578 |
| Atv  | morati   | 12097  | Adv   | godina      | 7496 |
| Obj  | ulaznica | 11126  | Obj   | zemljište   | 7180 |

Table 3: Top 16 LMI-scored tuples for the verb *kupiti (to buy)*

ized *Verb* links is identical to overall performance on lexicalized verb links). The overall  $F_1$  score of tuple extraction is 74.6%.

Following DM and DM.DE, we score each extracted tuple using Local Mutual Information (LMI) (Evert, 2005):

$$\text{LMI}(i, j, k) = f(i, j, k) \log \frac{P(i, j, k)}{P(i)P(j)P(k)}$$

For a tuple  $(w_1, l, w_2)$ , LMI scores the association strength between word  $w_1$  and word  $w_2$  via link  $l$  by comparing their joint distribution against the distribution under the independence assumption, multiplied with the observed frequency  $f(w_1, l, w_2)$  to discount infrequent tuples. The probabilities are computed from tuple counts as maximum likelihood estimates. We exclude from the tensor all tuples with a negative LMI score. Finally, we symmetrize the tensor by introducing inverse links.

**Model statistics.** The resulting DM.HR tensor consists of 2.3M lemmas, 121M links and 165K link types (including inverse links). On average, each lemma has 53 links. This makes DM.HR more sparse than English DM (796 link types), but less sparse than German DM (220K link types; 22 links per lemma). Table 3 shows an example of the extracted tuples for the verb *kupiti (to buy)*. DM.HR tensor is freely available for download.<sup>4</sup>

## 5 Evaluating DM.HR

**Task.** We present a pilot evaluation DM.HR on a standard task from distributional semantics, namely synonym choice. In contrast to tasks like predicting word similarity We use the dataset created by Karan et al. (2012), with more than 11,000 synonym choice questions. Each question consists of one target word (nouns, verbs, and adjectives) with

<sup>4</sup><http://takelab.fer.hr/dmhr>

| Model        | Accuracy (%) |             |             | Coverage (%) |      |     |
|--------------|--------------|-------------|-------------|--------------|------|-----|
|              | N            | A           | V           | N            | A    | V   |
| DM.HR        | <b>70.0</b>  | 66.3        | <b>63.2</b> | 99.9         | 99.1 | 100 |
| BOW-LSA      | 67.2         | <b>68.9</b> | 61.0        | 100          | 100  | 100 |
| BOW baseline | 59.9         | 65.7        | 55.9        | 99.9         | 99.7 | 100 |

Table 4: Results on synonym choice task

four synonym candidates (one is correct). The questions were extracted automatically from a machine-readable dictionary of Croatian. An example item is *težak (farmer): poljoprivrednik (farmer), umjetnost (art), radijacija (radiation), bod (point)*. We sampled from the dataset questions for nouns, verbs, and adjectives, with 1000 questions each.<sup>5</sup> Additionally, we manually corrected some errors in the dataset, introduced by the automatic extraction procedure. To make predictions, we compute pairwise cosine similarities of the target word vectors with the four candidates and predict the candidate(s) with maximal similarity (note that there may be ties).

**Evaluation.** Our evaluation follows the scheme developed by Mohammad et al. (2007), who define accuracy as the average number of correct predictions per covered question. Each correct prediction with a single most similar candidate receives a full credit (A), while ties for maximal similarity are discounted (B: two-way tie, C: three-way tie, D: four-way tie):  $A + \frac{1}{2}B + \frac{1}{3}C + \frac{1}{4}D$ . We consider a question item to be covered if the target and at least one answer word are modeled. In our experiments, ties occur when vector similarities are zero for all word pairs (due to vector sparsity). Note that a random baseline would perform at 0.25 accuracy.

As baseline to compare against the DM.HR, we build a standard bag-of-words model from the same corpus. It uses a  $\pm 5$ -word within-sentence context window, and the 10,000 most frequent context words (nouns, adjectives, and verbs) as dimensions. We also compare against BOW-LSA, a state-of-the-art synonym detection model from Karan et al. (2012), which uses 500 latent dimensions and paragraphs as contexts. We determine the significance of differences between the models by computing 95% confidence intervals with bootstrap resampling (Efron and Tibshirani, 1993).

**Results.** Table 4 shows the results for the three considered models on nouns (N), adjectives (A),

<sup>5</sup>Available at: <http://takelab.fer.hr/crosyn>

and verbs (V). The performance of BOW-LSA differs slightly from that reported by Karan et al. (2012), because we evaluate on a sample of their dataset. DM.HR outperforms the baseline BOW model for nouns and verbs (differences are significant at  $p < 0.05$ ). Moreover, on these categories DM.HR performs slightly better than BOW-LSA, but the differences are not statistically significant. Conversely, on adjectives BOW-LSA performs slightly better than DM.HR, but the difference is again not statistically significant. All models achieve comparable and almost perfect coverage on this dataset (BOW-LSA achieves complete coverage because of the way how the original dataset was filtered).

Overall, the biggest improvement over the baseline is achieved for nouns. Nouns occur as heads and dependents of many link types (unlexicalized and lexicalized), and are thus well represented in the semantic space. On the other hand, adjectives seem to be less well modeled. Although the majority of adjectives occur as heads or dependents of the *Atr* relation, for which extraction accuracy is the highest (cf. Table 2), it is likely that a single link type is not sufficient. As noted by a reviewer, more insight could perhaps be gained by comparing the predictions of BOW-LSA and DM.HR models. The generally low performance on verbs suggests that their semantic is not fully covered in word- and syntax-based spaces.

## 6 Conclusion

We have described the construction of DM.HR, a syntax-based distributional memory for Croatian built from a dependency-parsed web corpus. To the best of our knowledge, DM.HR is the first freely available distributional memory for a Slavic language. We have conducted a preliminary evaluation of DM.HR on a synonym choice task, where DM.HR outperformed the bag-of-words model and performed comparable to an LSA model.

This work provides a starting point for a systematic study of dependency-based distributional semantics for Croatian and similar languages. Our first priority will be to analyze how corpus preprocessing and the choice of link types relates to model performance on different semantic tasks. Better modeling of adjectives and verbs is also an important topic for future research.

## Acknowledgments

The first author was supported by the Croatian Science Foundation (project 02.03/162: “Derivational Semantic Models for Information Retrieval”). We thank the reviewers for their constructive comments. Special thanks to Hiko Schamoni, Tae-Gil Noh, and Mladen Karan for their assistance.

## References

- Željko Agić and Danijela Merkle. 2013. Three syntactic formalisms for data-driven dependency parsing of Croatian. *Proceedings of TSD 2013, Lecture Notes in Artificial Intelligence*.
- Željko Agić, Marko Tadić, and Zdravko Dovedan. 2008. Improving part-of-speech tagging accuracy for Croatian by morphological analysis. *Informatika*, 32(4):445–451.
- Željko Agić, Marko Tadić, and Zdravko Dovedan. 2009. Evaluating full lemmatization of Croatian texts. In *Recent Advances in Intelligent Information Systems*, pages 175–184. EXIT Warsaw.
- Željko Agić. 2012. K-best spanning tree dependency parsing with verb valency lexicon reranking. In *Proceedings of COLING 2012: Posters*, pages 1–12, Bombay, India.
- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.
- Bartosz Broda and Maciej Piasecki. 2008. Supermatrix: a general tool for lexical semantic knowledge acquisition. In *Speech and Language Technology*, volume 11, pages 239–254. Polish Phonetics Association.
- Bartosz Broda, Magdalena Derwojedowa, Maciej Piasecki, and Stanisław Szpakowicz. 2008. Corpus-based semantic relatedness for the construction of Polish WordNet. In *Proceedings of LREC*, Marrakech, Morocco.
- Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.
- Tomaž Erjavec. 2012. MULTTEXT-East: Morphosyntactic resources for Central and Eastern European languages. *Language Resources and Evaluation*, 46(1):131–142.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A Flexible, Corpus-driven Model of Regular and Inverse Selectional Preferences. *Computational Linguistics*, 36(4):723–763.

- Stefan Evert. 2005. *The statistics of word cooccurrences*. Ph.D. thesis, PhD Dissertation, Stuttgart University.
- Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos: An open source trigram tagger. In *Proceedings of ACL 2007*, pages 209–212, Prague, Czech Republic.
- Zelig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.
- Anton Karl Ingason, Sigrún Helgadóttir, Hrafn Loftsson, and Eiríkur Rögnvaldsson. 2008. A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI). In *Proceedings of GoTAL*, pages 205–216.
- Vedrana Janković, Jan Šnajder, and Bojana Dalbelo Bašić. 2011. Random indexing distributional semantic models for Croatian language. In *Proceedings of Text, Speech and Dialogue*, pages 411–418, Plzeň, Czech Republic.
- Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Distributional semantics approach to detecting synonyms in Croatian language. In *Proceedings of the Language Technologies Conference, Information Society*, Ljubljana, Slovenia.
- Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool.
- Nikola Ljubešić and Tomaž Erjavec. 2011. hrWac and slWac: Compiling web corpora for Croatian and Slovene. In *Proceedings of Text, Speech and Dialogue*, pages 395–402, Plzeň, Czech Republic.
- Nikola Ljubešić, Damir Boras, Nikola Bakarić, and Jasmina Njavro. 2008. Comparing measures of semantic similarity. In *Proceedings of the ITI 2008 30th International Conference of Information Technology Interfaces*, Cavtat, Croatia.
- Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL-X*, pages 216–220, New York, NY.
- Olga Mitrofanova, Anton Mukhin, Polina Panicheva, and Vyacheslav Savitsky. 2007. Automatic word clustering in Russian texts. In *Proceedings of Text, Speech and Dialogue*, pages 85–91, Plzeň, Czech Republic.
- Saif Mohammad, Iryna Gurevych, Graeme Hirst, and Torsten Zesch. 2007. Cross-lingual distributional profiles of concepts for measuring semantic distance. In *Proceedings of EMNLP/CoNLL*, pages 571–580, Prague, Czech Republic.
- Preslav Nakov. 2001a. Latent semantic analysis for Bulgarian literature. In *Proceedings of Spring Conference of Bulgarian Mathematicians Union*, Borovets, Bulgaria.
- Preslav Nakov. 2001b. Latent semantic analysis for Russian literature investigation. In *Proceedings of the 120 years Bulgarian Naval Academy Conference*.
- Sebastian Padó and Jason Utt. 2012. A distributional memory for German. In *Proceedings of the KONVENS 2012 workshop on lexical-semantic resources and applications*, pages 462–470, Vienna, Austria.
- Maciej Piasecki. 2009. Automated extraction of lexical meanings from corpus: A case study of potentialities and limitations. In *Representing Semantics in Digital Lexicography. Innovative Solutions for Lexical Entry Content in Slavic Lexicography*, pages 32–43. Institute of Slavic Studies, Polish Academy of Sciences.
- Pavel Smrž and Pavel Rychlý. 2001. Finding semantically related words in large corpora. In *Text, Speech and Dialogue*, pages 108–115. Springer.
- Marko Tadić. 2005. The Croatian Lemmatization Server. *Southern Journal of Linguistics*, 29(1):206–217.
- Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

# Generalizing Image Captions for Image-Text Parallel Corpus

Polina Kuznetsova, Vicente Ordonez, Alexander Berg,  
Tamara Berg and Yejin Choi

Department of Computer Science  
Stony Brook University  
Stony Brook, NY 11794-4400

{pkuznetsova, vordonezroma, aberg, tlberg, ychoi}@cs.stonybrook.edu

## Abstract

The ever growing amount of web images and their associated texts offers new opportunities for integrative models bridging natural language processing and computer vision. However, the potential benefits of such data are yet to be fully realized due to the complexity and noise in the alignment between image content and text. We address this challenge with contributions in two folds: first, we introduce the new task of *image caption generalization*, formulated as visually-guided sentence compression, and present an efficient algorithm based on dynamic beam search with dependency-based constraints. Second, we release a new large-scale corpus with 1 million image-caption pairs achieving tighter content alignment between images and text. Evaluation results show the intrinsic quality of the generalized captions and the extrinsic utility of the new image-text parallel corpus with respect to a concrete application of image caption transfer.

## 1 Introduction

The vast number of online images with accompanying text raises hope for drawing synergistic connections between human language technologies and computer vision. However, subtleties and complexity in the relationship between image content and text make exploiting paired visual-textual data an open and interesting problem.

Some recent work has approached the problem of composing natural language descriptions for images by using computer vision to retrieve images with similar content and then transferring

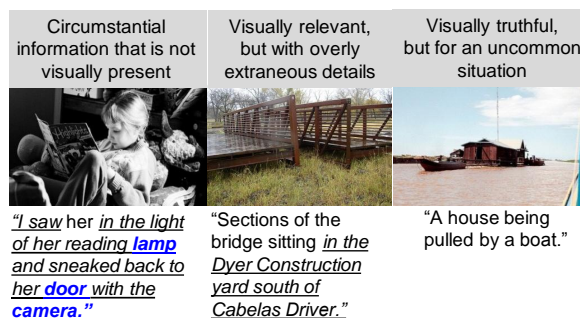


Figure 1: Examples of captions that are not readily applicable to other visually similar images.

text from the retrieved samples to the query image (e.g. Farhadi et al. (2010), Ordonez et al. (2011), Kuznetsova et al. (2012)). Other work (e.g. Feng and Lapata (2010a), Feng and Lapata (2010b)) uses computer vision to bias summarization of text associated with images to produce descriptions. All of these approaches rely on existing text that describes visual content, but many times existing image descriptions contain significant amounts of extraneous, non-visual, or otherwise non-desirable content. The goal of this paper is to develop techniques to automatically clean up visually descriptive text to make it more directly usable for applications exploiting the connection between images and language.

As a concrete example, consider the first image in Figure 1. This caption was written by the photo owner and therefore contains information related to the context of when and where the photo was taken. Objects such as "lamp", "door", "camera" are not visually present in the photo. The second image shows a similar but somewhat different issue. Its caption describes visible objects such as "bridge" and "yard", but "Cabelas Driver" are overly specific and not visually detectable. The

| Dependency Constraints with Examples |                                    |               | Additional Dependency Constraints  |
|--------------------------------------|------------------------------------|---------------|--|
| Constraints                          | Sentence                           | Dependency    |  |
| advcl*(←)                            | Taken when it was running...       | taken←running | acomp*(↔), advmod(←), agent*(←), attr(↔)<br>auxpass(↔), cc*(↔), complm(←), cop*(↔)<br>csubj*/csubjpass*(↔), expl(↔), mark*(↔)<br>infmod*(↔), mwe(↔), nsubj*/nsubjpass*(↔)<br>npadvmod(←), nn(←), conj*(↔), num*(←)<br>number(↔), parataxis(←), ↔<br>partmod*(←), pcomp*(↔), purpcl*(←)<br>possessive(↔), preconj*(←), predet*(←)<br>prt(↔), quantmod(←), rcmmod(←), ref(←)<br>rel*(↔), tmod*(←), xcomp*(→), xsubj(→) |
| amod(←)                              | A wooden chair in the living room  | chair← wooden |  |
| aux(↔)                               | This crazy dog was jumping...      | jumping↔was   |  |
| ccomp*(→)                            | I believe a bear was in the box... | believe→was   |  |
| prep(←)                              | A view from the balcony            | view←from     |  |
| det(↔)                               | A cozy street cafe...              | cafe↔A        |  |
| dobj*(↔)                             | A curious cow surveys the road...  | surveys↔road  |  |
| iobj*(↔)                             | ...rock gives the water the color  | gives↔water   |  |
| neg(↔)                               | Not a cloud in the sky...          | cloud↔Not     |  |
| pobj*(↔)                             | This branch was on the ground...   | on↔ground     |  |

Table 1: Dependency-based Constraints

text of the third image, “*A house being pulled by a boat*”, pertains directly to the visual content of the image, but is unlikely to be useful for tasks such as caption transfer because the depiction is unusual.<sup>1</sup> This phenomenon of information gap between the visual content of the images and their corresponding narratives has been studied closely by Dodge et al. (2012).

The content misalignment between images and text limits the extent to which visual detectors can learn meaningful mappings between images and text. To tackle this challenge, we introduce the new task of *image caption generalization* that rewrites captions to be more visually relevant and more readily applicable to other visually similar images. Our end goal is to convert noisy image-text pairs in the wild (Ordonez et al., 2011) into pairs with tighter content alignment, resulting in new simplified captions over 1 million images. Evaluation results show both the intrinsic quality of the generalized captions and the extrinsic utility of the new image-text parallel corpus. The new parallel corpus will be made publicly available.<sup>2</sup>

## 2 Sentence Generalization as Constraint Optimization

Casting the generalization task as *visually-guided* sentence compression with lightweight revisions, we formulate a constraint optimization problem that aims to maximize content selection and local linguistic fluency while satisfying constraints driven from dependency parse trees. Dependency-based constraints guide the generalized caption

<sup>1</sup>Open domain computer vision remains to be an open problem, and it would be difficult to reliably distinguish pictures of subtle visual differences, e.g., pictures of “*a water front house with a docked boat*” from those of “*a floating house pulled by a boat*”.

<sup>2</sup>Available at <http://www.cs.stonybrook.edu/~ychoi/imgcaption/>

to be grammatically valid (e.g., keeping articles in place, preventing dangling modifiers) while remaining semantically compatible with respect to a given image-text pair (e.g., preserving predicate-argument relations). More formally, we maximize the following objective function:

$$F(y; x) = \Phi(y; x, v) + \Psi(y; x)$$

subject to  $\mathcal{C}(y; x, v)$

where  $x = \{x_i\}$  is the input caption (a sentence),  $v$  is the accompanying image,  $y = \{y_i\}$  is the output sentence,  $\Phi(y; x, v)$  is the content selection score,  $\Psi(y; x)$  is the linguistic fluency score, and  $\mathcal{C}(y; x, v)$  is the set of hard constraints. Let  $l(y_i)$  be the index of the word in  $x$  that is selected as the  $i$ 'th word in the output  $y$  so that  $x_{l(y_i)} = y_i$ . Then, we factorize  $\Phi(\cdot)$  and  $\Psi(\cdot)$  as:

$$\begin{aligned} \Phi(y; x, v) &= \sum_i \phi(y_i, x, v) = \sum_i \phi(x_{l(y_i)}, v) \\ \Psi(y; x) &= \sum_i \psi(y_i, \dots, y_{i-K}) \\ &= \sum_i \psi(x_{l(y_i)}, \dots, x_{l(y_{i-K})}) \end{aligned}$$

where  $K$  is the size of local context.

### Content Selection – Visual Estimates:

The computer vision system used consists of 7404 visual classifiers for recognizing leaf level WordNet synsets (Fellbaum, 1998). Each classifier is trained using labeled images from the ImageNet dataset (Deng et al., 2009) – an image database of over 14 million hand labeled images organized according to the WordNet hierarchy. Image similarity is represented using a Spatial Pyramid Match Kernel (SPM) (Lazebnik et al., 2006) with Locality-constrained Linear Coding (Wang et al., 2010) on shape based SIFT features (Lowe, 2004).

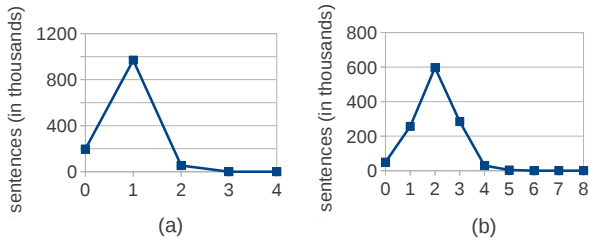


Figure 2: Number of sentences (y-axis) for each *average* (x-axis in (a)) and *maximum* (x-axis in (b)) number of words with future dependencies

Models are linear SVMs followed by a sigmoid to produce probability for each node.<sup>3</sup>

### Content Selection – Salient Topics:

We consider Tf.Idf driven scores to favor salient topics, as those are more likely to generalize across many different images. Additionally, we assign a very low content selection score ( $-\infty$ ) for proper nouns and numbers and a very high score (larger than maximum idf or visual score) for the 2k most frequent words in our corpus.

### Local Linguistic Fluency:

We model linguistic fluency with 3-gram conditional probabilities:

$$\begin{aligned} & \psi(x_{l(y_i)}, x_{l(y_{i-1})}, x_{l(y_{i-2})}) \\ & = p(x_{l(y_i)} | x_{l(y_{i-2})}, x_{l(y_{i-1})}) \end{aligned} \quad (1)$$

We experiment with two different ngram statistics, one extracted from the Google Web 1T corpus (Brants and Franz., 2006), and the other computed from the 1M image-caption corpus (Ordonez et al., 2011).

### Dependency-driven Constraints:

Table 1 defines the list of dependencies used as constraints driven from the typed dependencies (de Marneffe and Manning, 2009; de Marneffe et al., 2006). The direction of arrows indicate the direction of inclusion requirements. For example,  $dep(X \leftarrow Y)$ , denotes that “X” must be included whenever “Y” is included. Similarly,  $dep(X \longleftrightarrow Y)$  denotes that “X” and “Y” must either be included together or eliminated together. We determine the uni- or bi-directionality of these constraints by manually examining a few example sentences corresponding to each of these typed dependencies. Note that some dependencies such as  $det(\longleftrightarrow)$  would hold regardless of the particular

<sup>3</sup>Code was provided by Deng et al. (2012).

| Method-1 (M1) | v.s. | Method-2 (M2)     | M1 wins over M2 |
|---------------|------|-------------------|-----------------|
| SALIENCY      |      | ORIG              | 76.34%          |
| VISUAL        |      | ORIG              | 81.75%          |
| VISUAL        |      | SALIENCY          | 72.48%          |
| VISUAL        |      | VISUAL W/O CONSTR | 83.76%          |
| VISUAL        |      | NGRAM-ONLY        | 90.20%          |
| VISUAL        |      | HUMAN             | 19.00%          |

Table 2: Forced Choice Evaluation (LM Corpus = Google)

lexical items, while others, e.g.,  $do_{bj}(\longleftrightarrow)$  may or may not be necessary depending on the context. Those dependencies that we determine as largely context dependent are marked with \* in Table 1.

One could consider enforcing all dependency constraints in Table 1 as hard constraints so that the compressed sentence must not violate any of those directed dependency constraints. Doing so would lead to overly conservative compression with least compression ratio however. Therefore, we relax those that are largely context dependent as soft constraints (marked in Table 1 with \*) by introducing a constant penalty term in the objective function. Alternatively, the dependency based constraints can be learned statistically from the training corpus of paired original and compressed sentences. Since we do not have such in-domain training data at this time, we leave this exploration as future research.

### Dynamic Programming with Dynamic Beam:

The constraint optimization we formulated corresponds to an NP-hard problem. In our work, hard constraints are based only on typed dependencies, and we find that long range dependencies occur infrequently in actual image descriptions, as plotted in Figure 2. With this insight, we opt for decoding based on dynamic programming with dynamically adjusted beam.<sup>4</sup> Alternatively, one can find an approximate solution using Integer Linear Programming (e.g., Clarke and Lapata (2006), Clarke and Lapata (2007), Martins and Smith (2009)).

## 3 Evaluation

Since there is no existing benchmark data for image caption generalization, we crowdsource evaluation using Amazon Mechanical Turk (AMT). We empirically compare the following options:

<sup>4</sup>The required beam size at each step depends on how many words have dependency constraints involving any word following the current one – beam size is at most  $2^p$ , where  $p$  is the max number of words dependent on any future words.





Figure 3: Example Image Caption Transfer

| Method   | LM Corpus     | strict matching |              |              |              | semantic matching |              |              |              |
|----------|---------------|-----------------|--------------|--------------|--------------|-------------------|--------------|--------------|--------------|
|          |               | BLEU            | P            | R            | F            | BLEU              | P            | R            | F            |
| ORIG     | N/A           | 0.063           | 0.064        | <b>0.139</b> | <b>0.080</b> | 0.215             | 0.220        | <b>0.508</b> | 0.276        |
| SALIENCY | Image Corpus  | 0.060           | 0.074        | 0.077        | 0.068        | 0.302             | 0.411        | 0.399        | 0.356        |
| VISUAL   | Image Corpus  | 0.060           | <b>0.075</b> | 0.075        | 0.068        | <b>0.305</b>      | <b>0.422</b> | 0.397        | <b>0.360</b> |
| SALIENCY | Google Corpus | 0.064           | 0.070        | 0.101        | 0.074        | 0.286             | 0.337        | 0.459        | 0.340        |
| VISUAL   | Google Corpus | <b>0.065</b>    | 0.071        | 0.098        | 0.075        | 0.296             | 0.354        | 0.457        | 0.350        |

Table 3: Image Description Transfer: performance in BLEU and F1 with *strict* & *semantic* matching.

- ORIG: original uncompressed captions
- HUMAN: compressed by humans (See § 3.2)
- SALIENCY: linguistic fluency + saliency-based content selection + dependency constraints
- VISUAL: linguistic fluency + visually-guided content selection + dependency constraints
- $x$  W/O CONSTR: method  $x$  without dependency constraints
- NGRAM-ONLY: linguistic fluency only

### 3.1 Intrinsic Evaluation: Forced Choice

Turkers are provided with an image and two captions (produced by different methods) and are asked to select a better one, i.e., the most relevant and plausible caption that contains the least extraneous information. Results are shown in Table 2. We observe that VISUAL (full model with visually guided content selection) performs the best, being selected over SALIENCY (content-selection without visual information) in 72.48% cases, and *even over the original image caption in 81.75% cases.*

This forced-selection experiment between VISUAL and ORIG demonstrates the degree of noise prevalent in the image captions in the wild. Of course, if compared against human-compressed captions, the automatic captions are preferred much less frequently – in 19% of the cases. In those 19% cases when automatic captions are preferred over human-compressed ones, it is sometimes that humans did not fully remove information that is not visually present or verifiable, and other times humans overly compressed. To ver-

ify the utility of dependency-based constraints, we also compare two variations of VISUAL, with and without dependency-based constraints. As expected, the algorithm with constraints is preferred in the majority of cases.

### 3.2 Extrinsic Evaluation: Image-based Caption Retrieval

We evaluate the usefulness of our new image-text parallel corpus for automatic generation of image descriptions. Here the task is to produce, for a query image, a relevant description, i.e., a visually descriptive caption. Following Ordonez et al. (2011), we produce a caption for a query image by finding top  $k$  most similar images within the 1M image-text corpus (Ordonez et al., 2011) and then transferring their captions to the query image. To compute evaluation measures, we take the average scores of BLEU(1) and F-score (unigram-based with respect to content-words) over  $k = 5$  candidate captions.

Image similarity is computed using two global (whole) image descriptors. The first is the GIST feature (Oliva and Torralba, 2001), an image descriptor related to perceptual characteristics of scenes – naturalness, roughness, openness, etc. The second descriptor is also a global image descriptor, computed by resizing the image into a “tiny image” (Torralba et al., 2008), which is effective in matching the structure and overall color of images. To find visually relevant images, we compute the similarity of the query image to im-



Figure 4: Good (left three, in blue) and bad examples (right three, in red) of generalized captions

ages in the whole dataset using an unweighted sum of gist similarity and tiny image similarity.

Gold standard (human compressed) captions are obtained using AMT for 1K images. The results are shown in Table 3. *Strict matching* gives credit only to identical words between the gold-standard caption and the automatically produced caption. However, words in the original caption of the query image (and its compressed caption) do not overlap exactly with words in the retrieved captions, even when they are semantically very close, which makes it hard to see improvements even when the captions of the new corpus are more general and transferable over other images. Therefore, we also report scores based on *semantic matching*, which gives partial credits to word pairs based on their lexical similarity.<sup>5</sup> The best performing approach with semantic matching is VISUAL (with LM = Image corpus), improving BLEU, Precision, F-score substantially over those of ORIG, demonstrating the extrinsic utility of our newly generated image-text parallel corpus in comparison to the original database. Figure 3 shows an example of caption transfer.

#### 4 Related Work

Several recent studies presented approaches to automatic caption generation for images (e.g., Farhadi et al. (2010), Feng and Lapata (2010a), Feng and Lapata (2010b), Yang et al. (2011), Kulkarni et al. (2011), Li et al. (2011), Kuznetsova et al. (2012)). The end goal of our work differs in that we aim to revise original image captions into

<sup>5</sup>We take Wu-Palmer Similarity as similarity measure (Wu and Palmer, 1994). When computing BLEU with semantic matching, we look for the match with the highest similarity score among words that have not been matched before. Any word matched once (even with a partial credit) will be removed from consideration when matching next words.

descriptions that are more general and align more closely to the visual image content.

In comparison to prior work on sentence compression, our approach falls somewhere between unsupervised to distant-supervised approach (e.g., Turner and Charniak (2005), Filippova and Strube (2008)) in that there is not an in-domain training corpus to learn generalization patterns directly. Future work includes exploring more direct supervision from human edited sample generalization (e.g., Knight and Marcu (2000), McDonald (2006)) Galley and McKeown (2007), Zhu et al. (2010)), and the inclusion of edits beyond deletion, e.g., substitutions, as has been explored by e.g., Cohn and Lapata (2008), Cordeiro et al. (2009), Napoles et al. (2011).

#### 5 Conclusion

We have introduced the task of image caption generalization as a means to reduce noise in the parallel corpus of images and text. Intrinsic and extrinsic evaluations confirm that the captions in the resulting corpus align better with the image contents (are often preferred over the original captions by people), and can be practically more useful with respect to a concrete application.

#### Acknowledgments

This research was supported in part by the Stony Brook University Office of the Vice President for Research. Additionally, Tamara Berg is supported by NSF #1054133 and NSF #1161876. We thank reviewers for many insightful comments and suggestions.

## References

- Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. In *Linguistic Data Consortium*.
- James Clarke and Mirella Lapata. 2006. Constraint-based sentence compression: An integer programming approach. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 144–151, Sydney, Australia, July. Association for Computational Linguistics.
- James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1–11, Prague, Czech Republic, June. Association for Computational Linguistics.
- Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK, August. Coling 2008 Organizing Committee.
- Joao Cordeiro, Gael Dias, and Pavel Brazdil. 2009. Unsupervised induction of sentence compression rules. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCLG+Sum 2009)*, pages 15–22, Suntec, Singapore, August. Association for Computational Linguistics.
- Marie-Catherine de Marneffe and Christopher D. Manning. 2009. Stanford typed dependencies manual.
- Marie-Catherine de Marneffe, Bill Maccartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Language Resources and Evaluation Conference 2006*.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition*.
- Jia Deng, Jonathan Krause, Alexander C. Berg, and L. Fei-Fei. 2012. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Conference on Computer Vision and Pattern Recognition*.
- Jesse Dodge, Amit Goyal, Xufeng Han, Alyssa Mensch, Margaret Mitchell, Karl Stratos, Kota Yamaguchi, Yejin Choi, Hal Daume III, Alex Berg, and Tamara Berg. 2012. Detecting visual text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 762–772, Montréal, Canada, June. Association for Computational Linguistics.
- Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young1, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: generating sentences for images. In *European Conference on Computer Vision*.
- Christiane D. Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.
- Yansong Feng and Mirella Lapata. 2010a. How many words is a picture worth? automatic caption generation for news images. In *Association for Computational Linguistics*.
- Yansong Feng and Mirella Lapata. 2010b. Topic models for image annotation and text illustration. In *Human Language Technologies*.
- Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference, INLG '08*, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *AAAI/IAAI*, pages 703–710.
- Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2011. Babytalk: Understanding and generating simple image descriptions. In *Conference on Computer Vision and Pattern Recognition*.
- Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara Berg, and Yejin Choi. 2012. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 359–368, Jeju Island, Korea, July. Association for Computational Linguistics.
- Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching. In *Conference on Computer Vision and Pattern Recognition*, June.
- Siming Li, Girish Kulkarni, Tamara L. Berg, Alexander C. Berg, and Yejin Choi. 2011. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, Portland, Oregon, USA, June. Association for Computational Linguistics.
- David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November.

- Andre Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 1–9, Boulder, Colorado, June. Association for Computational Linguistics.
- Ryan T. McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.
- Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011. Paraphrastic sentence compression with a character-based metric: Tightening without deletion. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 84–90, Portland, Oregon, June. Association for Computational Linguistics.
- Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*.
- Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *Neural Information Processing Systems (NIPS)*.
- Antonio Torralba, Rob Fergus, and William T. Freeman. 2008. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *Pattern Analysis and Machine Intelligence*, 30.
- Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 290–297, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong. 2010. Locality-constrained linear coding for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- 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, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yezhou Yang, Ching Teo, Hal Daume III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 444–454, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Zheming Zhu, Delphine Bernhard, and Iryna 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, Beijing, China, August. Coling 2010 Organizing Committee.

# Recognizing Identical Events with Graph Kernels

Goran Glavaš and Jan Šnajder

University of Zagreb

Faculty of Electrical Engineering and Computing

Text Analysis and Knowledge Engineering Lab

Unska 3, 10000 Zagreb, Croatia

{goran.glavas, jan.snajder}@fer.hr

## Abstract

Identifying news stories that discuss the same real-world events is important for news tracking and retrieval. Most existing approaches rely on the traditional vector space model. We propose an approach for recognizing identical real-world events based on a structured, event-oriented document representation. We structure documents as graphs of event mentions and use graph kernels to measure the similarity between document pairs. Our experiments indicate that the proposed graph-based approach can outperform the traditional vector space model, and is especially suitable for distinguishing between topically similar, yet non-identical events.

## 1 Introduction

News stories typically describe real-world events. Topic detection and tracking (TDT) aims to detect stories that discuss identical or directly related events, and track these stories as they evolve over time (Allan, 2002). Being able to identify the stories that describe the same real-world event is essential for TDT, and event-based information retrieval in general.

In TDT, an event is defined as something happening in a certain place at a certain time (Yang et al., 1999), while a topic is defined as a set of news stories related by some seminal real-world event (Allan, 2002). To identify news stories on the same topic, most TDT approaches rely on traditional vector space models (Salton et al., 1975), as more sophisticated natural language processing techniques have not yet proven to be useful for this task. On the other hand, significant advances in sentence-level event extraction have been made over the last decade, in particular as the result of

standardization efforts such as TimeML (Pustejovsky et al., 2003a) and TimeBank (Pustejovsky et al., 2003b), as well as dedicated evaluation tasks (ACE, 2005; Verhagen et al., 2007; Verhagen et al., 2010). However, these two lines of research have largely remained isolated from one another.

In this paper we bridge this gap and address the task of recognizing stories discussing identical events by considering structured representations from sentence-level events. More concretely, we structure news stories into *event graphs* built from individual event mentions extracted from text. To measure event-based similarity of news stories, we compare their event graphs using graph kernels (Borgwardt, 2007). We conduct preliminary experiments on two event-oriented tasks and show that the proposed approach can outperform traditional vector space model in recognizing identical real-world events. Moreover, we demonstrate that our approach is especially suitable for distinguishing between topically similar, yet non-identical real-world events.

## 2 Related Work

The traditional vector space model (VSM) (Salton et al., 1975) computes the cosine between bag-of-words representations of documents. The VSM is at the core of most approaches that identify same-topic news stories (Hatzivassiloglou et al., 2000; Brants et al., 2003; Kumaran and Allan, 2005; Atkinson and Van der Goot, 2009). However, it has been observed that some word classes (e.g., named entities, noun phrases, collocations) have more significance than the others. Among them, named entities have been considered as particularly important, as they often identify the participants of an event. In view of this, Hatzivassiloglou et al. (2000) restrict the set of words to be used for document representation to words constituting noun phrases and named entities. Makkonen et

al. (2004) divide document terms into four semantic categories (locations, temporal expressions, proper names, and general terms) and construct separate vector for each of them. Kumaran and Allan (2004) represent news stories with three different vectors, modeling all words, named-entity words, and all non-named-entity words occurring in documents. When available, recognition of identical events can rely on meta-information associated with news stories, such as document creation time (DCT). Atkinson and Van der Goot (2009) combine DCT with VSM, assuming that temporally distant news stories are unlikely to describe the same event.

In research on event extraction, the task of recognizing identical events is known as *event coreference resolution* (Bejan and Harabagiu, 2010; Lee et al., 2012). There, however, the aim is to identify sentence-level event mentions referring to the same real-world events, and not stories that discuss identical events.

### 3 Kernels on Event Graphs

To identify the news describing the same real-world event, we (1) structure event-oriented information from text into event graphs and (2) use graph kernels to measure the similarity between a pair of event graphs.

#### 3.1 Event graphs

An event graph is a vertex- and edge-labeled mixed graph in which vertices represent individual event mentions and edges represent temporal relations between event mentions. We adopt a generic representation of event mentions, as proposed by Glavaš and Šnajder (2013): each mention consists of an *anchor* (a word that conveys the core meaning) and four types of *arguments* (agent, target, time, location). Furthermore, we consider four types of temporal relations between event mentions: *before*, *after*, *overlap*, and *equal* (Allen, 1983). As relations *overlap* and *equal* are symmetric, whereas *before* and *after* are not, an event graph may contain both directed and undirected edges.

Formally, an event graph  $G$  is represented as a tuple  $G = (V, E, A, m, r)$ , where  $V$  is the set of vertices,  $E$  is the set of undirected edges,  $A$  is the set of directed edges (arcs),  $m : V \rightarrow M$  is a bijection mapping the vertices to event mentions, and  $r : E \rightarrow R$  is the edge-labeling function, as-

signing temporal relations to edges (cf. Fig. 1).

The construction of an event graph from a news story involves the extraction of event mentions (anchors and arguments) and the extraction of temporal relations between mentions. We use a supervised model (with 80% F1 extraction performance) based on a rich set of features similar to those proposed by Bethard (2008) to extract event anchors. We then employ a robust, rule-based approach proposed by Glavaš and Šnajder (2013) to extract generic event arguments. Finally, we employ a supervised model (60% micro-averaged F1 classification performance) with a rich set of features, similar to those proposed by Bethard (2008), to extract temporal relations between event mentions. A detailed description of the graph construction steps is outside the scope of this paper.

To compute event graph kernels (cf. Section 3.2), we need to determine whether two event mentions co-refer. To resolve cross-document event coreference, we use the model proposed by Glavaš and Šnajder (2013). The model determines coreference by comparing factual event anchors and arguments of four coarse-grained semantic types (*agent*, *target*, *location*, and *time*), and achieves an F-score of 67% (79% precision and 57% recall) on the cross-document mention pairs from the EventCorefBank dataset (Bejan and Harabagiu, 2008). In what follows,  $cf(m_1, m_2)$  denotes whether event mentions  $m_1$  and  $m_2$  co-refer (equals 1 if mentions co-refer, 0 otherwise).

#### 3.2 Graph kernels

Graph kernels are fast polynomial alternatives to traditional graph comparison techniques (e.g., subgraph isomorphism), which provide an expressive measure of similarity between graphs (Borgwardt, 2007). We employ two different graph kernels: *product graph kernel* and *weighted decomposition kernel*. We chose these kernels because their general forms have intuitive interpretations for event matching. These particular kernels have shown to perform well on a number of tasks from cheminformatics (Mahé et al., 2005; Menchetti et al., 2005).

**Product graph kernel.** A product graph kernel (PGK) counts the common walks between two input graphs (Gärtner et al., 2003). The graph product of two labeled graphs,  $G$  and  $G'$ , denoted  $G_P = G \times G'$ , is a graph with the vertex set

$$V_P = \{(v, v') \mid v \in V_G, v' \in V_{G'}, \delta(v, v')\}$$

where  $\delta(v, v')$  is a predicate that holds when vertices  $v$  and  $v'$  are identically labeled (Ham-mack et al., 2011). Given event graphs  $G = (V, E, A, m, r)$  and  $G' = (V', E', A', m', r')$ , we consider the vertices to be identically labeled if the corresponding event mentions co-refer, i.e.,  $\delta(v, v') \doteq cf(m(v), m'(v'))$ . The edge set of the graph product depends on the type of the product. We experiment with two different products: *tensor product* and *conormal product*. In the tensor product, an edge is introduced iff the corresponding edges exist in both input graphs and the labels of those edges match (i.e., both edges represent the same temporal relation). In the conormal product, an edge is introduced iff the corresponding edge exists in at least one input graph. Thus, a conormal product may compensate for omitted temporal relations in the input graphs.

Let  $A_P$  be the adjacency matrix of the graph product  $G_P$  built from input graphs  $G$  and  $G'$ . The product graph kernel that counts common walks in  $G$  and  $G'$  can be computed efficiently as:

$$K_{PG}(G, G') = \sum_{i,j=1}^{|V_P|} [(I - \lambda A_P)^{-1}]_{ij} \quad (1)$$

when  $\lambda < 1/t$ , where  $t$  is the maximum degree of a vertex in the graph product  $G_P$ . In our experiments, we set  $\lambda$  to  $1/(t + 1)$ .

**Weighted decomposition kernel.** A weighted decomposition kernel (WDK) compares small graph parts, called *selectors*, being matched according to an equality predicate. The importance of the match is weighted by the similarity of the contexts in which the matched selectors occur. For a description of a general form of WDK, see Menchetti et al. (2005).

Let  $S(G)$  be the set of all pairs  $(s, z)$ , where  $s$  is the selector (subgraph of interest) and  $z$  is the context of  $s$ . We decompose event graphs into individual vertices, i.e., we define selectors to be the individual vertices. In this case, similarly as above, the equality predicate  $\delta(v, v')$  for two vertices  $v \in G$  and  $v' \in G'$  holds if and only if the corresponding event mentions  $m(v)$  and  $m'(v')$  co-refer. Using selectors that consist of more than one vertex would require a more complex and perhaps a less intuitive definition of the equality predicate  $\delta$ . The selector context  $Z_v$  of vertex  $v$  is a subgraph of  $G$  that contains  $v$  and all its immediate neighbors. In other words, we consider as context all event men-

tions that are in a direct temporal relation with the selected mention. WDK between event graphs  $G$  and  $G'$  is computed as:

$$K_{WD}(G, G') = \sum_{v \in V_G, v' \in V_{G'}} cf(m(v), m'(v')) \kappa(Z_v, Z_{v'}) \quad (2)$$

where  $\kappa(Z_v, Z_{v'})$  is the *context kernel* measuring the similarity between the context  $Z_v$  of selector  $v \in G$  and the context  $Z_{v'}$  of selector  $v' \in G'$ . We compute the context kernel  $\kappa$  as the number of coreferent mention pairs found between the contexts, normalized by the context size:

$$\kappa(Z_v, Z_{v'}) = \frac{\sum_{w \in V_{Z_v}, w' \in V_{Z_{v'}}} cf(m(w), m'(w'))}{\max(|V_{Z_v}|, |V_{Z_{v'}}|)}$$

The intuition behind this is that a pair of coreferent mentions  $m(v)$  and  $m'(v')$  should contribute to the overall event similarity according to the number of pairs of coreferent mentions,  $m(w)$  and  $m'(w')$ , that are in temporal relation with  $v$  and  $v'$ , respectively.

**Graph kernels example.** As an example, consider the following two story snippets describing the same sets of real-world events:

**Story 1:** *A Cezanne masterpiece worth at least \$131 million that was the yanked from the wall of a Zurich art gallery in 2008 has been recovered, Serbian police said today. Four arrests were made overnight in connection with the theft, which was one of the biggest art heists in recent history.*

**Story 2:** *Serbian police have recovered a painting by French impressionist Paul Cezanne worth an estimated 100 million euros (131.7 million U.S. dollars), media reported on Thursday. The painting "A boy in a red vest" was stolen in 2008 from a Zurich museum by masked perpetrators. Four members of an international crime ring were arrested Wednesday.*

The corresponding event graphs  $G$  and  $G'$  are shown in Fig. 1a and 1b, respectively, while their product is shown in Fig. 1c. There are three pairs of coreferent event mentions between  $G$  and  $G'$ : (*yanked, stolen*), (*recovered, recovered*), and (*arrests, arrested*). Accordingly, the product graph  $P$  has three nodes. The dashed edge between vertices (*yanked, stolen*) and (*arrests, arrested*) exists only in the conormal product graph. By substituting into (1) the adjacency matrix and maximum vertex degree of tensor product graph  $P$ , we obtain

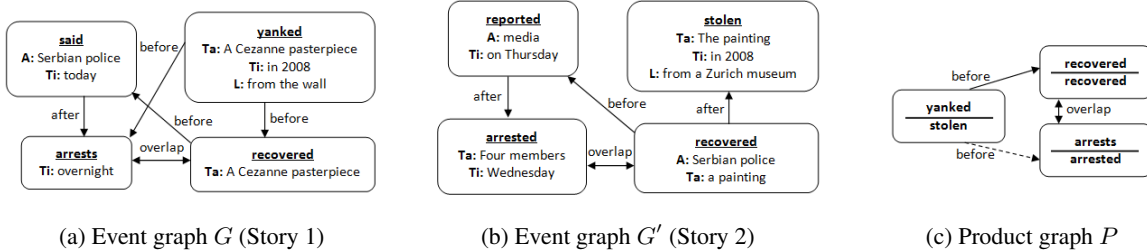


Figure 1: Example event graphs and their product

the tensor PGK score as:

$$K_{PG} = \sum_{i,j=1}^3 \left[ \left( I - \frac{1}{3} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \right)^{-1} \right]_{i,j} \approx 5.6$$

Similarly, for the conormal product graph  $P$  we obtain the conormal PGK score of  $K_{PG} = 9$ . By substituting  $G$  and  $G'$  into (2), we obtain the WDK score as:

$$K_{WD} = \sum_{(v,v') \in V_P} \kappa(Z_v, Z_{v'}) = \frac{2}{3} + \frac{3}{4} + \frac{2}{4} \approx 1.9$$

where  $V_P$  contains pairs of coreferent event mentions:  $(yanked, stolen)$ ,  $(recovered, recovered)$ , and  $(arrests, arrested)$ .

## 4 Experiments

We conducted two preliminary experiments to investigate whether kernels on event graphs can be used to recognize identical events.

### 4.1 Task 1: Recognizing identical events

**Dataset.** In the first experiment, we classify pairs of news stories as either describing identical real-world events or not. For this we need a collection of stories in which pairs of stories on identical events have been annotated as such. TDT corpora (Wayne, 2000) is not directly usable because it has no such annotations. We therefore decided to build a small annotated dataset.<sup>1</sup> To this end, we use the news clusters of the EMM NewsBrief service (Steinberger et al., 2009). EMM clusters news stories from different sources using a document similarity score. We acquired 10 randomly chosen news clusters, manually inspected each of them, and retained in each cluster only the documents that describe the same real-world events. Additionally, we ensured that no documents from

<sup>1</sup>Datasets for both experiments are available at: <http://takelab.fer.hr/evkernels>

| Model           | P           | R           | F           |
|-----------------|-------------|-------------|-------------|
| Tensor PGK      | 89.7        | 82.3        | 85.8        |
| Conormal PGK    | 89.3        | 77.8        | 83.2        |
| WDK             | 88.6        | 73.7        | 80.5        |
| SVM Graph       | 91.1        | 87.6        | 89.3        |
| SVM Graph + VSM | <b>93.8</b> | <b>96.2</b> | <b>95.0</b> |
| VSM baseline    | 90.9        | 82.9        | 86.7        |

Table 1: Results for recognition of identical events

different clusters discuss the same event. To obtain the gold standard dataset, we build all pairs of documents. The final dataset consists of 64 documents in 10 clusters, with 195 news pairs from the same clusters (positive pairs) and 1821 news pairs from different clusters (negative pairs). We divide the dataset into a train and a test set (7:3 split ratio). Note that, although our dataset has ground-truth annotations, it is incomplete in the sense that some pairs of documents describing the same events, which were not recognized as such by the EMM, are not included. Furthermore, because EMM similarity score uses VSM cosine similarity as one of the features, VSM cosine similarity constitutes a competitive baseline on this dataset.

**Results.** For each graph kernel and the VSM baseline, we determine the optimal threshold on the train set and evaluate the classification performance on the test set. The results are given in Table 1. The precision is consistently higher than recall for all kernels and the baseline. High precision is expected, as clusters represent topically dissimilar events. PGK models (both tensor and conormal) outperform the WDK model, indicating that common walks correlate better to event-based document similarity than common subgraphs. Individually, none of the graph kernels outperforms the baseline. To investigate whether the two kernels complement each other, we fed the



|   |
|---|
| <b>Original</b><br>"Taliban militants have attacked a prison in north-west Pakistan, freeing at least 380 prisoners. . ."                         |
| <b>Event-preserving paraphrase</b><br>"Taliban militants in northwest Pakistan attacked the prison, liberated at least 380 prisoners. . ."        |
| <b>Event-shifting paraphrase</b><br>"Taliban militants have been arrested in north-west Pakistan. At least 380 militants have been arrested. . ." |

Table 2: Event paraphrasing example

individual kernel scores to an SVM model (with RBF kernel), along with additional graph-based features such as the number of nodes and the number of edges (*SVM graph* model). Finally, we combined the graph-based features with the VSM cosine similarity (*SVM graph + VSM* model). *SVM graph* model significantly (at  $p < 0.05$ , student’s 2-tailed t-test) outperforms the individual kernel models and the baseline. The combined model (*SVM graph + VSM*) significantly (at  $p < 0.01$ ) outperforms the baseline and all kernel models.

## 4.2 Task 2: Event-based similarity ranking

**Dataset.** In the second experiment we focus on the task of distinguishing between news stories that describe topically very similar, yet distinct events. For this purpose, we use a small set of event paraphrases, constructed as follows. We manually selected 10 news stories from EMM NewsBrief and altered each of them to obtain two meaning-preserving (event-preserving) and two meaning-changing (event-shifting) paraphrases. To obtain the meaning-preserving paraphrases, we use Google translate and round-trip translation via two pairs of arbitrarily chosen languages (Danish/Finnish and Croatian/Hungarian). Annotators manually corrected lexical and syntactic errors introduced by the round-trip translation. To obtain meaning-changing paraphrases, we asked human annotators to alter each story so that it topically resembles the original, but describes a different real-world event. The extent of the alteration was left to the annotators, i.e., no specific transformations were proposed. Paraphrase examples are given in Table 2. The final dataset consists of 60 news pairs: 30 positive and 30 negative.

**Results.** For each method we ranked the pairs based on the assigned similarity scores. An ideal method would rank all positive pairs above all negative pairs. We evaluated the performance using

| Model        | R-prec.     | Avg. prec.  |
|--------------|-------------|-------------|
| Tensor PGK   | 86.7        | 96.8        |
| Conormal PGK | <b>93.3</b> | <b>97.5</b> |
| WDK          | 86.7        | 95.7        |
| VSM baseline | 80.0        | 77.1        |

Table 3: Results for event-based similarity ranking

two different rank evaluation metrics: R-precision (precision at rank 30, as there are 30 positive pairs) and average precision. The performance of graph kernel models and the VSM baseline is given in Table 3. We tested the significance of differences using stratified shuffling (Yeh, 2000). When considering average precision, all kernel models significantly (at  $p < 0.01$ ) outperform the baseline. However, when considering R-precision, only the conormal PGK model significantly (at  $p < 0.05$ ) outperforms the baseline. There is no statistical significance in performance differences between the considered kernel methods. Inspection of the rankings reveals that graph kernels assign very low scores to negative pairs, i.e., they distinguish well between textual representations of topically similar, but different real-world events.

## 5 Conclusion

We proposed a novel approach for recognizing identical events that relies on structured, graph-based representations of events described in a document. We use graph kernels as an expressive framework for modeling the similarity between structured events. Preliminary results on two event-similarity tasks are encouraging, indicating that our approach can outperform traditional vector-space model, and is suitable for distinguishing between topically very similar events. Further improvements could be obtained by increasing the accuracy of event coreference resolution, which has a direct influence on graph kernels.

The research opens up many interesting directions for further research. Besides a systematic evaluation on larger datasets, we intend to investigate the applications in event tracking and event-oriented information retrieval.

## Acknowledgments

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986. We thank the reviewers for their constructive comments.

## References

- ACE. 2005. Evaluation of the detection and recognition of ACE: Entities, values, temporal expressions, relations, and events.
- James Allan. 2002. *Topic Detection and Tracking: Event-based Information Organization*, volume 12. Kluwer Academic Pub.
- James Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Martin Atkinson and Erik Van der Goot. 2009. Near real time information mining in multilingual news. In *Proceedings of the 18th International Conference on World Wide Web*, pages 1153–1154. ACM.
- Cosmin Adrian Bejan and Sanda Harabagiu. 2008. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.
- Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422. Association for Computational Linguistics.
- Steven Bethard. 2008. *Finding Event, Temporal and Causal Structure in Text: A Machine Learning Approach*. Ph.D. thesis, University of Colorado at Boulder.
- Karsten Michael Borgwardt. 2007. *Graph Kernels*. Ph.D. thesis, Ludwig-Maximilians-Universität München.
- Thorsten Brants, Francine Chen, and Ayman Farahat. 2003. A system for new event detection. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 330–337. ACM.
- Thomas Gärtner, Peter Flach, and Stefan Wrobel. 2003. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer.
- Goran Glavaš and Jan Šnajder. 2013. Exploring coreference uncertainty of generically extracted event mentions. In *Proceedings of 14th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 408–422. Springer.
- Richard Hammack, Wilfried Imrich, and Sandi Klavžar. 2011. *Handbook of Product Graphs*. Discrete Mathematics and Its Applications. CRC Press.
- Vasileios Hatzivassiloglou, Luis Gravano, and Anki-needu Maganti. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 224–231. ACM.
- Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 297–304. ACM.
- Giridhar Kumaran and James Allan. 2005. Using names and topics for new event detection. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 121–128. Association for Computational Linguistics.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500. Association for Computational Linguistics.
- Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. 2005. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *Journal of Chemical Information and Modeling*, 45(4):939–951.
- Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. 2004. Simple semantics in topic detection and tracking. *Information Retrieval*, 7(3):347–368.
- Sauro Menchetti, Fabrizio Costa, and Paolo Frasconi. 2005. Weighted decomposition kernels. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 585–592. ACM.
- James Pustejovsky, José Castano, Robert Ingria, Roser Sauri, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 3:28–34.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003b. The TimeBank corpus. In *Corpus Linguistics*, volume 2003, page 40.
- Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

- Ralf Steinberger, Bruno Pouliquen, and Erik Van Der Goot. 2009. An introduction to the european media monitor family of applications. In *Proceedings of the Information Access in a Multilingual World-Proceedings of the SIGIR 2009 Workshop*, pages 1–8.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics.
- Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. Association for Computational Linguistics.
- Charles Wayne. 2000. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Proceedings of the Second International Conference on Language Resources and Evaluation Conference (LREC 2000)*, volume 2000, pages 1487–1494.
- Yiming Yang, Jaime G Carbonell, Ralf D Brown, Thomas Pierce, Brian T Archibald, and Xin Liu. 1999. Learning approaches for detecting and tracking news events. *Intelligent Systems and their Applications, IEEE*, 14(4):32–43.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational linguistics*, pages 947–953. Association for Computational Linguistics.

# Automatic Term Ambiguity Detection

Tyler Baldwin   Yunyao Li   Bogdan Alexe   Ioana R. Stanoi

IBM Research - Almaden

650 Harry Road, San Jose, CA 95120, USA

{tbaldwi, yunyaoli, balexe, irs}@us.ibm.com

## Abstract

While the resolution of term ambiguity is important for information extraction (IE) systems, the cost of resolving each instance of an entity can be prohibitively expensive on large datasets. To combat this, this work looks at ambiguity detection at the term, rather than the instance, level. By making a judgment about the general ambiguity of a term, a system is able to handle ambiguous and unambiguous cases differently, improving throughput and quality. To address the term ambiguity detection problem, we employ a model that combines data from language models, ontologies, and topic modeling. Results over a dataset of entities from four product domains show that the proposed approach achieves significantly above baseline F-measure of 0.96.

## 1 Introduction

Many words, phrases, and referring expressions are semantically ambiguous. This phenomenon, commonly referred to as polysemy, represents a problem for NLP applications, many of which inherently assume a single sense. It can be particularly problematic for information extraction (IE), as IE systems often wish to extract information about only one sense of polysemous terms. If nothing is done to account for this polysemy, frequent mentions of unrelated senses can drastically harm performance.

Several NLP tasks, such as word sense disambiguation, word sense induction, and named entity disambiguation, address this ambiguity problem to varying degrees. While the goals and initial data assumptions vary between these tasks, all of them attempt to map an instance of a term seen in context to an individual sense. While making

a judgment for every instance may be appropriate for small or medium sized data sets, the cost of applying these ambiguity resolution procedures becomes prohibitively expensive on large data sets of tens to hundreds of million items. To combat this, this work zooms out to examine the ambiguity problem at a more general level.

To do so, we define an IE-centered ambiguity detection problem, which ties the notion of ambiguity to a given topical domain. For instance, given that the terms *Call of Juarez* and *A New Beginning* can both reference video games, we would like to discover that only the latter case is likely to appear frequently in non-video game contexts. The goal is to make a binary decision as to whether, given a term and a domain, we can expect every instance of that term to reference an entity in that domain. By doing so, we segregate ambiguous terms from their unambiguous counterparts. Using this segregation allows ambiguous and unambiguous instances to be treated differently while saving the processing time that might normally be spent attempting to disambiguate individual instances of unambiguous terms.

Previous approaches to handling word ambiguity employ a variety of disparate methods, variously relying on structured ontologies, gleaming insight from general word usage patterns via language models, or clustering the contexts in which words appear. This work employs an ambiguity detection pipeline that draws inspiration from all of these methods to achieve high performance.

## 2 Term Ambiguity Detection (TAD)

A term can be ambiguous in many ways. It may have **non-referential** senses in which it shares a name with a common word or phrase, such as in the films *Brave* and *2012*. A term may have referential senses **across topical domains**, such as *The Girl with the Dragon Tattoo*, which may reference either the book or the film adaptation. Terms may

also be ambiguous **within a topical domain**. For instance, the term *Final Fantasy* may refer to the video game franchise or one of several individual games within the franchise. In this work we concern ourselves with the first two types of ambiguity, as within topical domain ambiguity tends to pose a less severe problem for IE systems.

IE systems are often asked to perform extraction over a dictionary of terms centered around a single topic. For example, in brand management, customers may give a list of product names and ask for sentiment about each product. With this use case in mind, we define the *term ambiguity detection* (TAD) problem as follows: Given a term and a corresponding topic domain, determine whether the term uniquely references a member of that topic domain. That is, given a term such as *Brave* and a category such as *film*, the task is make a binary decision as to whether all instances of *Brave* reference a film by that name.

## 2.1 Framework

Our TAD framework is a hybrid approach consisting of three modules (Figure 1). The first module is primarily designed to detect non-referential ambiguity. This module examines n-gram data from a large text collection. Data from The Corpus of Contemporary American English (Davies, 2008) was used to build our n-grams.

The rationale behind the n-gram module is based on the understanding that terms appearing in non-named entity contexts are likely to be non-referential, and terms that can be non-referential are ambiguous. Therefore, detecting terms that have non-referential usages can also be used to detect ambiguity. Since we wish for the ambiguity detection determination to be fast, we develop our method to make this judgment solely on the n-gram probability, without the need to examine each individual usage context. To do so, we assume that an all lowercased version of the term is a reasonable proxy for non-named entity usages in formal text. After removing stopwords from the term, we calculate the n-gram probability of the lower-cased form of the remaining words. If the probability is above a certain threshold, the term is labeled as ambiguous. If the term is below the threshold, it is tentatively labeled as unambiguous and passed to the next module. To avoid making judgments of ambiguity based on very infrequent uses, the ambiguous-unambiguous determination

threshold is empirically determined by minimizing error over held out data.

The second module employs ontologies to detect across domain ambiguity. Two ontologies were examined. To further handle the common phrase case, Wiktionary<sup>1</sup> was used as a dictionary. Terms that have multiple senses in Wiktionary were labeled as ambiguous. The second ontology used was Wikipedia disambiguation pages. All terms that had a disambiguation page were marked as ambiguous.

The final module attempts to detect both non-referential and across domain ambiguity by clustering the contexts in which words appear. To do so, we utilized the popular Latent Dirichlet Allocation (LDA (Blei et al., 2003)) topic modeling method. LDA represents a document as a distribution of topics, and each topic as a distribution of words. As our domain of interest is Twitter, we performed clustering over a large collection of tweets. For a given term, all tweets that contained the term were used as a document collection. Following standard procedure, stopwords and infrequent words were removed before topic modeling was performed. Since the clustering mechanism was designed to make predictions over the already filtered data of the other modules, it adopts a conservative approach to predicting ambiguity. If the category term (e.g., *film*) or a synonym from the WordNet synset does not appear in the 10 most heavily weighted words for any cluster, the term is marked as ambiguous.

A term is labeled as ambiguous if any one of the three modules predicts that it is ambiguous, but only labeled as unambiguous if all three modules make this prediction. This design allows each module to be relatively conservative in predicting ambiguity, keeping precision of ambiguity prediction high, under the assumption that other modules will compensate for the corresponding drop in recall.

## 3 Experimental Evaluation

### 3.1 Data Set

**Initial Term Sets** We collected a data set of terms from four topical domains: *books*, *films*, *video games*, and *cameras*. Terms for the first three domains are lists of books, films, and video games respectively from the years 2000-2011 from dbpedia (Auer et al., 2007), while the initial terms

<sup>1</sup><http://www.wiktionary.org/>

| Tweet  | Term             | Category | Judgment |
|--|------------------|----------|----------|
| Woke up from a nap to find a beautiful mind on. #win                             | A Beautiful Mind | film     | yes      |
| I Love Tyler Perry ; He Has A Beautiful Mind.                                    | A Beautiful Mind | film     | no       |
| I might put it in the top 1. RT @CourtesyFlushMo Splice. Top 5 worst movies ever | Splice           | film     | yes      |
| Splice is a great, free replacement to iMove for your iPhone.                    | Splice           | film     | no       |

Table 1: Example tweet annotations.

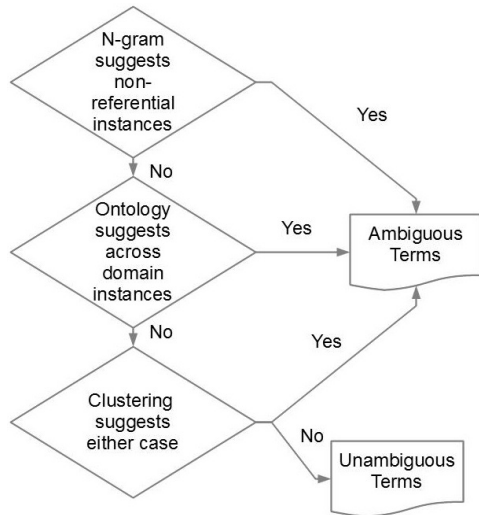


Figure 1: Overview of the ambiguity detection framework.

for *cameras* includes all the cameras from the six most popular brands on flickr<sup>2</sup>.

**Gold Standard** A set of 100 terms per domain were chosen at random from the initial term sets. Rather than annotating each term directly, ambiguity was determined by examining actual usage. Specifically, for each term, usage examples were extracted from large amounts of Twitter data. Tweets for the *video game* and *film* categories were extracted from the TREC Twitter corpus.<sup>3</sup> The less common *book* and *camera* cases were extracted from a subset of all tweets from September 1st-9th, 2012.

For each term, two annotators were given the term, the corresponding topic domain, and 10 randomly selected tweets containing the term. They were then asked to make a binary judgment as to whether the usage of the term in the tweet referred to an instance of the given category. The degree of ambiguity is then determined by calculating the percentage of tweets that did not reference a member of the topic domain. Some example judgments are given in Table 1. If all individual tweet judgments for a term were marked as referring to a

| Configuration | Precision | Recall | F-measure |
|---------------|-----------|--------|-----------|
| Baseline      | 0.675     | 1.0    | 0.806     |
| NG            | 0.979     | 0.848  | 0.909     |
| ON            | 0.979     | 0.704  | 0.819     |
| CL            | 0.946     | 0.848  | 0.895     |
| NG + ON       | 0.980     | 0.919  | 0.948     |
| NG + CL       | 0.942     | 0.963  | 0.952     |
| ON + CL       | 0.945     | 0.956  | 0.950     |
| NG + ON + CL  | 0.943     | 0.978  | 0.960     |

Table 2: Performance of various framework configurations on the test data.

member of the topic domain, the term was marked as fully unambiguous within the data examined. All other cases were considered ambiguous.<sup>4</sup>

Inter-annotator agreement was high, with raw agreement of 94% ( $\kappa = 0.81$ ). Most disagreements on individual tweet judgments had little effect on the final judgment of a term as ambiguous or unambiguous, and those that did were resolved internally.

### 3.2 Evaluation and Results

**Effectiveness** To understand the contribution of the n-gram (NG), ontology (ON), and clustering (CL) based modules, we ran each separately, as well as every possible combination. Results are shown in Table 2, where they are compared to a majority class (ambiguous) baseline.

As shown, all configurations outperform the baseline. Of the three individual modules, the n-gram and clustering methods achieve F-measure of around 0.9, while the ontology-based module performs only modestly above baseline. Unsurprisingly, the ontology method is affected heavily by its coverage, so its poor performance is primarily attributable to low recall. As noted, many IE tasks may involve sets of entities that are not found in common ontologies, limiting the ability of the ontology-based method alone. Additionally, ontologies may be apt to list cases of strict ambiguity, rather than practical ambiguity. That is, an ontology may list a term as ambiguous if there are

<sup>2</sup><http://www.flickr.com/cameras/>

<sup>3</sup><http://trec.nist.gov/data/tweets/>

<sup>4</sup>The annotated data is available at [http://researcher.watson.ibm.com/researcher/view\\_person\\_subpage.php?id=4757](http://researcher.watson.ibm.com/researcher/view_person_subpage.php?id=4757).

several potential named entities it could refer to, even if the vast majority of references were to only a single entity.

Combining any two methods produced substantial performance increases over any of the individual runs. The final system that employed all modules produced an F-measure of 0.960, a significant ( $p < 0.01$ ) absolute increase of 15.4% over the baseline.

**Usefulness** To establish that term ambiguity detection is actually helpful for IE, we conducted a preliminary study by integrating our pipeline into a commercially available rule-based IE system (Chiticariu et al., 2010; Alexe et al., 2012). The system takes a list of product names as input and outputs tweets associated with each product. It utilizes rules that employ more conservative extraction for ambiguous entities.

Experiments were conducted over several million tweets using the terms from the video game and camera domains. When no ambiguity detection was performed, all terms were treated as unambiguous. The system produced very poor precision of 0.16 when no ambiguity detection was used, due to the extraction of irrelevant instances of ambiguous objects. In contrast, the system produced precision of 0.96 when ambiguity detection was employed. However, the inclusion of disambiguation did reduce the overall recall; the system that employed disambiguation returned only about 57% of the true positives returned by the system that did not employ disambiguation. Although this reduction in recall is significant, the overall impact of disambiguation is clearly positive, due to the stark difference in precision. Nonetheless, this limited study suggests that there is substantial room for improvement in the extraction system, although this is out of the scope of the current work.

## 4 Related Work

Polysemy is a known problem for many NLP-related applications. Machine translation systems can suffer, as ambiguity in the source language may lead to incorrect translations, and unambiguous sentences in one language may become ambiguous in another (Carpuat and Wu, 2007; Chan et al., 2007). Ambiguity in queries can also hinder the performance of information retrieval systems (Wang and Agichtein, 2010; Zhong and Ng, 2012).

The ambiguity detection problem is similar to

the well studied problems of named entity disambiguation (NED) and word sense disambiguation (WSD). However, these tasks assume that the number of senses a word has is given, essentially assuming that the ambiguity detection problem has already been solved. This makes these tasks inapplicable in many IE instances where the amount of ambiguity is not known ahead of time. Both named entity and word sense disambiguation are extensively studied, and surveys on each are available (Nadeau and Sekine, 2007; Navigli, 2009).

Another task that shares similarities with TAD is word sense induction (WSI). Like NED and WSD, WSI frames the ambiguity problem as one of determining the sense of each individual instance, rather than the term as a whole. Unlike those approaches, the word sense induction task attempts to both figure out the number of senses a word has, and what they are. WSI is unsupervised, relying solely on the information that surrounds word mentions in the text.

Many different clustering-based WSI methods have been examined. Pantel and Lin (2002) employ a clustering by committee method that iteratively adds words to clusters based on their similarities. Topic model-based methods have been attempted using variations of Latent Dirichlet Allocation (Brody and Lapata, 2009) and Hierarchical Dirichlet Processes (Lau et al., 2012). Several graph-based methods have also been examined (Klapaftis and Manandhar, 2010; Navigli and Crisafulli, 2010). Although the words that surround the target word are the primary source of contextual information in most cases, additional feature sources such as syntax (Van de Cruys, 2008) and semantic relations (Chen and Palmer, 2004) have also been explored.

## 5 Conclusion

This paper introduced the term ambiguity detection task, which detects whether a term is ambiguous relative to a topical domain. Unlike other ambiguity resolution tasks, the ambiguity detection problem makes general ambiguity judgments about terms, rather than resolving individual instances. By doing so, it eliminates the need for ambiguity resolution on unambiguous objects, allowing for increased throughput of IE systems on large data sets.

Our solution for the term ambiguity detection

task is based on a combined model with three distinct modules based on n-grams, ontologies, and clustering. Our initial study suggests that the combination of different modules designed for different types of ambiguity used in our solution is effective in determining whether a term is ambiguous for a given domain. Additionally, an examination of a typical use case confirms that the proposed solution is likely to be useful in improving the performance of an IE system that does not employ any disambiguation.

Although the task as presented here was motivated with information extraction in mind, it is possible that term ambiguity detection could be useful for other tasks. For instance, TAD could be used to aid word sense induction more generally, or could be applied as part of other tasks such as coreference resolution. We leave this avenue of examination to future work.

## Acknowledgments

We would like to thank the anonymous reviewers of ACL for helpful comments and suggestions. We also thank Howard Ho and Rajasekar Krishnamurthy for help with data annotation and Shivakumar Vaithyanathan for his comments on a preliminary version of this work.

## References

- Bogdan Alexe, Mauricio A. Hernández, Kirsten Hildrum, Rajasekar Krishnamurthy, Georgia Koutrika, Meenakshi Nagarajan, Haggai Roitman, Michal Shmueli-Scheuer, Ioana Roxana Stanoi, Chitra Venkatramani, and Rohit Wagle. 2012. Surfacing time-critical insights from social media. In *SIGMOD Conference*, pages 657–660.
- 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.
- David Blei, Andrew Ng, and Micheal I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January.
- Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09*, pages 103–111, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72.
- Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40, Prague, Czech Republic, June. Association for Computational Linguistics.
- Jinying Chen and Martha Palmer. 2004. Chinese verb sense discrimination using an em clustering model with rich linguistic features. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL '04*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. SystemT: An algebraic approach to declarative information extraction. In *ACL*, pages 128–137.
- Mark Davies. 2008-. The corpus of contemporary american english: 450 million words, 1990-present. Available online at: <http://corpus.byu.edu/coca/>.
- Ioannis P. Klapaftis and Suresh Manandhar. 2010. Word sense induction & disambiguation using hierarchical random graphs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 745–755, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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 '12*, pages 591–601, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Roberto Navigli and Giuseppe Crisafulli. 2010. Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 116–126, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February.
- Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth*



*ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '02*, pages 613–619, New York, NY, USA. ACM.

Tim Van de Cruys. 2008. Using three way data for word sense discrimination. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, COLING '08*, pages 929–936, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yu Wang and Eugene Agichtein. 2010. Query ambiguity revisited: Clickthrough measures for distinguishing informational and ambiguous queries. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 361–364, Los Angeles, California, June. Association for Computational Linguistics.

Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, Jeju Island, Korea, July. Association for Computational Linguistics.

# Towards Accurate Distant Supervision for Relational Facts Extraction

Xingxing Zhang<sup>1</sup> Jianwen Zhang<sup>2\*</sup> Junyu Zeng<sup>3</sup> Jun Yan<sup>2</sup> Zheng Chen<sup>2</sup> Zhifang Sui<sup>1</sup>

<sup>1</sup>Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, China

<sup>2</sup>Microsoft Research Asia

<sup>3</sup>Beijing University of Posts and Telecommunications

<sup>1</sup>{zhangxingxing, szf}@pku.edu.cn

<sup>2</sup>{jiazhan, junyan, zhengc}@microsoft.com

<sup>3</sup>junyu.zeng@gmail.com

## Abstract

Distant supervision (DS) is an appealing learning method which learns from existing relational facts to extract more from a text corpus. However, the accuracy is still not satisfying. In this paper, we point out and analyze some critical factors in DS which have great impact on accuracy, including valid entity type detection, negative training examples construction and ensembles. We propose an approach to handle these factors. By experimenting on Wikipedia articles to extract the facts in Freebase (the top 92 relations), we show the impact of these three factors on the accuracy of DS and the remarkable improvement led by the proposed approach.

## 1 Introduction

Recently there are great efforts on building large structural knowledge bases (KB) such as Freebase, Yago, etc. They are composed of relational facts often represented in the form of a triplet, (*SrcEntity*, *Relation*, *DstEntity*), such as “(Bill Gates, BornIn, Seattle)”. An important task is to enrich such KBs by extracting more facts from text. Specifically, this paper focuses on extracting facts for existing relations. This is different from OpenIE (Banko et al., 2007; Carlson et al., 2010) which needs to discover new relations.

Given large amounts of labeled sentences, supervised methods are able to achieve good performance (Zhao and Grishman, 2005; Bunescu and Mooney, 2005). However, it is difficult to handle large scale corpus due to the high cost of labeling. Recently an approach called distant supervision (DS) (Mintz et al., 2009) was proposed, which does not require any labels on the text. It treats the extraction problem as classifying

a candidate entity pair to a relation. Then an existing fact in a KB can be used as a labeled example whose label is the relation name. Then the features of all the sentences (from a given text corpus) containing the entity pair are merged as the feature of the example. Finally a multi-class classifier is trained.

However, the accuracy of DS is not satisfying. Some variants have been proposed to improve the performance (Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012). They argue that DS introduces a lot of noise into the training data by merging the features of all the sentences containing the same entity pair, because a sentence containing the entity pair of a relation may not talk about the relation. Riedel et al. (2010) and Hoffmann et al. (2011) introduce hidden variables to indicate whether a sentence is noise and try to infer them from the data. Takamatsu et al. (2012) design a generative model to identify noise patterns. However, as shown in the experiments (Section 4), the above variants do not lead to much improvement in accuracy.

In this paper, we point out and analyze some critical factors in DS which have great impact on the accuracy but has not been touched or well handled before. First, each relation has its own schema definition, i.e., the source entity and the destination entity should be of valid types, which is overlooked in DS. Therefore, we propose a component of entity type detection to check it. Second, DS introduces many false negative examples into the training set and we propose a new method to construct negative training examples. Third, we find it is difficult for a single classifier to achieve high accuracy and hence we train multiple classifiers and ensemble them.

We also notice that Nguyen and Moschitti (2011a) and Nguyen and Moschitti (2011b) utilize external information such as more facts from Yago and labeled sentences from ACE to improve the

\* The contact author.

performance. These methods can also be equipped with the approach proposed in this paper.

## 2 Critical Factors Affecting the Accuracy

DS has four steps: (1) Detect candidate entity pairs in the corpus. (2) Label the candidate pairs using the KB. (3) Extract features for the pair from sentences containing the pair. (4) Train a multi-class classifier. Among these steps, we find the following three critical factors have great impact on the accuracy (see Section 4 for the experimental results).

**Valid entity type detection.** In DS, a sentence with a candidate entity pair a sentence with two candidate entities is noisy. First, the schema of each relation in the KB requires that the source and destination entities should be of valid types, e.g., the source and destination entity of the relation “DirectorOfFilm” should be of the types “Director” and “Film” respectively. If the two entities in a sentence are not of the valid types, the sentence is noisy. Second, the sentence may not talk about the relation even when the two entities are of the valid types. The previous works (Riedel et al., 2010; Hoffmann et al., 2011; Takamatsu et al., 2012) do not distinguish the two types of noise but directly infer the overall noise from the data. We argue that the first type of noise is very difficult to be inferred just from the noisy relational labels. Instead, we decouple the two types of noise, and utilize external labeled data, i.e., the Wikipedia anchor links, to train an entity type detection module to handle the first type of noise. We notice that when Ling and Weld (2012) studied a fine-grained NER method, they applied the method to relation extraction by adding the recognized entity tags to the features. We worry that the contribution of the entity type features may be drowned when many other features are used. Their method works well on relatively small relations, but not that well on big ones (Section 4.2).

**Negative examples construction.** DS treats the relation extraction as a multi-class classification task. For a relation, it implies that the facts of all the other relations together with the “Other” class are negative examples. This introduces many false negative examples into the training data. First, many relations are not exclusive with each other, e.g., “PlaceOfBorn” and “PlaceOfDeath”, the born place of a person can be also the death place.

Second, in DS, the “Other” class is composed of all the candidate entity pairs not existed in the KB, which actually contains many positive facts of non-Other relations because the KB is not complete. Therefore we use a different way to construct negative training examples.

**Feature space partition and ensemble.** The features used in DS are very sparse and many examples do not contain any features. Thus we employ more features. However we find it is difficult for a single classifier on all the features to achieve high accuracy and hence we divide the features into different categories and train a separate classifier for each category and then ensemble them finally.

## 3 Accurate Distant Supervision (ADS)

Different from DS, we treat the extraction problem as  $N$  binary classification problems, one for each relation. We modify the four steps of DS (Section 2). In step (1), when detecting candidate entity pairs in sentences, we use our entity type detection module (Section 3.1) to filter out the sentences where the entity pair is of invalid entity types. In step (2), we use our new method to construct negative examples (Section 3.2). In step (3), we employ more features and design an ensemble classifier (Section 3.3). In step (4), we train  $N$  binary classifiers separately.

### 3.1 Entity Type Detection

We divide the entity type detection into two steps. The first step, called boundary detection, is to detect phrases as candidate entities. The second step, called named entity disambiguation, maps a detected candidate entity to some entity types, e.g., “FilmDirector”. Note that an entity might be mapped to multiple types. For instance, “Ventura Pons” is a “FilmDirector” and a “Person”.

**Boundary Detection** Two ways are used for boundary detection. First, for each relation, from the training set of facts, we get two dictionaries (one for source entities and one for destination entities). The two dictionaries are used to detect the source and destination entities. Second, an existing NER tool (StanfordNER here) is used with the following postprocessing to filter some unwanted entities, because a NER tool sometimes produces too many entities. We first find the *compatible NER tags* for an entity type in the KB. For example,

for the type ‘‘FilmDirector’’, the compatible NER tag of Stanford NER is ‘‘Person’’. To do this, for each entity type in the KB, we match all the entities of that type (in the training set) back to the training corpus and get the probability  $P_{tag}(t_i)$  of each NER tag (including the ‘‘NULL’’ tag meaning not recognized as a named entity) recognized by the NER tool. Then we retain the top  $k$  tags  $S_{tags} = \{t_1, \dots, t_k\}$  with the highest probabilities to account for an accumulated mass  $z$ :

$$k = \arg \min_k \left( \left( \sum_{i=1}^k P_{tag}(t_i) \right) \geq z \right) \quad (1)$$

In the experiments we set  $z = 0.9$ . The *compatible ner tags* are  $S_{tags} \setminus \{\text{‘‘NULL’’}\}$ . If the retained tags contain only ‘‘NULL’’, the candidate entities recognized by NER tool will be discarded.

**Named Entity Disambiguation (NED)** With a candidate entity obtained by the boundary detection, we need a NED component to assign some entity types to it. To obtain such a NED, we leverage the anchor text in Wikipedia to generate training data and train a NED component. The referred Freebase entity and the types of an anchor link in Wikipedia can be obtained from Freebase.

The following features are used to train the NED component. **Mention Features:** Uni-grams, Bi-grams, POS tags, word shapes in the mention, and the length of the mention. **Context Features:** Uni-grams and Bi-grams in the windows of the mention (window size = 5).

### 3.2 Negative Examples Construction

Treating the problem as a multi-class classification implies introducing many false negative examples for a relation; therefore, we handle each relation with a separate binary classifier. However, a KB only tells us which entity pairs belong to a relation, i.e., it only provides positive examples for each relation. But we also need negative examples to train a binary classifier. To reduce the number of false negative examples, we propose a new method to construct negative examples by utilizing the 1-to-1/1-to-n/n-to-1/n-to-n property of a relation.

**1-to-1/n-to-1/1-to-n Relation** A 1-to-1 or n-to-1 relation is a functional relation: for a relation  $r$ , for each valid source entity  $e_1$ , there is only one unique destination entity  $e_2$  such that  $(e_1, e_2) \in r$ . However, in a real KB like Freebase, very few relations meet the exact criterion. Thus we use the

following approximate criterion instead: relation  $r$  is *approximately* a 1-to-1/n-to-1 relation if the Inequalities (2,3) hold, where  $M$  is the number of unique source entities in relation  $r$ , and  $\delta(\cdot)$  is an indicator function which returns 1 if the condition is met and returns 0 otherwise. Inequality (2) says the proportion of source entities which have exactly one counterpart destination entity should be greater than a given threshold. Inequality (3) says the average number of destination entities of a source entity should be less than the threshold. To check whether  $r$  is a 1-to-n relation, we simply swap the source and destination entities of the relation and check whether the reversed relation is a n-to-1 relation by the above two inequalities. In experiments we set  $\theta = 0.7$  and  $\gamma = 1.1$ .

$$\frac{1}{M} \sum_{i=1}^M \delta(|\{e' | (e_i, e') \in r\}| = 1) \geq \theta \quad (2)$$

$$\frac{1}{M} \sum_{i=1}^M |\{e' | (e_i, e') \in r\}| \leq \gamma \quad (3)$$

**n-to-n Relation** Relations other than 1-to-1/n-to-1/1-to-n are n-to-n relations. We approximately categorize a n-to-n relation to n-to-1 or 1-to-n by checking which one it is closer to. This is done by computing the following two values  $\alpha_{src}$  and  $\alpha_{dst}$ .  $r$  is treated as a 1-to-n relation if  $\alpha_{src} > \alpha_{dst}$  and as a 1-to-1 relation otherwise.

$$\alpha_{src} = \frac{1}{M_{src}} \sum_{i=1}^{M_{src}} |\{e' | (e_i, e') \in r\}| \quad (4)$$

$$\alpha_{dst} = \frac{1}{M_{dst}} \sum_{i=1}^{M_{dst}} |\{e' | (e', e_i) \in r\}|$$

**Negative examples** For a candidate entity pair  $(e_1, e_2)$  not in the relation  $r$  of the KB, we first determine whether it is 1-to-n or n-to-1 using the above method. If  $r$  is 1-to-1/n-to-1 and  $e_1$  exists in some fact of  $r$  as the source entity, then  $(e_1, e_2)$  is a negative example as it violates the 1-to-1/n-to-1 constraint. If  $r$  is 1-to-n, the judgement is similar and just simply swap the source and destination entities of the relation.

### 3.3 Feature Space Partition and Ensemble

The features of DS (Mintz et al., 2009) are very sparse in the corpus. We add some features in (Yao et al., 2011): **Trigger Words** (the words on the dependency path except stop words) and **Entity String** (source entity and destination entity).

| Relation              | Taka | Ensemble |
|-----------------------|------|----------|
| works_written         | 0.76 | 0.98     |
| river/basin_countries | 0.48 | 1        |
| /film/director/film   | 0.82 | 1        |
| Average               | 0.79 | 0.89     |

Table 1: Manual evaluation of top-ranked 50 relation instances for the most frequent 15 relations.

We find that without considering the reversed order of entity pairs in a sentence, the precision can be higher, but the recall decreases. For example, for the entity pair  $\langle$ Ventura Pons, Actrius $\rangle$ , we only consider sentences with the right order (e.g. *Ventura Pons is directed by Actrius.*). For each relation, we train four classifiers:  $C_1$  (without considering reversed order),  $C_2$  (considering reversed order),  $C_{1more}$  (without considering reversed order and employ more feature) and  $C_{2more}$  (considering reversed order and employ more feature). We then ensemble the four classifiers by averaging the probabilities of predictions:

$$P(y|x) = \frac{P_1 + P_2 + P_{1more} + P_{2more}}{4} \quad (5)$$

## 4 Experiments

### 4.1 Dataset and Configurations

We aimed to extract facts of the 92 most frequent relations in Freebase 2009. The facts of each relation were equally split to two parts for training and testing. Wikipedia 2009 was used as the target corpus, where 800,000 articles were used for training and 400,000 for testing. During the NED phrase, there are 94 unique entity types (they are also relations in Freebase) for the source and destination entities. Note that some entity types contain too few entities and they are discarded. We used 500,000 Wikipedia articles (2,000,000 sentences) for generating training data for the NED component. We used Open NLP POS tagger, Stanford NER (Finkel et al., 2005) and MaltParser (Nivre et al., 2006) to label/tag sentences. We employed liblinear (Fan et al., 2008) as classifiers for NED and relation extraction and the solver is L2LR.

### 4.2 Performance of Relation Extraction

**Held-out Evaluation.** We evaluate the performance on the half hold-on facts for testing. We compared performance of the  $n = 50,000$  best extracted relation instances of each method and the Precision-Recall (PR) curves are in Figure 1 and

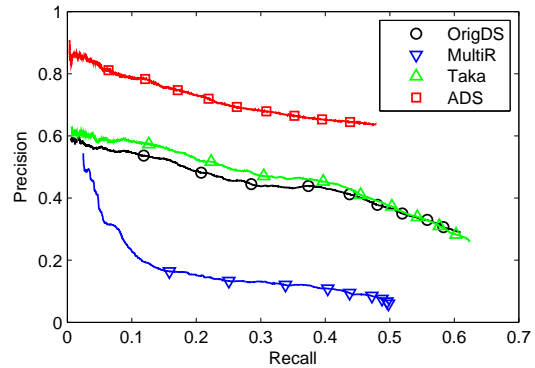


Figure 1: Performance of different methods.

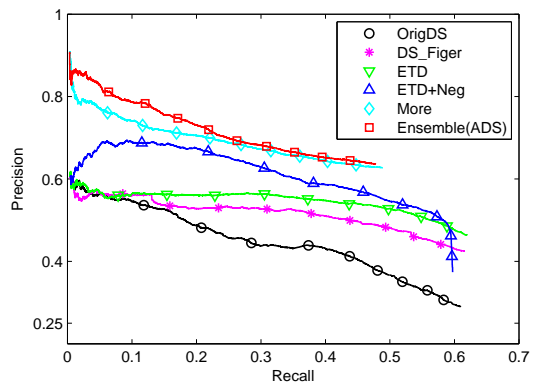


Figure 2: Contributions of different components.

Figure 2. For a candidate fact without any entity existing in Freebase, we are not able to judge whether it is correct. Thus we only evaluate the candidate facts that at least one entity occurs as the source or destination entity in the test fact set.

In Figure 1, we compared our method with two previous methods: MultiR (Hoffmann et al., 2011) and Takamatsu et al. (2012) (Taka). For MultiR, we used the author’s implementation<sup>1</sup>. We re-implemented Takamatsu’s algorithm. As Takamatsu’s dataset (903,000 Wikipedia articles for training and 400,000 for testing) is very similar to ours, we used their best reported parameters. Our method leads to much better performance.

**Manual Evaluation.** Following (Takamatsu et al., 2012), we selected the top 50 ranked (according to their classification probabilities) relation facts of the 15 largest relations. We compared our results with those of Takamatsu et al. (2012) and we achieved greater average precision (Table 1).

<sup>1</sup>available at <http://www.cs.washington.edu/ai/raphaelh/mr>  
We set  $T = 120$ , which leads to the best performance.

| $P_{micro}$ | $R_{micro}$ | $P_{macro}$ | $R_{macro}$ |
|-------------|-------------|-------------|-------------|
| 0.950       | 0.845       | 0.947       | 0.626       |

Table 2: Performance of the NED component

### 4.3 Contribution of Each Component

In Figure 2, with the entity type detection (ETD), the performance is better than the original DS method (OrigDS). As for the performance of NED in the Entity Type Detection, the Micro/Macro Precision-Recall of our NED component are in Table 2. ETD is also better than adding the entity types of the pair to the feature vector (DS.Figer)<sup>2</sup> as in (Ling and Weld, 2012). If we also employ the negative example construction strategy in Section 3.2 (ETD+Neg), the precision of the top ranked instances is improved. By adding more features (More) and employing the ensemble learning (Ensemble(ADS)) to ETD+Neg, the performance is further improved.

## 5 Conclusion

This paper dealt with the problem of improving the accuracy of DS. We find some factors are crucially important, including valid entity type detection, negative training examples construction and ensembles. We have proposed an approach to handle these issues. Experiments show that the approach is very effective.

## References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artificial intelligence, IJCAI'07*, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 2, pages 3–3.

<sup>2</sup>We use Figer (Ling and Weld, 2012) to detect entity types

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.

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 of the Association for Computational Linguistics (ACL-05)*. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June. Association for Computational Linguistics.

X. Ling and D.S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August. Association for Computational Linguistics.

Truc-Vien T. Nguyen and Alessandro Moschitti. 2011a. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages 277–282, Stroudsburg, PA, USA. Association for Computational Linguistics.

Truc-Vien T Nguyen and Alessandro Moschitti. 2011b. Joint distant and direct supervision for relation extraction. In *Proceeding of the International Joint Conference on Natural Language Processing*, pages 732–740.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *In Proc. of LREC-2006*, pages 2216–2219.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML-2010)*, pages 148–163.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th*

*Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–729, Jeju Island, Korea, July. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 419–426, Ann Arbor, Michigan, June. Association for Computational Linguistics.

# Extra-Linguistic Constraints on Stance Recognition in Ideological Debates

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 by an author from a post written for a two-sided debate in an online debate forum is a relatively new problem. We seek to improve Anand et al.'s (2011) approach to debate stance classification by modeling two types of soft extra-linguistic constraints on the stance labels of debate posts, user-interaction constraints and ideology constraints. Experimental results on four datasets demonstrate the effectiveness of these inter-post constraints in improving debate stance classification.

## 1 Introduction

While a lot of work on document-level opinion mining has involved determining the polarity expressed in a customer review (e.g., whether a review is “thumbs up” or “thumbs down”) (see Pang and Lee (2008) and Liu (2012) for an overview of the field), researchers have begun exploring new opinion mining tasks in recent years. One such task is *debate stance classification*: given a post written for a *two-sided* topic discussed in an online debate forum (e.g., “*Should abortion be banned?*”), determine which of the two sides (i.e., *for* and *against*) its author is taking.

Debate stance classification is potentially more interesting and challenging than polarity classification for at least two reasons. First, while in polarity classification sentiment-bearing words and phrases have proven to be useful (e.g., “excellent” correlates strongly with the positive polarity), in debate stance classification it is not uncommon to find debate posts where stances are not expressed in terms of sentiment words, as exemplified in Figure 1, where the author is *for* abortion.

Second, while customer reviews are typically written independently of other reviews in an online forum, the same is not true for debate posts. In

The fetus is simply a part of the mother's body and she can have an abortion because it is her human rights. Also I take this view because every woman can face with situation when two lives are at stake and the moral obligation is to save the one closest at hand — namely, that of the mother, whose life is always more immediate than that of the unborn child within her body. Permission for an abortion could then be based on psychiatric considerations such as prepartum depression, especially if there is responsible psychiatric opinion that a continued pregnancy raises the strong probability of suicide in a clinically depressed patient.

Figure 1: A sample post on abortion.

a debate forum, debate posts form *threads*, where later posts often support or oppose the viewpoints raised in earlier posts in the same thread.

Previous approaches to debate stance classification have focused on three debate settings, namely congressional floor debates (Thomas et al., 2006; Bansal et al., 2008; Balahur et al., 2009; Yessenalina et al., 2010; Burfoot et al., 2011), company-internal discussions (Murakami and Raymond, 2010), and online social, political, and ideological debates in public forums (Agrawal et al., 2003; Somasundaran and Wiebe, 2010; Wang and Rosé, 2010; 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 can potentially make stance classification of online debates more challenging than that of the other two types of debates.

Our goal in this paper is to improve the state-of-the-art supervised learning approach to debate stance classification of online debates proposed by Anand et al. (2011), focusing in particular on *ideological debates*. Specifically, we hypothesize that there are two types of soft extra-linguistic constraints on the stance labels of debate posts that,



| Domain | Number of posts | “for” posts (%) | % of 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.

if explicitly modeled, could improve a learning-based stance classification system. We refer to these two types of inter-post constraints as *user-interaction constraints* and *ideology constraints*. We show how they can be learned from stance-annotated debate posts in Sections 4.1 and 4.2, respectively.

## 2 Datasets

For our experiments, we collect debate posts from four popular *domains*, Abortion (ABO), Gay Rights (GAY), Obama (OBA), and Marijuana (MAR), from an online debate forum<sup>1</sup>. All debates are two-sided, so each post receives one of two *domain labels*, *for* or *against*, depending on whether the author of the post *supports* or *opposes* abortion, gay rights, Obama, or the legalization of marijuana.

We construct one dataset for each domain (see Table 1 for statistics). The fourth column of the table shows the percentage of posts in each domain that appear in a *thread*. More precisely, a *thread* is a tree with one or more nodes such that (1) each node corresponds to a debate post, and (2) a post  $y_i$  is the parent of another post  $y_j$  if  $y_j$  is a reply to  $y_i$ . Given a thread, we can generate *post sequences*, each of which is a path from the root of the thread to one of its leaves.

## 3 Baseline Systems

We employ as baselines two stance classification systems, Anand et al.’s (2011) approach and an enhanced version of it, as described below.

Our first baseline, Anand et al.’s approach is a supervised method that trains a stance classifier for determining whether the stance expressed in a debate post is *for* or *against* the topic. Hence, we create one training instance from each post in the training set, using the stance it expresses as its class label. Following Anand et al., we represent a training instance using three types of lexico-syntactic features, which are briefly summarized in Table 2. In our implementation, we train the

<sup>1</sup><http://www.createdebate.com/>

| Feature type | Features  |
|--------------|---|
| Basic        | Unigrams, bigrams, syntactic and POS-generalized dependencies |
| Sentiment    | LIWC counts, opinion dependencies                             |
| Argument     | Cue words, repeated punctuation, context                      |

Table 2: Anand et al.’s features.

stance classifier using SVM<sup>light</sup> (Joachims, 1999). After training, we can apply the classifier to classify the test instances, which are generated in the same way as the training instances.

Related work on stance classification of *congressional debates* has found that enforcing *author constraints* (ACs) can improve classification performance (e.g., Thomas et al. (2006), Bansal et al. (2008), Burfoot et al. (2011), Lu et al. (2012), Walker et al. (2012)). 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. We hypothesize that ACs could similarly be used to improve stance classification of ideological debates, and therefore propose a second baseline where we enhance the first baseline with ACs. Enforcing ACs is simple. We first use the learned stance classifier to classify the test posts as in the first baseline, and then *post-process* the labels of the test posts. Specifically, we sum up the confidence values<sup>2</sup> assigned to the set of test posts written by the same author for the same debate domain. If the sum is positive, then we label *all* the posts in this set as *for*; otherwise we label them as *against*.

## 4 Extra-Linguistic Constraints

In this section, we introduce two types of inter-post constraints on debate stance classification.

### 4.1 User-Interaction Constraints

We call the first type of constraints *user-interaction constraints* (UCs). UCs are motivated by the observation that the stance labels of the posts in a post sequence are not independent of each other. Consider the post sequence in Figure 2, where each post is a response to the preceding post. It shows an opening anti-abortion post (P1), followed by a pro-abortion comment (P2), which is in turn followed by another anti-abortion view (P3). While this sequence contains alternating posts from opposing stances, in general there is no hard constraint on the stance of a post given

<sup>2</sup>We use as the confidence value the signed distance of the associated test point from the SVM hyperplane.

**[P1: Anti-abortion]** There are thousands of people who want to take these children because they cannot have their own. If you do not want a child, have it and put it up for adoption. At least you will be preserving a human life rather than killing one.

**[P2: Pro-abortion]** I agree that if people don't want their babies, they should have the choice of putting it up for adoption. But it should not be made compulsory, which is essentially what happens if you ban abortion.

**[P3: Anti-abortion]** Why should it not be made compulsory? Those children have as much right to live as you and I. Besides, no one loses with adoption, so why wouldn't you utilize it?

Figure 2: A sample post sequence. P2 and P3 are replies to P1 and P2, respectively.

the preceding sequence of posts. Nevertheless, we found that in our training data, a *for* (*against*) post is followed by a *against* (*for*) post 80% of the time.

UCs aim to model the regularities in how users interact with each other in a post sequence as soft constraints. These kinds of soft constraints can be naturally encoded as *factors* over adjacent posts in a post sequence (see Kschischang et al. (2001)), which can in turn be learned by recasting stance classification as a *sequence labeling* task. In our experiments, we seek to derive the best sequence of stance labels for each post sequence of length  $\geq 1$  using a Conditional Random Field (CRF) (Lafferty et al., 2001).

We train the CRF model using the CRF implementation in Mallet (McCallum, 2002). Each training sequence corresponds to a post sequence. Each post in a sequence is represented using the same set of features as in the baselines.

After training, the resulting CRF model can be used to assign a stance sequence to each test post sequence. There is a caveat, however. Since a given test post may appear in more than one sequence, different occurrences of it may be assigned different stance labels by the CRF. To determine the final stance label for the post, we average the probabilities assigned to the *for* stance over all its occurrences; if the average is  $\geq 0.5$ , then its final label is *for*; otherwise, its label is *against*.

## 4.2 Ideology Constraints

Next, we introduce our second type of inter-post constraints, *ideology constraints* (ICs). ICs are *cross-domain, author-based* constraints: they are only applicable to debate posts written by the same author in different domains. ICs model the fact that for some authors, their stances on various issues are determined in part by their ideological

values, and in particular, their stances on different issues may be correlated. For example, someone who opposes abortion is likely to be a conservative and has a good chance of opposing gay rights. ICs aim to capture this kind of inter-domain correlation of stances. Below we describe how we implement ICs and show how they can be integrated with ACs.

### 4.2.1 Implementing Ideology Constraints

We first compute a set of conditional probabilities,  $P(\text{stance}(d_q)=s_d|\text{stance}(d_p)=s_c)$ , where (1)  $d_p, d_q \in \text{Domains}$  (i.e., the set of four domains), (2)  $s_c, s_d \in \{\text{for}, \text{against}\}$ , and (3)  $d_p \neq d_q$ . To compute  $P(\text{stance}(d_q)=s_d|\text{stance}(d_p)=s_c)$ , we (1) determine for each author  $a$  in the training set and each domain  $d_p$  the stance of  $a$  in  $d_p$  (denoted by  $\text{author-stance}(d_p, a)$ ), where  $\text{author-stance}(d_p, a)$  is computed as the majority stance labels associated with the debate posts in the training set that  $a$  wrote for  $d_p$ ; and (2) compute  $P(\text{stance}(d_q)=s_d|\text{stance}(d_p)=s_c)$  as the ratio of  $\sum_{a \in A} \text{Count}(\text{author-stance}(d_p, a)=s_c, \text{author-stance}(d_q, a)=s_d)$  to  $\sum_{a \in A} \text{Count}(\text{author-stance}(d_p, a)=s_c)$ , where  $A$  is the set of authors in the training set who posted in both  $d_p$  and  $d_q$ . It should be fairly easy to see that these conditional probabilities measure the degree of correlation between the stances in different domains.

### 4.2.2 Inference Using ILP

Recall that in our second baseline, we employ ACs to postprocess the output of the stance classifier simply by summing up the confidence values assigned to the posts written by the same author for the same debate domain. However, since we now want to enforce two types of inter-post constraints (namely, ACs and ICs), we will have to employ a more sophisticated inference mechanism. Previous work has focused on employing graph minimum cut (MinCut) as the inference algorithm. However, since MinCut suffers from the weakness of not being able to enforce negative constraints (i.e., two posts cannot receive the same label) (Bansal et al., 2008), we propose to use integer linear programming (ILP) as the underlying inference mechanism. Below we show how to implement ACs and ICs within the ILP framework.

Owing to space limitations, we refer the reader to Roth and Yih (2004) for details of the ILP framework. Briefly, ILP seeks to optimize an objective function subject to a set of linear con-

straints. Below we focus on describing the ILP program and how the ACs and ICs can be encoded.

Let  $Y = y_1, \dots, y_n$  be the set of debate posts. For each  $y_i$ , we create one (binary-valued) indicator variable  $x_i$ , which will be used in the ILP program. Let  $p_i = P(\text{for}|y_i)$  be the “benefit” of setting  $x_i$  to 1, where  $P(\text{for}|y_i)$  is provided by the CRF. Consequently, after optimization,  $y_i$ ’s stance is *for* if its  $x_i$  is set to 1. We optimize the following objective function:

$$\max \sum_i p_i x_i + (1 - p_i)(1 - x_i)$$

subject to a set of *linear* constraints, which encode the ACs and the ICs, as described below.

**Implementing author constraints.** If  $y_i$  and  $y_j$  are composed by the same author, we ensure that  $x_i$  and  $x_j$  will be assigned the same value by employing the linear constraint  $|x_i - x_j| = 0$ .

**Implementing ideology constraints.** For convenience, below we use the notation introduced in Section 4.2.1, and assume that  $y_i$  and  $y_j$  are two arbitrary posts written by the same author in domains  $d_p$  and  $d_q$ , respectively.

**Case 1:** If  $P(\text{stance}(d_q)=\text{for}|\text{stance}(d_p)=\text{for}) \geq t$ , we want to ensure that  $x_i=1 \implies x_j=1$ .<sup>3</sup> This can be achieved using the constraint  $(1 - x_j) \leq (1 - x_i)$ .

**Case 2:** If  $P(\text{stance}(d_q)=\text{against}|\text{stance}(d_p)=\text{against}) \geq t$ , we want to ensure that  $x_i=0 \implies x_j=0$ . This can be achieved using the constraint  $x_j \leq x_i$ .

**Case 3:** If  $P(\text{stance}(d_q)=\text{against}|\text{stance}(d_p)=\text{for}) \geq t$ , we want to ensure that  $x_i=1 \implies x_j=0$ . This can be achieved using the constraint  $x_j \leq (1 - x_i)$ .

**Case 4:** If  $P(\text{stance}(d_q)=\text{for}|\text{stance}(d_p)=\text{against}) \geq t$ , we want to ensure that  $x_i=0 \implies x_j=1$ . This can be achieved using the constraint  $(1 - x_j) \leq x_i$ .

Two points deserve mention. First, cases 3 and 4 correspond to negative constraints, and unlike in MinCut, they can be implemented easily in ILP. Second, if ICs are used, one ILP program will be created to perform inference over the debate posts in all four domains.

## 5 Evaluation

### 5.1 Experimental Setup

Results are expressed in terms of *accuracy* obtained via 5-fold cross validation, where accuracy

<sup>3</sup>Intuitively, if this condition is satisfied, it means that there is sufficient evidence that the two nodes from different domains should have the same stance, and so we convert the soft ICs into (hard) linear constraints in ILP. Note that  $t$  is a threshold to be tuned using development data.

| System         | ABO         | GAY         | OBA         | MAR         |
|----------------|-------------|-------------|-------------|-------------|
| Anand          | 61.4        | 62.6        | 58.1        | 66.9        |
| Anand+AC       | 72.0        | 64.9        | 62.7        | 67.8        |
| Anand+AC+UC    | 73.7        | 69.9        | 64.1        | <b>75.4</b> |
| Anand+AC+UC+IC | <b>74.9</b> | <b>70.9</b> | <b>72.7</b> | <b>75.4</b> |

Table 3: 5-fold cross-validation accuracies.

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.

### 5.2 Results

Results are shown in Table 3. Row 1 shows the results of the Anand et al. (2011) baseline (see Section 3) on the four datasets, obtained by training a SVM stance classifier using the SVM<sup>light</sup> software.<sup>4</sup> Row 2 shows the results of the second baseline, Anand et al.’s system enhanced with ACs. As we can see, incorporating ACs into Anand et al.’s system improves its performance significantly on all datasets and yields a system that achieves an average improvement of 4.6 accuracy points.<sup>5</sup>

Next, we incorporate our first type of constraints, UCs, into the better of the two baselines (i.e., the second baseline). Results of applying the CRF for modeling UCs to the test posts and post-processing them using the ACs are shown in row 3 of Table 3. As we can see, incorporating UCs into the second baseline significantly improves its performance and yields a system that achieves an average improvement of 3.93 accuracy points.

Finally, we incorporate our second type of constraints, ICs, effectively performing inference over the CRF output using ILP with ACs and ICs as the inter-post constraints. Results of this experiment are shown in row 4 of Table 3. As we can see, incorporating the ICs significantly improves the performance of the system on all but MAR and yields a system that achieves an average improvement of 2.7 accuracy points.

Overall, our inter-post constraints yield a stance classification system that significantly outperforms the better baseline on all four datasets, with an average improvement of 6.63 accuracy points.

<sup>4</sup>For all SVM experiments, the regularization parameter  $C$  is tuned using development data, but the remaining learning parameters are set to their default values.

<sup>5</sup>All significance tests are paired  $t$ -tests, with  $p < 0.05$ .

### 5.3 Discussion

Next, we make some observations on the results of applying ICs to our datasets.

First, ICs do not improve the MAR dataset. An examination of the domains reveals the reason. We find three pairs of ICs involving the other three domains — ABO, GAY, and OBA — in our training data. More specifically, the stances of the posts written by an author for these three domains are all positively co-related. In other words, if an author supports abortion, it is likely that she supports both gay rights and Obama as well. On the other hand, we find no co-relation between MAR and the remaining domains. This means that no ICs can be established between the posts in MAR and those in the remaining domains.

Second, the improvement resulting from the application of ICs is much larger on the OBA dataset than on ABO and GAY. The reason can be attributed to the fact that ICs exist more frequently between OBA and ABO and between OBA and GAY than between ABO and GAY. Specifically, ICs are seen in all five folds of the data in the first two pairs of domains, whereas they are seen in only two folds in the last pair of domains.

## 6 Related Work

Previous work has investigated the use of extra-linguistic constraints to improve stance classification. Introduced by Thomas et al. (2006), ACs are arguably the most commonly used extra-linguistic constraints. Since then, they have been employed and extended in different ways (see, for example, Bansal et al. (2008), Burfoot et al. (2011), Lu et al. (2012), and Walker et al. (2012)).

ICs are different from ACs in at least two respects. First, ICs are softer than ACs, so accurate modeling of ICs has to be based on stance-annotated data. Although we employ ICs as hard constraints (owing in part to our use of the ILP framework), they can be used directly as soft constraints in other frameworks, such as MinCut. Second, ICs are inter-domain constraints, whereas ACs are intra-domain constraints. To our knowledge, this is the first time inter-domain constraints are employed for stance classification.

There has been work related to the modeling of user interaction in a post sequence. Recall that between two adjacent posts in a post sequence that have opposing stances, there exists a *rebuttal* link. Walker et al. (2012) employ manually identified

rebuttal links as hard inter-post constraints during inference. However, since automatic discovery of rebuttal links is a non-trivial problem, employing gold rebuttal links substantially simplifies the stance classification task. Lu et al. (2012), on the other hand, predict whether a link is of type *agreement* or *disagreement* using a bootstrapped classifier. Anand et al. (2011) do not predict links. Instead, hypothesizing that the content of the preceding post in a post sequence would be useful for predicting the stance of the current post, they employ features computed based on the preceding post when training a stance classifier. Hence, unlike us, they classify each post independently of the others, whereas we classify the posts in a sequence in dependent relation to each other.

The ILP framework has been applied to perform joint inference for a variety of stance prediction tasks. Lu et al. (2012) address the task of discovering *opposing opinion networks*, where the goal is to partition the authors in a debate (e.g., gay rights) based on whether they support or oppose the given issue. To this end, they employ ILP to coordinate different sources of information. In our previous work on debate stance classification (Hasan and Ng, 2012), we employ ILP to coordinate the output of *two* classifiers: a *post-stance* classifier, which determines the stance of a debate post written for a domain (e.g., gay rights); and a *topic-stance* classifier, which determines the author’s stance on each *topic* mentioned in her post (e.g., gay marriage, gay adoption). In this work, on the other hand, we train only one classifier, but use ILP to coordinate two types of constraints, ACs and ICs.

## 7 Conclusions

We examined the under-studied task of stance classification of ideological debates. Employing our two types of extra-linguistic constraints yields a system that outperforms an improved version of Anand et al.’s approach by 2.9–10 accuracy points. While the effectiveness of ideology constraints depends to some extent on the “relatedness” of the underlying ideological domains, we believe that the gains they offer will increase with the number of authors posting in different domains and the number of related domains.<sup>6</sup>

<sup>6</sup>Only a small fraction of the authors posted in multiple domains in our datasets: 12% and 5% of them posted in two and three domains, respectively.

## References

- Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th International Conference on World Wide Web, WWW '03*, pages 529–535.
- Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2011)*, pages 1–9.
- Alexandra Balahur, Zornitsa Kozareva, and Andrés Montoyo. 2009. Determining the polarity and source of opinions expressed in political debates. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing '09*, pages 468–480.
- Mohit Bansal, Claire Cardie, and Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *Proceedings of the 22nd International Conference on Computational Linguistics: Companion volume: Posters*, pages 15–18.
- Or Biran and Owen Rambow. 2011. Identifying justifications in written dialogs. In *Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing, ICSC '11*, pages 162–168.
- Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective classification of congressional floor-debate transcripts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1506–1515.
- Kazi Saidul Hasan and Vincent Ng. 2012. Predicting stance in ideological debate with rich linguistic knowledge. In *Proceedings of the 24th International Conference on Computational Linguistics: Posters*, pages 451–460.
- Thorsten Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 44–56. MIT Press.
- Frank Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519.
- 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 18th International Conference on Machine Learning*, pages 282–289.
- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
- Yue Lu, Hongning Wang, ChengXiang Zhai, and Dan Roth. 2012. Unsupervised discovery of opposing opinion networks from forum discussions. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 1642–1646.
- Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Akiko Murakami and Rudy Raymond. 2010. Support or oppose? Classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 869–875.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
- Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning*, pages 1–8.
- Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124.
- Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335.
- Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596.
- Yi-Chia Wang and Carolyn 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*, pages 673–676.
- Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1046–1056.

# Are School-of-thought Words Characterizable?

Xiaorui Jiang<sup>¶1</sup> Xiaoping Sun<sup>¶2</sup> Hai Zhuge<sup>¶†‡3\*</sup>

<sup>¶</sup>Key Lab of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China

<sup>†</sup>Nanjing University of Posts and Telecommunications, Nanjing, China

<sup>‡</sup>Aston University, Birmingham, UK

<sup>1</sup> xiaoruijiang@gmail.com    <sup>2</sup> sunxp@kg.ict.ac.cn  
<sup>3</sup> zhuge@ict.ac.cn

## Abstract

School of thought analysis is an important yet not-well-elaborated scientific knowledge discovery task. This paper makes the first attempt at this problem. We focus on one aspect of the problem: do characteristic school-of-thought words exist and whether they are characterizable? To answer these questions, we propose a probabilistic generative School-Of-Thought (SOT) model to simulate the scientific authoring process based on several assumptions. SOT defines a school of thought as a distribution of topics and assumes that authors determine the school of thought for each sentence before choosing words to deliver scientific ideas. SOT distinguishes between two types of school-of-thought words for either the general background of a school of thought or the original ideas each paper contributes to its school of thought. Narrative and quantitative experiments show positive and promising results to the questions raised above.

## 1 Introduction

With more powerful computational analysis tools, researchers are now devoting efforts to establish a “science of better science” by analyzing the ecosystem of scientific discovery (Goth, 2012). Amongst this ambition, school of thought analysis has been identified an important fine-grained scientific knowledge discovery task. As mentioned by Teufel (2010), it is important for an experienced scientist to know which papers belong to which *school of thought* (or technical route) through years of knowledge accumulation. Schools of thought typically emerge with the evolution of a research domain or scientific topic.

\* Corresponding author.

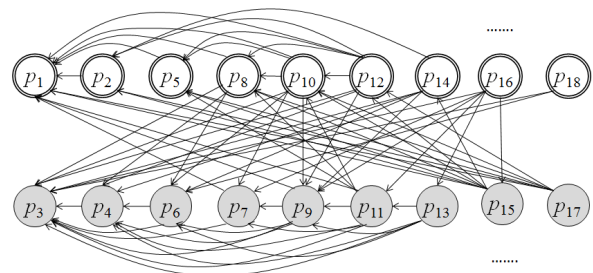


Figure 1. The citation graph of the reachability indexing domain (c.f. the RE data set in Table 1).

Take reachability indexing for example, which we will repeatedly turn to later, there are two schools of thought, the *cover-based* (since about 1990) and *hop-based* (since the beginning of the 2000s) methods. Most of the following works belong to either school of thought and thus two streams of innovative ideas emerge. Figure 1 illustrates this situation. Two chains of subsequently published papers represent two schools of thought of the reachability indexing domain. The top chain of white double-line circles and the bottom chain of shaded circles represent the *cover-based* and *hop-based* streams respectively.

However it is not easy to gain this knowledge about school of thought. Current citation indexing services are not very helpful for this kind of knowledge discovery tasks. As explained in Figure 1, papers of different schools of thought cite each other heavily and form a rather dense citation graph. An extreme example is  $p_{14}$ , which cites more *hop-based* papers than its own school of thought.

If the current citation indexing service can be equipped with school of thought knowledge, it will help scientists, especially novice researchers, a lot in grasping the core ideas of a scientific domain quickly and making their own way of innovation (Upham et al., 2010). School of thought analysis is also useful for knowledge

flow discovery (Zhuge, 2006; Zhuge, 2012), knowledge mapping (Chen, 2004; Herrera et al., 2010) and scientific paradigm summarization (Joang and Kan, 2010; Qazvinian et al., 2013) etc.

This paper makes the first attempts to unsupervised school of thought analysis. Three main aspects of school of thought analysis can be identified: determining the number of schools of thought, characterizing school-of-thought words and categorizing papers into one or several school(s) of thought (if applicable). This paper focuses on the second subproblem and leaves the other two as future work. Particularly, we purpose to investigate whether characteristic school-of-thought words exist and whether they can be automatically characterized. To answer these questions, we propose the probabilistic generative School-Of-Thought model (SOT for short) based on the following assumptions on the scientific authoring process.

**Assumption A1.** The co-occurrence patterns are useful for revealing which words and sentences are school-of-thought words and which schools of thought they describe. Take reachability indexing for example, hop-based papers try to get the “**optimum labeling**” by finding the “**densest intermediate hops**” to encode reachability information captured by an intermediate data structure called “**transitive closure contour**”. To accomplish this, they solve the “**densest subgraph problem**” on specifically created “**bipartite**” graphs centered at “**hops**” by transforming the problem into an equivalent “**minimum set cover**” framework. Thus, these bold-faced words often occur as hop-based school-of-thought words. In cover-based methods, however, one or several “**spanning tree(s)**” are extracted and “**(multiple) intervals**” are assigned to each node as reachability labels by “**pre-order**” and “**post-order traversals**”. Meanwhile, graph theory terminologies like “**root**”, “**child**” and “**ancestor**” etc. also frequently occur as cover-based school-of-thought words.

**Assumption A2.** Before writing a sentence to deliver their ideas, the authors need to determine which school of thought this sentence is to portray. This is called the *one-sot-per-sentence* assumption, where “sot” abbreviates “school of thought”. The one-sot-per-sentence assumption does not mean that authors intentionally write this way, but only simulates the outcome of the scientific paper organization. Investigations into scientific writing reveal that sentences of different schools of thought can occur anywhere and are often interleaved. This is because authors of a scientific paper not only contribute to the school of thought they follow but also discuss different

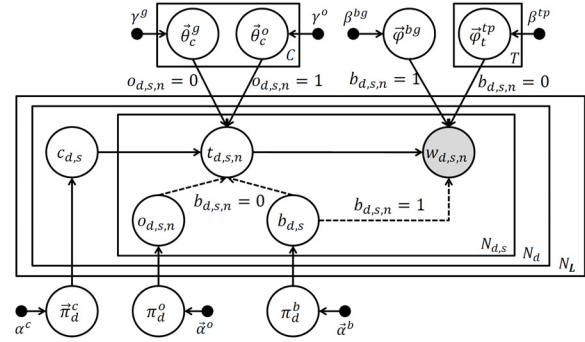


Figure 2. The SOT Model

schools of thought. For example, in the **Method** part, the authors may turn to discuss another paper (possibly of a different school of thought) for comparison. This phenomenon also occurs frequently in the **Results** or **Discussions** section. Besides, citation sentences often acknowledge related works of different schools of thought.

**Assumption A3.** All the papers of a domain talk about the general domain backgrounds. For example, reachability indexing aims to build “**compact indices**” for facilitating “**reachability queries**” between “**source**” and “**target nodes**”. Other background words include “**(complete) transitive closure**”, “**index size**” and “**reach**” etc., as well as classical graph theory terminologies like “**predecessors**” and “**successors**” etc.

**Assumption A4.** Besides contributing *original* ideas, papers of the same school of thought typically need to follow some *general* strategies that make them fall into the same school of thought. For example, all the hop-based methods follow the general ideas of designing approximate algorithms for choosing good hops, while the original ideas of each paper lead to different labeling algorithms. Scientific readers pay attention to the original ideas of each paper as well as the general ideas of each school of thought. This assumes that a word can be either a *generality* or *originality* word to deliver general and original ideas of a school of thought respectively.

## 2 The School-of-Thought Model

Figure 2 shows the proposed SOT model. SOT reflects all the assumptions made in Sect. 1. The plate notation follows Bishop (2006) where a shaded circle means an observed variable, in this context word occurrence in text, a white circle denotes either a latent variable or a model parameter, and a small solid dot represents a hyperparameter of the corresponding model parameter. The generative scientific authoring process illustrated in Figure 2 is elaborated as follows.

### Step 1. School of thought assignment (A2).

| DATA SETS | $N_L$ | $W$   | $S$  | $N_d$<br>(avg) | $C$ | SCHOOLS OF THOUGHT<br>(NUMBER OF PAPERS UNDER THIS SCHOOL OF THOUGHT)   |
|-----------|-------|-------|------|----------------|-----|---|
| RE        | 18    | 54035 | 5300 | 294            | 2   | Hop-Based (9), Cover-Based (9)  |
| NP        | 24    | 36227 | 3329 | 138            | 3   | Mention-Pair Models (14), Entity-Mention Models (5), Ranking Models (5)   |
| PP        | 20    | 21941 | 2182 | 109            | 4   | Using Single Monolingual Corpus (3), Using Monolingual Parallel Corpora (6), Using Monolingual Comparable Corpora (5), Using Bilingual Parallel Corpora (5) |
| TE        | 34    | 55671 | 5335 | 156            | 2   | Finite-State Transducer models (17), Synchronous Context-Free Grammar models (17)   |
| WA        | 18    | 19219 | 1807 | 100            | 3   | Asymmetric Models (5), Symmetric Alignment Models (9), Supervised Learning for Alignment (4)  |
| DP        | 56    | 68384 | 6021 | 107            | 3   | Transition-Based (20), Graph-Based (17), Grammar-Based (19)   |
| LR        | 44    | 77024 | 7395 | 168            | 3   | Point-wise Approach (11), Pair-wise Approach (17), List-wise Approach (16)  |

Notes: RE – REachability indexing; NP – Noun Phrase co-reference resolution; PP – ParaPhrase; TE – Translational Equivalence; WA – Word Alignment; DP – Dependency Parsing; LR – Learning to Rank;  $W$  – number of words;  $S$  – number of sentences;  $C$  – gold-standard number of schools of thought;  $N_d$  – number of sentences in document  $d$ .

Table 1. Data Sets

To simulate the one-sot-per-sentence assumption, we introduce a latent school-of-thought assignment variable  $c_{d,s}$  ( $1 \leq c_{d,s} \leq C$ , where  $C$  is the number of schools of thought) for each sentence  $s$  in paper  $d$ , dependent on which are topic assignment and word occurrence variables. As different papers and their authors have different foci, flavors and writing styles, it is appropriate to assume that each paper  $d$  has its own Dirichlet distribution of schools of thought  $\bar{\pi}_d^c \sim \text{Dir}(\bar{\alpha}^c)$  (refer to Heinrich (2008) for Dirichlet analysis of texts).  $c_{d,s}$  is thus multinomially sampled from  $\bar{\pi}_d^c$ , that is,  $c_{d,s} \sim \text{Mult}(\bar{\pi}_d^c)$ .

### Step 2. Background word emission (A3).

Before choosing a word  $w_{d,s,n}$  to deliver scientific ideas, the authors first need to determine whether this word describes domain backgrounds or depicts a specific school-of-thought. This information is indicated by the latent background word indicator variable  $b_{d,s,n} \sim \text{Bern}(\pi_d^b)$ , where  $\pi_d^b \sim \text{Beta}(\alpha_0^b, \alpha_1^b)$  is the probability of Bernoulli test.  $b_{d,s,n} = 1$  means  $w_{d,s,n}$  is a background word that is multinomially sampled from the Dirichlet background word distribution  $\bar{\varphi}^{bg} \sim \text{Dir}(\bar{\beta}^{bg})$ , i.e.  $w_{d,s,n} \sim \text{Mult}(\bar{\varphi}^{bg})$ .

### Step 3. Originality indicator assignment (A4).

If  $b_{d,s,n} = 0$ ,  $w_{d,s,n}$  is a school-of-thought word. Then the authors need to determine whether  $w_{d,s,n}$  talks about the general ideas of a certain school of thought (i.e. a *generality* word when  $o_{d,s,n} = 0$ ) or delivers original contributions to the specific school of thought (i.e. an *originality* word when  $o_{d,s,n} = 1$ ). The latent originality indicator variable  $o_{d,s,n}$  is assigned in a similar way to  $b_{d,s,n}$ .

### Step 4. Topical word emission.

SOT regards schools of thought and topics as two different levels of semantic information. A school of thought is modeled as a distribution of topics discussed by the papers of a research domain. Each topic in turn is defined as a distribution of the topical words. Reflected in Figure 1,  $\bar{\theta}_c^g$  and  $\bar{\theta}_c^o$  are Dirichlet distributions of general-

ity and originality topics respectively, with  $\gamma^g$  and  $\gamma^o$  being the Dirichlet priors. According to the assignment of the originality indicator, the topic  $t_{d,s,n}$  of the current token is multinomially selected from either  $\bar{\theta}_c^g$  ( $o_{d,s,n} = 0$ ) or  $\bar{\theta}_c^o$  ( $o_{d,s,n} = 1$ ). After that, a word  $w_{d,s,n}$  is multinomially emitted from the topical word distribution  $\bar{\varphi}_{t_{d,s,n}}^{pp}$ , where  $\bar{\varphi}_t^{pp} \sim \text{Dir}(\beta^{pp})$  for each  $1 \leq t \leq T$ .

Gibbs sampling is used for SOT model inference. Considering the logic of presentation, it is detailed in Appendix B.

## 3 Experiments

### 3.1 Datasets

Lacking standard test benchmarks, we compiled 7 data sets according to well-known recent surveys (see Appendix A). Each data set consists of several dozens of papers of the same domain. When constructing these data sets, the only place of human intervention is the de-duplication step, which means typically only one of a number of highly duplicated references is kept in the data set. Different from previous studies reviewed in Sect. 4, full texts but not abstracts are used. We extracted texts from the collected papers and removed tables, figures and sentences full of math equations or unrecognizable symbols. The statistics of the resulting data sets are listed in Table 1. The gold-standard number and the classification of schools of thoughts reflect not only the viewpoints of the survey authors but also the consensus of the corresponding research communities.

### 3.2 Qualitative Results

This section looks at the capabilities of SOT in learning background and school-of-thought words using the RE data set as an example. Given the estimated model parameters, the distributions of the school-of-thought words of SOT can be calculated as weighted sums of topical word emission probabilities ( $\varphi_{t,w}^{pp}$  for each word  $w$ ) over all the topics ( $\Sigma_t$ ) and papers ( $\Sigma_d$ ), as in Eq. (1).



| BACKGROUND WORDS   |                     |                    | SCHOOL-OF-THOUGHT WORDS   |                  |                     |  |                    |                      |
|--|---------------------|--------------------|---|------------------|---------------------|--|--------------------|----------------------|
|  |                     |                    | SOT-1 (COVER-BASED)   |                  |                     | SOT-2 (HOP-BASED)  |                    |                      |
| <b>node</b>  | <b>arc</b>          | <b>figure</b>      | node  | reachable        | find                | <b>2-hop</b>   | <b>problem</b>     | <b>hop</b>           |
| <b>closure</b>   | <b>size</b>         | deleted            | graph   | reach            | reachability        | vertex   | tree               | <b>subgraph</b>      |
| chain  | lists               | incremental        | nodes   | size             | <u><b>cover</b></u> | vertices   | edges              | proposed             |
| <b>graph</b>   | procedure           | <b>predecessor</b> | closure   | <b>chains</b>    | acyclic             | <u><b>cover</b></u>  | graph              | construction         |
| <b>nodes</b>   | arcs                | directed           | <b>tree</b>   | graphs           | database            | algorithm  | <b>approach</b>    | <b>path-hop</b>      |
| <b>compressed</b>  | update              | <b>edge</b>        | edges   | storage          | <b>traversal</b>    | size   | indexing           | <b>lin</b>           |
| list   | off-chain           | systems            | chain   | instance         | components          | chain  | <b>contour</b>     | spanning             |
| <b>transitive</b>  | <b>acyclic</b>      | <b>connected</b>   | transitive  | <b>intervals</b> | directed            | chain  | processing         | smaller              |
| successor  | <b>reduction</b>    | techniques         | <b>non-tree</b>   | <b>spanning</b>  | lists               | <b>labeling</b>  | chain              | <b>optimal</b>       |
| compression  | relation            | single             | number  | segment          | reduction           | closure  | pairs              | <b>densest</b>       |
| <b>storage</b>   | <b>source</b>       | <b>cycles</b>      | compressed  | <b>order</b>     | g.                  | reachability   | <b>compression</b> | <b>decomposition</b> |
| chains   | <b>reach</b>        | updates            | <b>path</b>   | connected        | addition            | transitive   | reachable          | dag                  |
| <b>required</b>  | effort              | depth              | edge  | component        | technique           | graphs   | property           | paths                |
| <b>index</b>   | <b>obtained</b>     | <b>materialize</b> | index   | case             | degree              | time   | figure             | data                 |
| number   | <b>component</b>    | concatenation      | list  | <b>postorder</b> | gs                  | number   | path-tree          | <b>ratio</b>         |
| <b>database</b>  | <b>path</b>         | presented          | <u><b>set</b></u>   | strongly         | successors          | <b>3-hop</b>   | <b>bipartite</b>   | nodes                |
| case   | <b>assignment</b>   | added              | <b>interval</b>   | original         | <b>structure</b>    | index  | <b>scheme</b>      | edge                 |
| technique  | <b>predecessors</b> | original           | successor   | <b>ris</b>       | single              | <b>labels</b>  | <b>density</b>     | <b>finding</b>       |
| <b>degree</b>  | addition            | <b>components</b>  | figure  | required         | paths               | query  | queries            | <b>rank</b>          |
| <b>successors</b>  | <b>indices</b>      | <b>strongly</b>    | compression   | source           | arc                 | <u><b>set</b></u>  | reach              | note                 |
| <b>destination</b> (65), <b>determine</b> (76), <b>pair</b> (77), <b>resulting</b> (84), <b>merging</b> (86), <b>reached</b> (87), <b>store</b> (96) |                     |                    | <b>root</b> (67), <b>pre-</b> (85), <b>topological</b> (96), <b>sub-tree</b> (102), <b>ancestor</b> (105), <b>child</b> (106), <b>multiple</b> (113), <b>preorder</b> (117) |                  |                     | <b>lout</b> (66), <b>segment</b> (68), <b>minimum</b> (69), <b>intermediate</b> (77), <b>greedy</b> (87), <b>faster</b> (88), <b>heuristics</b> (92), <b>approximate</b> (120) |                    |                      |

Table 2. The distributions of top-120 background and school-of-thought words.

$$\begin{aligned}
& p(w|c, o=0/1) \\
& = \sum_d \left( \frac{N_{d,v}(d, w)}{N_v(w)} \pi_{d,0/1}^o \sum_t \theta_{c,t}^{g/o} \phi_{t,w}^p \right) \quad (1)
\end{aligned}$$

The first row of Table 2 lists the top-60 background and school-of-thought words learned by SOT for the RE data set sorted in descending order of their probabilities column by column. The words at the bottom are some of the remaining characteristic words together with their positions on the top-120 list. In the experiments,  $T$  is set to 20. As the data sets are relative small, it is not appropriate to set  $T$  too large, otherwise most of the topics are meaningless or duplicate. Either case will impose additive negative influences on the usefulness of the model, for example when applied to schools of thought clustering in the next section.  $C$  is set to the gold-standard number of schools of thought as in this study we are mainly interested in whether school-of-thought words are characterizable. The problems of identifying the existence and number of schools of thought are left to future work. Other parameter settings follow Griffiths and Steyvers (2010). The learned word distributions are shown very meaningful at the first glance. They are further explained as follows.

For domain backgrounds, reachability indexing is a classical problem of the graph database “**domain**” which talks about the reachability between the “**source**” and “**destination nodes**” on a “**graph**”. Reachability “**index**” or “**indices**” aim at a “**reduction**” of the “**transitive closure**” so as to make the “**required storage**” smaller.

All current works preprocess the input graphs by “**merging strongly connected components**” into representative nodes to remove “**cycles**”.

We then give a deep investigation into the hop-based school-of-thought words (SoT-2). Cover-based ones conform well to the assumptions in Sect. 1 too. “**2-hop**”, “**3-hop**” and “**path-hop**” are three representative hop-based reachability “**labeling schemes**” (a phrase preferred by hop-based papers). Hop-based methods aim at “**finding**” the “**optimum labeling**” with “**minimum cost**” and achieving a higher “**compression ratio**” than cover-based methods. To accomplish this, hop-based methods define a “**densest subgraph problem**” on a “**bipartite**” graph, transform it to an equivalent “**set cover**” problem, and then apply “**greedy**” algorithms based on several “**heuristics**” to find “**approximate**” solutions. The “**intermediate hops**” with the highest “**density**” are found as labels and assigned to “**L<sub>out</sub>**” and “**L<sub>in</sub>**” of certain “**contour**” vertices. “**contour**” is used by hop-based methods as a concise representation of the remaining to-be-encoded reachability information.

The underlined bold italic words such as “**set**” and “**cover**” are misleading (yet not necessarily erroneous) words as both schools of thought use them heavily, but in quite different contexts, for example, a “**set**” of labels versus “**set cover**”, and “**cover(s)**” partial reachability information versus tree “**cover**”. To improve, one of our future works shall integrate multi-word expressions or  $n$ -grams (Wallach, 2006) and syntactic analysis (Griffiths et al., 2004) into the current model.

### 3.3 Quantitative Results

To see the usefulness of school-of-thought words, we use the SOT model as a way to feature space reduction for a more precise text representation in the school-of-thought clustering task. A subset of school-of-thought words whose accumulated probability exceeds a given threshold  $fsThr$  are used as the reduced feature vector. Text is represented in the vector space model weighted using  $tf-idf$ .  $K$ -means is used for clustering. To obtain a stable and reliable result, we choose 300 random seeds as initial cluster centroids, run  $K$ -means 300 times and, following the heuristic suggestion by Manning et al. (2009), output the best clustering by the minimum residual squared sum principle. Two baselines are the “RAW” method without dimension reduction and LDA-based (Blei et al., 2003) feature selection. Table 3 reports the  $F$ -measure values of different competitors. In the parentheses are the corresponding threshold values under which the reported clustering result is obtained. The larger the threshold value is, the less effective the method in dimension reduction.

Compared to the baselines, SOT has consistently the best clustering qualities. When  $fsThr \leq 0.70$ , the feature space is reduced from several thousand words to only a few hundreds. LDA is typically better than RAW (except on LR) but less efficient in dimension reduction, e.g. on WA and DP. In the latter two cases,  $fsThr = 0.80$  typically means LDA is much less efficient in feature reduction than SOT on these two data sets.

| DATA SETS | F-MEASURE ( $\beta = 2.0$ ) |                 |                    |
|-----------|-----------------------------|-----------------|--------------------|
|           | RAW                         | LDA ( $fsThr$ ) | SOT ( $fsThr$ )    |
| RE        | .7464                       | .7464 (.50)     | <b>.7482</b> (.60) |
| NP        | .4528                       | .6150 (.75)     | <b>.6911</b> (.75) |
| PP        | .3256                       | .4179 (.60)     | <b>.6025</b> (.75) |
| TE        | .2580                       | .5148 (.60)     | <b>.9405</b> (.40) |
| WA        | .3125                       | .4569 (.80)     | <b>.5519</b> (.60) |
| DP        | .4787                       | .6762 (.80)     | <b>.7155</b> (.50) |
| LR        | .5413                       | .5276 (.95)     | <b>.6583</b> (.75) |

Table 3. School-of-thought clustering results

### 4 Related Work

An early work in semantic analysis of scientific articles is Griffiths and Steyvers (2004) which focused on efficient browsing of large literature collections based on scientific topics. Other related researches include topic-based reviewer assignment (Mimno and McCallum, 2007), citation influence estimation (Dietz et al., 2007), research topic evolution (Hall et al., 2008) and expert finding (Tu et al., 2010) etc.

Another line of research is the joint modeling of topics and other types of semantic units such

as perspectives (Lin et al., 2006), sentiment (Mei et al., 2007) and opinions (Zhao et al., 2010) etc. These works also took a multi-dimensional view of document semantics. The TAM model (Paul and Girju, 2010) might be the most relevant to SOT. TAM simultaneously models aspects and topics with different assumptions from SOT and it models purely on word level.

Studies that introduce an explicit background distribution include Chemudugunta et al. (2006), Haghighi and Vanderwende (2009), and Li et al. (2010) etc. Different from these works, SOT assumes that not only some “meaningless” general-purpose words but also more meaningful words about the specific domain backgrounds can be learned. What’s more these works all model on a word level.

However, it is very useful to regard sentence as the basic processing unit, for example in the text scanning approach simulating human reading process by Xu and Zhuge (2013). Indeed, sentence-level school of thought assignment is crucial to SOT as it allows SOT to model the scientific authoring process. There are also other works that model text semantics on different levels other than words or tokens, such as Wallach (2006) on  $n$ -grams and Titov and McDonald (2008) on words within multinomially sampled sliding windows. The latter also distinguishes between different levels of topics, say global versus local topics, while in SOT such discrimination is generality versus originality topics.

### 5 Conclusion

This paper proposes a probabilistic generative model SOT for characterizing school-of-thought words. In SOT, a school of thought is modeled as a distribution of topics, with the latter defined as a distribution of topical words. School of thought assignment to each sentence is vital as it allows SOT to simulate the scientific authoring process in which each sentence conveys a piece of idea contributed to a certain school of thought as well as the domain backgrounds. Narrative and quantitative analysis show that high-quality school-of-thought words can be captured by the proposed model.

### Acknowledgements

This work is partially supported by National Science Foundation of China (No. 61075074 and No. 61070183) and funding from Nanjing University of Posts and Telecommunications. Special thanks go to Prof. Jianmin Yao at Soochow University and Suzhou Scientific Service Center of China for his advices and suggestions that help this paper finally come true.

## References

- Chemudugunta, C., Smyth P., and Steyvers, M. 2006. Modeling general ad specific aspects of documents with a probabilistic topic model. In *Proc. NIPS'06*.
- Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Ch. 8 Graphical Models. Springer.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3: 993–1022.
- Chen, C. 2004. Searching for intellectual turning points: Prograssive knowledge domain visualization. *Proc. Natl. Acad. Sci.*, 101(suppl. 1): 5303–5310.
- Dietz, L., Bickel, S., and Scheffer, T. 2007. Unsupervised prediction of citation influence. In *Proc. ICML'07*, 233–240.
- Goth, G. 2012. The science of better science. *Commun. ACM*, 55(2): 13–15.
- Griffiths, T., and Steyvers, M. 2004. Finding scientific topics. *Proc. Natl. Acad. Sci.*, 101 (suppl 1): 5228–5235.
- Griffiths, T., Steyvers, M., Blei, D. M., and Tenenbaum, J. B. 2004. Integrating topics and syntax. In *Proc. NIPS'04*.
- Haghighi, A., and Vanderwende, L. 2009. Exploring content models for multi-document summarization. In *Proc. HLT-NAACL'09*, 362–370.
- Hall, D., Jurafsky, D., and Manning, C. D. 2008. Studying the history of ideas using topic models. In *Proc. EMNLP'08*, 363–371.
- Heinrich, G. 2008. Parameter estimation for text analysis. Available at [www.arbylon.net/publications/text-est.pdf](http://www.arbylon.net/publications/text-est.pdf).
- Herrera, M., Roberts, D. C., and Gulbahce, N. 2010. Mapping the evolution of scientific fields. *PLoS ONE*, 5(5): e10355.
- Joang, C. D. V., and Kan, M.-Y. (2010). Towards automatic related work summarization. In *Proc. COLING 2010*.
- Li, P., Jiang, J., and Wang, Y. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proc. ACL'10*, 640–649.
- Lin, W., Wilson, T., Wiebe, J., and Hauptmann, A. 2006. Which side are you on? Identifying perspectives at the document and sentence levels. In *Proc. CoNLL'06*, 109–116.
- Manning, C. D., Raghavan, P., and Schütze, H. 2009. *Introduction to Information Retrieval*. Ch. 16. Flat Clustering. Cambridge University Press.
- Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proc. WWW'07*, 171–180.
- Mimno, D., and McCallum, A. 2007. Expertise modeling for matching papers with reviewers. In *Proc. SIGKDD'07*, 500–509.
- Paul, M., and Girju, R. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Proc. AAAI'10*, 545–550.
- Qazvinian, V., Radev, D. R., Mohammad, S. M., Dorr, B., Zajic, D., Whidby, M., and Moon T. (2013). Generating extractive summaries of scientific paradigms. *J. Artif. Intell. Res.*, 46: 165–201.
- Teufel, S. 2010. *The Structure of Scientific Articles*. CLSI Publications, Stanford, CA, USA.
- Titov, I., and McDonald R. 2008. Modeling online reviews with multi-grain topic models. In *Proc. WWW'08*, 111–120.
- Tu, Y., Johri, N., Roth, D., and Hockenmaier, J. 2010. Citation author topic model in expert search. In *Proc. COLING'10*, 1265–1273.
- Upham, S. P., Rosenkopf, L., Ungar, L. H. 2010. Positioning knowledge: schools of thought and new knowledge creation. *Scientometrics*, 83 (2): 555–581.
- Wallach, H. 2006. Topic modeling: beyond bag-of-words. In *Proc. ICML'06*, 977–984.
- Xu, B., and Zhuge, H. 2013. A text scanning mechanism simulating human reading process, In *Proc. IJCAI'13*.
- Zhao, X., Jiang, J., Yan, H., and Li, X. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proc. EMNLP'10*, 56–65.
- Zhuce, H. 2006. Discovery of knowledge flow in science. *Commun. ACM*, 49(5): 101–107.
- Zhuce, H. 2012. *The Knowledge Grid: Toward Cyber-Physical Society* (2nd edition). World Scientific Publishing Company, Singapore.

## Appendices

### A Survey Papers for Building Data Sets

- [RE] Yu, P. X., and Cheng, J. 2010. *Managing and Mining Graph Data*, Ch. 6, 181–215. Springer.
- [NP] Ng, V. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proc. ACL'10*, 1396–1141.
- [PP] Madnani, N., and Dorr, B. J. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Comput. Linguist.*, 36 (3): 341–387.
- [TE/WA] Lopez, A. 2008. Statistical machine translation. *ACM Comput. Surv.*, 40(3), Article 8, 49 pages.
- [DP] Kübler, S., McDonald, R., and Nivre, J. 2009. *Dependency parsing*, Ch. 3–5, 21–78. Morgan & Claypools Publishers.
- [LR] Liu, T. Y. 2011. *Learning to rank for information retrieval*, Ch. 2–4, 33–88. Springer.

### B Gibbs Sampling of the SOT Model

Using collapsed Gibbs sampling (Griffiths and Steyvers, 2004), the latent variable  $\vec{c}$  is inferred in Eq. (B1). In Eq. (B1),  $N_{c,b,o,t}(c,0,o,t)$

$$\begin{aligned}
p(c_{d,s} = c | \bar{c}^{-(d,s)}, \bullet) &\propto \prod_{t=1}^T \frac{\Gamma(N_{c,b,o,t}(c,0,0,t) + \gamma^g)}{\Gamma(N_{c,b,o,t}^{-(d,s)}(c,0,0,t) + \gamma^g)} \times \frac{\Gamma(N_{c,b,o,t}^{-(d,s)}(c,0,0,\Sigma) + T \cdot \gamma^g)}{\Gamma(N_{c,b,o,t}(c,0,0,\Sigma) + T \cdot \gamma^g)} \\
&\times \prod_{t=1}^T \frac{\Gamma(N_{c,b,o,t}(c,0,1,t) + \gamma^o)}{\Gamma(N_{c,b,o,t}^{-(d,s)}(c,0,1,t) + \gamma^o)} \times \frac{\Gamma(N_{c,b,o,t}^{-(d,s)}(c,0,1,\Sigma) + \gamma^o)}{\Gamma(N_{c,b,o,t}(c,0,1,\Sigma) + \gamma^o)} \times \frac{N_{d,c}^{-(d,s)}(d,c) + \alpha^c}{N_{d,c}^{-(d,s)}(d,\Sigma) + C \cdot \alpha^c}
\end{aligned} \tag{B1}$$

$$p(b_{d,s,n} = 1 | w_{d,s,n} = v, \bullet) \propto \frac{N_{d,b}^{-(d,s,n)}(d,1) + \alpha_1^b}{N_{d,b}^{-(d,s,n)}(d,\Sigma) + \alpha_0^b + \alpha_1^b} \times \frac{N_{b,v}^{-(d,s,n)}(1,v) + \beta^{bg}}{N_{b,v}^{-(d,s,n)}(1,\Sigma) + V \cdot \beta^{bg}} \tag{B2}$$

$$\begin{aligned}
p(b_{d,s,n} = 0, o_{d,s,n} = 0, t_{d,s,n} = t | c_{d,s} = c, \bar{b}^{-(d,s,n)}, \bar{o}^{-(d,s,n)}, \bar{t}^{-(d,s,n)}, w_{d,s,n} = v, \bullet) \\
\propto \frac{N_{d,b}^{-(d,s,n)}(d,0) + \alpha_0^b}{N_{d,b}^{-(d,s,n)}(d,\Sigma) + \alpha_0^b + \alpha_1^b} \times \frac{N_{d,b,o}^{-(d,s,n)}(d,0,0) + \alpha_0^o}{N_{d,b,o}^{-(d,s,n)}(d,0,\Sigma) + \alpha_0^o + \alpha_1^o} \\
\times \frac{N_{c,b,o,t}^{-(d,s,n)}(c,0,0,t) + \gamma^g}{N_{c,b,o,t}^{-(d,s,n)}(c,0,0,\Sigma) + T \cdot \gamma^g} \times \frac{N_{b,t,v}^{-(d,s,n)}(0,t,v) + \beta^{tp}}{N_{b,t,v}^{-(d,s,n)}(0,t,\Sigma) + V \cdot \beta^{tp}}
\end{aligned} \tag{B3}$$

$$\begin{aligned}
p(b_{d,s,n} = 0, o_{d,s,n} = 1, t_{d,s,n} = t | c_{d,s} = c, \bar{b}^{-(d,s,n)}, \bar{o}^{-(d,s,n)}, \bar{t}^{-(d,s,n)}, w_{d,s,n} = v, \bullet) \\
\propto \frac{N_{d,b}^{-(d,s,n)}(d,0) + \alpha_0^b}{N_{d,b}^{-(d,s,n)}(d,\Sigma) + \alpha_0^b + \alpha_1^b} \times \frac{N_{d,b,o}^{-(d,s,n)}(d,0,1) + \alpha_1^o}{N_{d,b,o}^{-(d,s,n)}(d,0,\Sigma) + \alpha_0^o + \alpha_1^o} \\
\times \frac{N_{c,b,o,t}^{-(d,s,n)}(c,0,1,t) + \gamma^o}{N_{c,b,o,t}^{-(d,s,n)}(c,0,1,\Sigma) + T \cdot \gamma^o} \times \frac{N_{b,t,v}^{-(d,s,n)}(0,t,v) + \beta^{tp}}{N_{b,t,v}^{-(d,s,n)}(0,t,\Sigma) + V \cdot \beta^{tp}}
\end{aligned} \tag{B4}$$

Figure B1. The SOT model inference.

is the number of words of topic  $t$  describing the common ideas ( $o = 0$ ) or original ideas ( $o = 1$ ) of school of thought  $c$ . The superscript  $-(d,s)$  means that words in sentence  $s$  of paper  $d$  are not counted.  $N_{d,c}^{-(d,s)}(d,c)$  counts the number of sentences in paper  $d$  describing school of thought  $c$  with sentence  $s$  removed from consideration. In Eqs. (B1)–(B4), the symbol  $\Sigma$  means summation over the corresponding variable. For example,

$$N_{c,b,o,t}(c,0,o,\Sigma) = \sum_{t=1,\dots,T} N_{c,b,o,t}(c,0,o,t) \tag{B5}$$

Latent variables  $\bar{b}$ ,  $\bar{o}$  and  $\bar{t}$  are jointly sampled in Eqs. (B2)–(B4).  $N_{d,b}^{-(d,s,n)}(d,b)$  counts the number of background ( $b = 0$ ) or school-of-thought ( $b = 1$ ) words in document  $d$  without counting the  $n$ -th token in sentence  $s$ .  $N_{b,v}^{-(d,s,n)}(1,v)$  is the number of times vocabulary item  $v$  occurs as background word in the literature collection without counting the  $n$ -th token in sentence  $s$  of paper  $d$ .  $N_{d,b,o}^{-(d,s,n)}(d,0,o)$  is the number of words describing either common ideas ( $o = 0$ ) or original ideas ( $o = 1$ ) of some school of thought without considering the  $n$ -th token in sentence  $s$  of paper  $d$ .  $N_{c,b,o,t}^{-(d,s,n)}(c,0,o,t)$  is the number of words of topic  $t$  in the literature collection describing either common ideas ( $o = 0$ ) or original ideas ( $o = 1$ ) of school of thought  $c$

without counting the  $n$ -th token in sentence  $s$  of paper  $d$ .  $N_{b,t,v}^{-(d,s,n)}(0,t,v)$  is the number of school-of-thought words of topic  $t$  which is instantiated by vocabulary item  $v$  in the literature collection without counting the  $n$ -th token in sentence  $s$  of paper  $d$ .

# Identifying Opinion Subgroups in Arabic Online Discussions

**Amjad Abu-Jbara**

Department of EECS  
University of Michigan  
Ann Arbor, MI, USA  
amjbara@umich.edu

**Ben King**

Department of EECS  
University of Michigan  
Ann Arbor, MI, USA  
benking@umich.edu

**Mona Diab**

Department of Computer Science  
George Washington University  
Washington DC, USA  
mtdiab@gwu.edu

**Dragomir Radev**

Department of EECS  
University of Michigan  
Ann Arbor, MI, USA  
radev@umich.edu

## Abstract

In this paper, we use Arabic natural language processing techniques to analyze Arabic debates. The goal is to identify how the participants in a discussion split into subgroups with contrasting opinions. The members of each subgroup share the same opinion with respect to the discussion topic and an opposing opinion to the members of other subgroups. We use opinion mining techniques to identify opinion expressions and determine their polarities and their targets. We use opinion predictions to represent the discussion in one of two formal representations: signed attitude network or a space of attitude vectors. We identify opinion subgroups by partitioning the signed network representation or by clustering the vector space representation. We evaluate the system using a data set of labeled discussions and show that it achieves good results.

## 1 Introduction

Arabic is one of the fastest growing languages on the internet. The number of internet users in the Arab region grew by 2500% over the past 10 years. As of January 2012, the number of Arabic-speaking internet users was 86 millions. The recent political and civic movements in the Arab World resulted in a revolutionary growth in the number of Arabic users on social networking sites. For example, Arabic is the fastest growing lan-

guage in Twitter history <sup>1</sup>.

This growth in the presence of Arab users on social networks and all the interactions and discussions that happen among them led to a huge amount of opinion-rich Arabic text being available. Analyzing this text could reveal the different viewpoints of Arab users with respect to the topics that they discuss online.

When a controversial topic is discussed, it is normal for the discussants to adopt different viewpoints towards it. This usually causes rifts in discussion groups and leads to the split of the discussants into subgroups with contrasting opinions. Our goal in this paper is to use natural language processing techniques to detect opinion subgroups in Arabic discussions. Our approach starts by identifying opinionated (subjective) text and determining its polarity (positive, negative, or neutral). Next, we determine the target of each opinion expression. The target of opinion can be a named entity mentioned in the discussion or an aspect of the discussed topic. We use the identified opinion-target relations to represent the discussion in one of two formal representations. In the first representation, each discussant is represented by a vector that encodes all his or her opinion information towards the discussion topic. In the second representation, each discussant is represented by a node in a signed graph. A positive edge connects two discussants if they have similar opinion towards the topic, otherwise the sign of the edge is nega-

---

<sup>1</sup>[http://semiocast.com/publications/2011\\_11\\_24\\_Arabic\\_highest\\_growth\\_on\\_Twitter](http://semiocast.com/publications/2011_11_24_Arabic_highest_growth_on_Twitter)

tive. To identify opinion subgroups, we cluster the vector space (the first representation) or partition the signed network (the second representation).

We evaluate this system using a data set of Arabic discussions collected from an Arabic debating site. We experiment with several variations of the system. The results show that the clustering the vector space representation achieves better results than partitioning the signed network representation.

## 2 Previous Work

Our work is related to a large body of research on opinion mining and sentiment analysis. Pang & Lee (2008) and Liu & Zhang (2012) wrote two recent comprehensive surveys about sentiment analysis and opinion mining techniques and applications.

Previous work has proposed methods for identifying subjective text that expresses opinion and distinguishing it from objective text that presents factual information (Wiebe, 2000; Hatzivassiloglou and Wiebe, 2000a; Banea et al., 2008; Riloff and Wiebe, 2003).

Subjective text may express positive, negative, or neutral opinion. Previous work addressed the problem of identifying the polarity of subjective text (Hatzivassiloglou and Wiebe, 2000b; Hassan et al., 2010; Riloff et al., 2006). Many of the proposed methods for text polarity identification depend on the availability of polarity lexicons (i.e. lists of positive and negative words). Several approaches have been devised for building such lexicons (Turney and Littman, 2003; Kanayama and Nasukawa, 2006; Takamura et al., 2005; Hassan and Radev, 2010). Other research efforts focused on identifying the holders and the targets of opinion (Zhai et al., 2010; Popescu and Etzioni, 2007; Bethard et al., 2004).

Opinion mining and sentiment analysis techniques have been used in various applications. One example of such applications is identifying perspectives (Grefenstette et al., 2004; Lin et al., 2006) which is most similar to the topic of this paper. For example, in (Lin et al., 2006), the authors experiment with several supervised and statistical models to capture how perspectives are expressed at the document and the sentence levels.

Laver et al. (2003) proposed a method for extracting perspectives from political texts. They used their method to estimate the policy positions of political parties in Britain and Ireland, on both economic and social policy dimensions.

Somasundaran and Wiebe (2009) present an unsupervised opinion analysis method for debate-side classification. They mine the web to learn associations that are indicative of opinion stances in debates and combine this knowledge with discourse information. Anand et al. (2011) present a supervised method for stance classification. They use a number of linguistic and structural features such as unigrams, bigrams, cue words, repeated punctuation, and opinion dependencies to build a stance classification model. In previous work, we proposed a method that uses participant-to-participant and participant-to-topic attitudes to identify subgroups in ideological discussions using attitude vector space clustering (Abu-Jbara and Radev, 2012). In this paper, we extend this method by adding latent similarity features to the attitude vectors and applying it to Arabic discussions. In another previous work, our group proposed a supervised method for extracting signed social networks from text (Hassan et al., 2012a). The signed networks constructed using this method were based only on participant-to-participant attitudes that are expressed explicitly in discussions. We used this method to extract signed networks from discussions and used a partitioning algorithm to detect opinion subgroups (Hassan et al., 2012b). In this paper, we extend this method by using participant-to-topic attitudes to construct the signed network.

Unfortunately, not much work has been done on Arabic sentiment analysis and opinion mining. Abbasi et al. (2008) applies sentiment analysis techniques to identify and classify document-level opinions in text crawled from English and Arabic web forums. Hassan et al. (2011) proposed a method for identifying the polarity of non-English words using multilingual semantic graphs. They applied their method to Arabic and Hindi. Abdul-Mageed and Diab (2011) annotated a corpus of Modern Standard Arabic (MSA) news text for subjectivity at the sentence level. In a later work (2012a), they expanded their corpus by la-

belonging data from more genres using Amazon Mechanical Turk. Abdul-Mageed et al. (2012a) developed SAMAR, a system for subjectivity and Sentiment Analysis for Arabic social media genres. We use this system as a component in our approach.

### 3 Approach

In this section, we present our approach to detecting opinion subgroups in Arabic discussions. We propose a pipeline that consists of five components. The input to the pipeline is a discussion thread in Arabic language crawled from a discussion forum. The output is the list of participants in the discussion and the subgroup membership of each discussant. We describe the components of the pipeline in the following subsections.

#### 3.1 Preprocessing

The input to this component is a discussion thread in HTML format. We parse the HTML file to identify the posts, the discussants, and the thread structure. We transform the Arabic content of the posts and the discussant names that are written in Arabic to the Buckwalter encoding (Buckwalter, 2004). We use AMIRAN (Diab, 2009), a system for processing Arabic text, to tokenize the text and identify noun phrases.

#### 3.2 Identifying Opinionated Text

To identify opinion-bearing text, we start from the word level. We identify the polarized words that appear in text by looking each word up in a lexicon of Arabic polarized words. In our experiments, we use Sifat (Abdul-Mageed and Diab, 2012b), a lexicon of 3982 Arabic adjectives labeled as positive, negative, or neutral.

The polarity of a word may be dependant on its context (Wilson et al., 2005). For example, a positive word that appears in a negated context should be treated as expressing negative opinion rather than positive. To identify the polarity of a word given the sentence it appears in, we use SAMAR (Abdul-Mageed et al., 2012b), a system for Subjectivity and Sentiment Analysis for Arabic social media genres. SAMAR labels a sentence that contains an opinion expression as positive, negative, or neutral taking into account the context of the opinion expression. The reported

accuracy of SAMAR on different data sets ranges between 84% and 95% for subjectivity classification and 65% and 81% for polarity classification.

#### 3.3 Identifying Opinion Targets

In this step, we determine the targets that the opinion is expressed towards. We treat as an opinion target any noun phrase (NP) that appears in a sentence that SAMAR labeled as polarized (positive or negative) in the previous step. To avoid the noise that may result from including all noun phrases, we limit what we consider as an opinion target, to the ones that appear in at least two posts written by two different participants. Since, the sentence may contain multiple possible targets for every opinion expression, we associate each opinion expression with the target that is closest to it in the sentence. For each discussant, we keep track of the targets mentioned in his/her posts and the number of times each target was mentioned in a positive/negative context.

#### 3.4 Latent Textual Similarity

If two participants share the same opinion, they tend to focus on similar aspects of the discussion topic and emphasize similar points that support their opinion. To capture this, we follow previous work (Guo and Diab, 2012; Dasigi et al., 2012) and apply Latent Dirichlet Allocation (LDA) topic models to the text written by the different participants. We use an LDA model with 100 topics. So, we represent all the text written in the discussion by each participant as a vector of 100 dimensions. The vector of each participant contains the topic distribution of the participant, as produced by the LDA model.

#### 3.5 Subgroup Detection

At this point, we have for every discussant the targets towards which he/she expressed explicit opinion and a 100-dimensions vector representing the LDA distribution of the text written by him/her. We use this information to represent the discussion in two representations. In the first representation, each discussant is represented by a vector. For every target identified in steps 3 of the pipeline, we add three entries in the vector. The first entry holds the total number of times the target was mentioned by the discussant. The second entry holds the

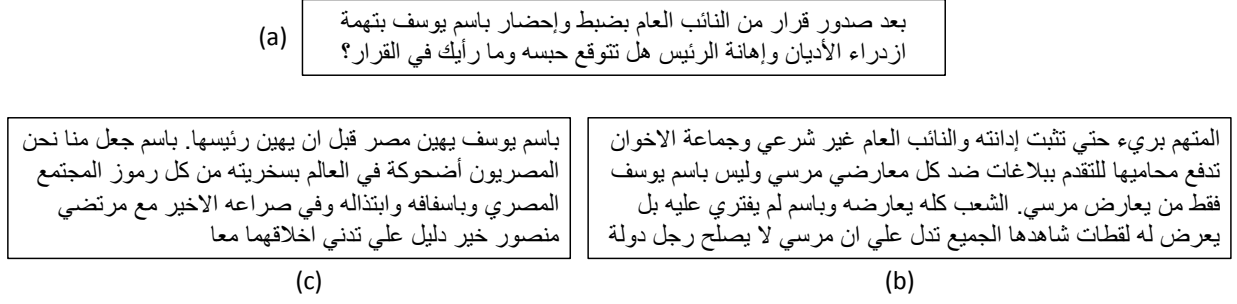


Figure 1: An example debate taken from our dataset. (a) is the discussion topic. (b) and (c) are two posts expressing contrasting viewpoints with respect to the topic.

number of times the target was mentioned in a positive context. The third entry holds the number of target mentions in a negative context. We also add to this vector the 100 topic entries from the LDA vector of that discussant. So, if the number of targets identified in step 3 of the pipeline is  $t$  then the number of entries in the discussant vector is  $3 * t + 100$ .

To identify opinion subgroups, we cluster the vector space. We experiment with several clustering algorithms including K-means (MacQueen, 1967), Farthest First (FF) (Hochbaum and Shmoys, 1985; Dasgupta, 2002), and Expectation Maximization (EM) (Dempster et al., 1977).

The second representation is a signed network representation. In this representation, each discussant is represented by a node in a graph. Two discussants are connected by an edge if they both mention at least one common target in their posts. If a discussant mentions a target multiple times in different contexts with different polarities, the majority polarity is assumed as the opinion of this discussant with respect to this target. A positive sign is assigned to the edge connecting two discussants if the number of targets that they have similar opinion towards is greater than the targets that they have opposing opinion towards, otherwise a negative sign is assigned to the edge.

To identify subgroups, we use a signed network partitioning algorithm to partition the network. Each resulting partition constitutes a subgroup. Following (Hassan et al., 2012b), we use the Dorian-Mrvar (1996) algorithm to partition the signed network. The optimization criterion aims

to have dense positive links within groups and dense negative links between groups.

The optimization function is as follows:

$$F(C) = \alpha \times |NEG| + (1 - \alpha) \times |POS| \quad (1)$$

where  $C$  is the clustering under evaluation,  $|NEG|$  is the number of negative links between nodes in the same subgroup,  $|POS|$  is the number of positive links between nodes in different subgroups, and  $\alpha$  is a parameter that specifies importance of the two terms. We set  $\alpha$  to 0.5 in all our experiments.

Clusters are selected such that  $P(C)$  is minimum. The best clustering that minimizes  $P(C)$  is found by moving nodes around clusters in a greedy way starting with a random clustering. To handle the possibility of finding a local minima, the whole process is repeated  $k$  times with random restarts and the clustering with the minimum value of  $P(C)$  is returned. We set  $k$  to 3 in all our experiments.

## 4 Data

We use data from an Arabic discussion forum called Naqeshny.<sup>2</sup> Naqeshny is a platform for two-sided debates. The debate starts when a person asks a question (e.g. which political party do you support?) and gives two possible answers or positions. The registered members of the website who are interested in the topic participate in the debate by selecting a position and then posting text to support that position and dispute the

<sup>2</sup>[www.Naqeshny.com](http://www.Naqeshny.com)



opposing position. This means that the data set is self-labeled for subgroup membership. Since the tools used in our system are trained on Modern Standard Arabic (MSA) text, we selected debates that are mostly MSA. The data set consists of 36 debates comprising a total of 711 posts written by 326 users. The average number of posts per discussion is 19.75 and the average number of participants per discussion is 13.08. Figure 1 shows an example from the data.

## 5 Experiments and Results

We use three metrics to evaluate the resulting subgroups: Purity (Manning et al., 2008), Entropy, and F-measure. We ran several variations of the system on the data set described in the previous section. In one variation, we use the signed network partitioning approach to detect subgroups. In the other variations, we use the vector space clustering approach. We experiment with different clustering algorithms. We also run two experiments to evaluate the contribution of both opinion-target counts and latent similarity features on the clustering accuracy. In one run, we use target-opinion counts only. In the other run, we use latent similarity features only. EM was used as the clustering algorithm in these two runs. Table 1 shows the results. All the results have been tested for statistical significance using a 2-tailed paired t-test. The differences between the results of the different methods shown in the table are statistically significant at the 0.05 level. The results show that the clustering approach achieves better results than the signed network partitioning approach. This can be explained by the fact that the vector representation is a richer representation and encodes all the discussants’ opinion information explicitly. The results also show that Expectation Maximization achieves significantly better results than the other clustering algorithms that we experimented with. The results also show that both latent text similarity and opinion-target features are important and contribute to the performance.

## 6 Conclusion

In this paper, we presented a system for identifying opinion subgroups in Arabic online discussions. The system uses opinion and text sim-

| System               | Purity      | F-Measure   | Entropy     |
|----------------------|-------------|-------------|-------------|
| Signed Network       | 0.71        | 0.67        | 0.68        |
| Clustering - K-means | 0.72        | 0.70        | 0.67        |
| Clustering - EM      | <b>0.77</b> | <b>0.76</b> | <b>0.50</b> |
| Clustering - FF      | 0.72        | 0.69        | 0.70        |
| Opinion-Target Only  | 0.67        | 0.65        | 0.72        |
| Text Similarity Only | 0.64        | 0.65        | 0.74        |

Table 1: Comparison of the different variations of the proposed approach

ilarity features to encode discussants’ opinions. Two approaches were explored for detecting subgroups. The first approach clusters a space of discussant opinion vectors. The second approach partitions a signed network representation of the discussion. Our experiments showed that the former approach achieves better results. Our experiments also showed that both opinion and similarity features are important.

## Acknowledgements

This research was funded in part by the Office of the Director of National Intelligence, Intelligence Advanced Research Projects Activity. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

The authors would like to thank Basma Siam for her help with collecting the data.

## References

- Ahmed Abbasi, Hsinchun Chen, and Arab 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.
- Muhammad Abdul-Mageed and Mona Diab. 2011. Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 110–118, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Muhammad Abdul-Mageed and Mona Diab. 2012a. Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios

- Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).
- Muhammad Abdul-Mageed and Mona Diab. 2012b. Toward building a large-scale arabic sentiment lexicon. In *Proceedings of the 6th International Global Word-Net Conference, Matsue, Japan*.
- Muhammad Abdul-Mageed, Sandra Kübler, and Mona Diab. 2012a. Samar: a system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, WASSA '12*, pages 19–28, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Muhammad Abdul-Mageed, Sandra Kuebler, and Mona Diab. 2012b. Samar: A system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 19–28, Jeju, Korea, July. Association for Computational Linguistics.
- Amjad Abu-Jbara and Dragomir Radev. 2012. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Jeju, Korea, July. The Association for Computational Linguistics.
- Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9, Portland, Oregon, June. Association for Computational Linguistics.
- Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *LREC'08*.
- Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*, page 2224.
- Tim Buckwalter. 2004. Issues in arabic orthography and morphology analysis. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Semitic '04*, pages 31–34, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sanjoy Dasgupta. 2002. Performance guarantees for hierarchical clustering. In *15th Annual Conference on Computational Learning Theory*, pages 351–363. Springer.
- Pradeep Dasigi, Weiwei Guo, and Mona Diab. 2012. Genre independent subgroup detection in online discussion threads: A study of implicit attitude using textual latent semantics. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 65–69, Jeju Island, Korea, July. Association for Computational Linguistics.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.
- Mona Diab. 2009. Second generation tools (amira 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.
- Patrick Doreian and Andrej Mrvar. 1996. A partitioning approach to structural balance. *Social Networks*, 18(2):149–168.
- Gregory Grefenstette, Yan Qu, James G Shanahan, and David A Evans. 2004. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of RIAO*, volume 4, pages 186–194. Citeseer.
- Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 864–872, Jeju Island, Korea, July. Association for Computational Linguistics.
- Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *ACL'10*.
- Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. 2010. What's with the attitude?: identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1245–1255.
- Ahmed Hassan, Amjad Abu-Jbara, Rahul Jha, and Dragomir Radev. 2011. Identifying the semantic orientation of foreign words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11*, pages 592–597, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012a. Extracting signed social networks from text. In *Workshop Proceedings of TextGraphs-7: Graph-based Methods for Natural Language Processing*, pages 6–14, Jeju, Republic of Korea, July. Association for Computational Linguistics.

- Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012b. Signed attitude networks: Predicting positive and negative links using linguistic analysis. In *Submitted to the Conference on Empirical Methods in Natural Language Processing*, Jeju, Korea, July. The Association for Computational Linguistics.
- Vasileios Hatzivassiloglou and Janyce Wiebe. 2000a. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING*, pages 299–305.
- Vasileios Hatzivassiloglou and Janyce M Wiebe. 2000b. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 299–305. Association for Computational Linguistics.
- Hochbaum and Shmoys. 1985. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184.
- Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP'06*, pages 355–363.
- Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(02):311–331.
- Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on?: identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 109–116. Association for Computational Linguistics.
- Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer US.
- J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- Ana-Maria Popescu and Oren Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer.
- Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP'03*, pages 105–112.
- Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 440–448. Association for Computational Linguistics.
- Swapna Somasundaran and Janyce Wiebe. 2009. 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*, pages 226–234, Suntec, Singapore, August. Association for Computational Linguistics.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL'05*, pages 133–140.
- Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
- Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP'05*, Vancouver, Canada.
- Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1272–1280. Association for Computational Linguistics.

# Extracting Events with Informal Temporal References in Personal Histories in Online Communities

Miaomiao Wen, Zeyu Zheng, Hyeju Jang, Guang Xiang, Carolyn Penstein Rosé

Language Technologies Institute, Carnegie Mellon University  
{mwen, zeyuz, hyejuj, guangx, cprose}@cs.cmu.edu

## Abstract

We present a system for extracting the dates of illness events (year and month of the event occurrence) from posting histories in the context of an online medical support community. A temporal tagger retrieves and normalizes dates mentioned informally in social media to actual month and year referents. Building on this, an event date extraction system learns to integrate the likelihood of candidate dates extracted from time-rich sentences with temporal constraints extracted from event-related sentences. Our integrated model achieves 89.7% of the maximum performance given the performance of the temporal expression retrieval step.

## 1 Introduction

In this paper we present a challenging new event date extraction task. Our technical contribution is a temporal tagger that outperforms previously published baseline approaches in its ability to identify informal temporal expressions (TE) and that normalizes each of them to an actual month and year (Chang and Manning, 2012; Strotgen and Gertz, 2010). This temporal tagger then contributes towards high performance at matching event mentions with the month and year in which they occurred based on the complete posting history of users. It does so with high accuracy on informal event mentions in social media by learning to integrate the likelihood of multiple candidate dates extracted from event mentions in time-rich sentences with temporal constraints extracted from event-related sentences.

Despite considerable prior work in temporal information extraction, to date state-of-the-art resources are designed for extracting temporally scoped facts about public figures/organizations from newswire or Wikipedia articles (Ji et al., 2011; McClosky and Manning, 2012; Garrido et

[11/15/2008] I have noticed some pulling recently and I won't start **rads** until *March*.  
[11/20/2008] It is slowwwly healing, so slowly, in fact, that she said she HOPES it will be healed by *March*, when I am supposed to start **rads**.  
[1/13/2009] I still have one last chemo to go on *the 19th* and then start **rads** in *5 wks*.  
[1/31/2009] I go for my first meeting with the **rad** onc on *2/10* (my *50th birthday!*).  
[2/23/2009] I had my first **rad** *today*.  
[3/31/2009] *Tomorrow* will be my last full **rads**  
[4/2/2009] I started **rads** in *Feb*, just did #29 *today*.  
[4/8/2009] The **rad** onc wants to see me again *next week* for a skin check as I have had cellulitis twice since *August*.  
[6/21/2010] My friend Lisa had her port put in *last week* and will begin *2 weeks* of **radiation** on *Tuesday*.

Figure 1: User posts containing keywords for the start of Radiation. Event keywords are in bold and temporal expressions are in italics.

al., 2012). When people are instead communicating informally about their lives, they refer to time more informally and frequently from their personal frame of reference rather than from an impersonal third person frame of reference. For example, they may use their own birthday as a time reference. The proportion of relative (e.g., “last week”, “two days from now”), or personal time references in our data is more than one and a half times as high as in newswire and Wikipedia. Therefore, it is not surprising that there would be difficulty in applying a temporal tagger designed for newswire to social media data (Strotgen and Gertz, 2012; Kolomiyets et al., 2011). Recent behavioral studies (Choudhury et al., 2013; Park and Choi, 2012; Wen et al., 2012) demonstrate that user-focused event mentions extracted from social media data can provide a useful timeline-like tool for studying how behavior patterns change over time in response to mentioned events. Our research contributes towards automating this work.

## 2 Task

Our task is to extract personal illness events mentioned in the posting histories of online community participants. The input to our system is

a candidate event and a posting history. The output is the event date (month and year) for the event if it occurred, or “unknown” if it did not occur. The process iterates through a list of 10 cancer events (CEs). This list includes breast cancer Diagnosis, Metastasis, Recurrence, Mastectomy, Lumpectomy, Reconstruction, Chemotherapy-Start, Chemotherapy-End, Radiation-Start and Radiation-End. For each of these target CEs, we manually designed an event keyword set that includes the name of the event, abbreviations, slang, aliases and related words.

For each of the 10 events, all sentences that mention a related event keyword are extracted from the user’s posting history. Figure 1 shows several sentences that were extracted for one user for the start date of Radiation. The task is to determine that the beginning of this user’s Radiation therapy was 2/2009. Note that the user began to post about Radiation before she started it. She first reported planning to start Radiation in March, but then rescheduled for February. Most of the TEs are non-standard and need to be resolved to calendar dates (year and month).

Once the full set of event mention sentences has been extracted for a user, all the temporal expressions (TEs) that appear in the same sentence with an event mention are resolved to a set of candidate dates. Besides a standard event-time classifier for within-sentence event-time anchoring, we leverage a new source of temporal information to train a constraint-based event-time classifier. Previous work only retrieves time-rich sentences that include both the query and some TEs (Ji et al., 2011; McClosky and Manning, 2012; Garrido et al., 2012). However, sentences that contain only the event mention but no explicit TE can also be informative. For example, the post time (usually referred to as document creation time or DCT) of the sentence “metastasis was found in my bone” might be labeled as being *after* the “metastasis” event date. These DCTs impose constraints on the possible event dates, which can be integrated with the event-time classifier, as a variant on related work (Chambers, 2012).

### 3 Related Work

Previous work on TE extraction has focused mainly on newswire text (Strotgen and Gertz, 2010; Chang and Manning, 2012). This paper presents a rule-based TE extractor that identifies

and resolves a higher percentage of nonstandard TEs than earlier state-of-art temporal taggers.

Our task is closest to the temporal slot filling track in the TAC-KBP 2011 shared task (Ji et al., 2011) and timelining task (McClosky and Manning, 2012). Their goal was to extract the temporal bounds of event relations. Our task has two key differences. First, they used newswire, Wikipedia and blogs as data sources from which they extract temporal bounds of facts found in Wikipedia infoboxes. Second, in the KBP task, the set of gold event relations are provided as input, so that the task is only to identify a date for an event that is guaranteed to have been mentioned. In our task, we provide a set of potential events. However, most of the candidate events won’t have ever been reported within a user’s posting history.

Temporal constraints have proven to be useful for producing a globally consistent timeline. In most temporal relation bound extraction systems, the constraints are included as input rather than learned by the system (Talukdar et al., 2012; Wang et al., 2011). A notable exception is McClosky et al. (2012) who developed an approach to learning constraints such as that people cannot attend school if they have not been born yet. A notable characteristic of our task is that constraints are softer. Diseases may occur in very different ways across patients. Recurring illnesses falsely appear to have an unpredictable order. Thus, there can be no universal logical constraints on the order of cancer events.

Our approach to using temporal constraints is a variant on previously published approaches. Garrido et al. (2012) made use of DCT (document creation time) as well, however, they have assumed the DCT is within the time-range of the event stated in the document, which is often not true in our data. Chambers (2012) utilized the within-sentence time-DCT relation to learn constraints for predicting DCT. We learn the event-DCT relations to produce constraints for the event date.

## 4 Corpus Annotation

We have scraped the posts, users, and profiles from a large online cancer support community. From this collection we extracted and then annotated two separate corpora, one for evaluating our TE retrieval and normalization, the other one for event date extraction.

For creating the TE extraction corpus, we ran-

domly picked one post from each of 1,000 randomly selected users. We used this sampling technique because each user tends to use a narrow range of date expression forms. From these posts, we manually extracted 601 TEs and resolved them to a specific month and year or just year if the month was not mentioned. Events not reported to have occurred were annotated as “unknown”. Our corpus for event date extraction consists of the complete posting history of 300 users that were randomly drawn from our dataset. Three annotators were provided with guidelines for how to infer the date of the events (Wen et al., 2013). We achieved .94 Kappa on identification of whether an event has a reported event date in a user’s history or not. In evaluation of agreement on extracted dates, we achieved a .99 Cronbach’s alpha. From this corpus, 509 events were annotated with occurrence dates (year and month). In our evaluation, we use data from 250 users for training, and 50 for testing.

## 5 Method

Now we explain on a more technical level how our system works on our task. Given an event and a user’s post history, the system searches for all of the sentences that contain an event keyword (*keyword sentence*) and all the sentences that contain both a keyword and a TE (*date sentence*). The TEs in the *date sentences* are resolved and then used as candidate dates for the event. For selecting among candidate dates, our model integrates two main components. First, the Date Classifier is trained from *date sentences* to predict how likely its candidate TE and the gold event date are to *overlap*. Then, because constraints over event dates can be informed by temporal relations between the event date and the DCT, the Constraint-based Classifier provides an indication of the plausibility of candidate dates. The integrated system combines the predictions from both classifiers.

### 5.1 Temporal Tagger

We design a rule-based temporal tagger that is built using regular expression patterns to recognize informal TEs. Similar to SUTime (Chang and Manning, 2012), we identify and resolve a wide range of non-standard TE types such as “Feb ’07 (2/2007)”. The additional types of TE we handle include: 1)**user-specific TEs**: A user’s age, cancer anniversary and survivorship can provide

temporal information about the user’s CEs. We obtain the birth date of users from their personal profile to resolve age date expressions such as “at the age of 57”. 2)**non-whole numbers** such as “a year and half” and “1/2 weeks”. 3)**abbreviations of time units** : e.g. “wk” as the abbreviation of “week”. 4)**underspecified month mentions**, we resolve the year information according to the DCT month, the mentioned month and the verb tense.

### 5.2 Date Classifier

We train a MaxEnt classifier to predict the temporal relationship between the retrieved TE and the event date as *overlap* or *no-overlap*, similar to the within-sentence event-time anchoring task in TempEval-2 (UzZaman and Allen, 2010). Features for the classifier include many of those in (McClosky and Manning, 2012; Yoshikawa et al., 2009): namely, event keyword and its dominant verb, verb and preposition that dominate TE, dependency path between TE and keyword and its length, unigram and bigram word and POS features. New features include the Event-Subject, Negative and Modality features. In online support groups, users not only tell stories about themselves, they also share other patients’ stories (as shown in Figure 1). So we add subject features to remove this kind of noise, which includes the governing subject of the event keyword and its POS tag. Modality features include the appearance of modals before the event keyword (e.g., may, might). Negative features include the presence/absence of negative words (e.g., no, never). These two features indicate a hypothetical or counter-factual expression of the event.

To calculate the likelihood of a candidate date for an event, we need to aggregate the hard decisions from the classifier. Let  $DS_u$  be the set of the user’s *date sentences*, let  $D_u$  be the set of dates resolved from each TE. We represent a MaxEnt classifier by  $P_{relation}(R|t, ds)$  for a candidate date  $t$  in *date sentence*  $ds$  and possible relation  $R = \{overlap, no-overlap\}$ . We map the distribution over relations to a distribution over dates by defining  $P_{DateSentence}(t|DS_u)$ :

$$P_{DateSentence}(t|DS_u) = \frac{1}{Z(D_u)} \sum_{t_j \in D_u} \delta_{t_j}(t) P_{relation}(overlap|t_j, ds_j) \quad (1)$$

$$\delta_{t_j}(t) = \begin{cases} 1 & \text{if } t = t_j \\ 0 & \text{otherwise} \end{cases}$$

We refer to this model as the Date Classifier.

### 5.3 Constraint-based Classifier

Previous work only retrieves time-rich sentences (i.e., *date sentences*) (Ling and Weld, 2010; Ji et al., 2011; McClosky and Manning, 2012; Garrido et al., 2012). However, *keyword sentences* can inform temporal constraints for events and therefore should not be ignored. For example, “Well, I’m officially a Radiation grad!” indicates the user has done radiation by the time of the post (DCT). “Radiation is not a choice for me.” indicates the user probably never had radiation. The topic of the sentence can also indicate the temporal relation. For example, *before* chemotherapy, the users tend to talk about choices of drug combinations. *After* chemotherapy, they talk about side-effects.

This section departs from the above Date Classifier and instead predicts whether each *keyword sentence* is posted *before* or *overlap-or-after* the user’s event date. The goal is to automatically learn time constraints for the event. This task is similar to the sentence event-DCT ordering task in TempEval-2 (UzZaman and Allen, 2010). We create training examples by computing the temporal relation between the DCT and the user’s gold event date. If the user has not reported an event date, the label should be *unknown*.

We train a MaxEnt classifier on each event mention paired with its corresponding DCT. All the features used in the classifier component that are not related to the TEs are included. Let  $KS_u$  be the set of the user’s *keyword sentences*, let  $D_u$  be the set of dates resolved from each *date sentence*. We define a MaxEnt classifier by  $P_{relation}(R|ks)$  for a keyword sentence  $ks$  and possible relation  $R = \{before, overlap-or-after, unknown\}$ . DCT is the post time of the *keyword sentence*  $ks$ . The  $rel(DCT, t)$  function simply determines if the DCT is *before* or *overlap-or-after* the candidate date  $t$ . We map this distribution over relations to a distribution over dates by defining  $P_{KeywordSentence}(t, KS_u)$ :

$$P_{KeywordSentence}(t, KS_u) = \frac{1}{Z(D_u)} \sum_{ks_j \in KS_u} P_{relation}(rel(dct_j, t)|ks_j) \quad (2)$$

$$rel(dct, t) = \begin{cases} before & \text{if } dct < t \\ overlap-or-after & \text{if } dct \geq t \end{cases}$$

### 5.4 Integrated Model

Given the Date Classifier of Section 5.2 and the Constraint-based Classifier of Section 5.3, we create a *Integrated Model* combining the two with the following linear interpolation as follows:

$$P(t|posts_u) = \lambda P_{DateSentence}(t|DS_u) + (1 - \lambda) P_{KeywordSentence}(t|KS_u)$$

where  $t$  is a candidate event date. The system will output  $t$  that maximizes  $P(t|posts_u)$  and *unknown* if  $DS_u$  is empty.  $\lambda$  was set to 0.7 by maximizing accuracy using five-fold cross-validation over the training set.

## 6 Evaluation Metric and Results

### 6.1 Temporal Expression Retrieval

We compare our temporal tagger’s performance with SUTime (Chang and Manning, 2012) on the 601 manually extracted TEs. We exclude user-specific TEs such as birthday references since SUTime cannot handle those. We first evaluate identification of the extent of a TE and then production of the correctly resolved date for each recognized expression. Table 1 shows that our tagger has significantly higher precision and recall for both.

|               |            | P           | R           | F1          |
|---------------|------------|-------------|-------------|-------------|
| Extents       | SUTime     | 97.5        | 75.4        | 85.0        |
|               | Our tagger | <b>97.9</b> | <b>91.8</b> | <b>94.8</b> |
| Normalization | SUTime     | 89.4        | 71.2        | 79.3        |
|               | Our tagger | <b>91.3</b> | <b>85.5</b> | <b>88.3</b> |

Table 1: Temporal expression retrieval results

### 6.2 Event-date Extraction

#### 6.2.1 Evaluation metric

The extracted date is only considered correct if it completely matches the gold date. For less than 4% of users, we have multiple dates for the same event (e.g., a user had a mastectomy twice). Similar to the evaluation metric in a previous study (Ji et al., 2011), in these cases, we give the system the benefit of the doubt and the extracted date is considered correct if it matches one of the gold dates. In previous work (McClosky and Manning, 2012; Ji et al., 2011), the evaluation metric score is defined as  $1/((1 + |d|))$  where  $d$  is the difference between the values in years. We choose a much stricter evaluation metric because we need a precise event date to study user behavior changes.

#### 6.2.2 Baselines and oracle

Based on our temporal tagger, we provide two baselines to describe heuristic methods of aggregating the hard decisions from the classifier

|                |       | Baseline1 |     |     | Baseline2 |     |     | Date |     |     | Integrated |     |     | Oracle |
|----------------|-------|-----------|-----|-----|-----------|-----|-----|------|-----|-----|------------|-----|-----|--------|
| CE             | count | P         | R   | F1  | P         | R   | F1  | P    | R   | F1  | P          | R   | F1  | F1     |
| Diagnosis      | 112   | .64       | .70 | .67 | .60       | .66 | .63 | .68  | .75 | .71 | .68        | .75 | .71 | .80    |
| Metastasis     | 7     | .16       | .58 | .25 | .12       | .43 | .19 | .25  | .86 | .39 | .25        | .86 | .39 | .86    |
| Recurrence     | 14    | .14       | .35 | .20 | .11       | .29 | .16 | .13  | .36 | .19 | .13        | .36 | .19 | .47    |
| Chemo-start    | 54    | .49       | .61 | .54 | .42       | .52 | .46 | .52  | .66 | .58 | .58        | .74 | .65 | .76    |
| Chemo-end      | 43    | .44       | .59 | .50 | .36       | .49 | .42 | .47  | .63 | .54 | .48        | .66 | .56 | .84    |
| Rad-start      | 38    | .35       | .47 | .40 | .30       | .40 | .34 | .36  | .47 | .41 | .40        | .53 | .46 | .64    |
| Rad-end        | 35    | .48       | .63 | .54 | .30       | .39 | .34 | .50  | .66 | .57 | .50        | .66 | .57 | .84    |
| Mastectomy     | 68    | .58       | .71 | .64 | .52       | .62 | .57 | .62  | .76 | .68 | .62        | .76 | .68 | .77    |
| Lumpectomy     | 33    | .49       | .71 | .58 | .43       | .76 | .46 | .46  | .79 | .58 | .46        | .79 | .62 | .91    |
| Reconstruction | 43    | .38       | .57 | .46 | .29       | .44 | .35 | .41  | .63 | .50 | .43        | .65 | .52 | .86    |

Table 2: Event-level five-fold cross-validation performance of models and baselines on training data.

learned in Section 5.3. The first baseline, Baseline1, is to pick the date with the highest classifier’s prediction confidence. The second baseline, Baseline2, is along the same lines as the Combined Classifier used in (McClosky and Manning, 2012). For example, if the candidate date is “6/2009” and we have retrieved two TEs that are resolved to “6/2009” and “4/2008”, then  $P(\text{“6/2009”}) = P_{relation}(\text{overlap}|\text{“6/2009”}) \times P_{relation}(\text{no-overlap}|\text{“4/2008”})$ .

To set an upper bound on performance given our TE retrieval system, we calculate the oracle score by considering an extraction as correct if the gold date is one of the retrieved candidate dates. The oracle score can differ from a perfect score since we can only use candidate temporal expressions if (a)the relation is known and (b)mentions of the event are retrievable, (c)the TE and event keyword appear in the same sentence, and (d)our temporal tagger is able to recognize and resolve it correctly.

### 6.2.3 Results

We present the performance of our models, baselines and the oracle in Table 2. Both the Date Classifier and Integrated model significantly outperform the baselines ( $p < 0.0001$ , McNemar’s test, 2-tailed). This shows the value of our approach to leveraging redundancy of event date mentions. Incorporating time constraints further improves the F1 of the Date Classifier by 3%. The Integrated model achieves 89.7% of the oracle result.

| Model            | P           | R           | F1          |
|------------------|-------------|-------------|-------------|
| Baseline1        | 46.1        | 63.7        | 53.5        |
| Baseline2        | 39.3        | 54.4        | 45.6        |
| Date Classifier  | 49.6        | 67.7        | 57.3        |
| Integrated Model | <b>51.0</b> | <b>69.3</b> | <b>58.8</b> |
| Oracle           | 77.3        | 77.3        | 77.3        |

Table 3: Performance of systems on the test set.

Table 3 shows the performance of our systems and baselines on individual event types. The Joint

Model derives most of its improvement from performance related to the Chemotherapy/Radiation-start date. This is mainly because Chemotherapy and Radiation last for a period of time and there are more event-related discussions containing the event keyword. None of our systems improves on cancer Metastasis and Recurrence. This is likely due to the sparsity of these events.

## 7 Conclusion

We presented a novel event date extraction task that requires extraction and resolution of non-standard TEs, namely personal illness event dates, from the posting histories of online community participants. We constructed an evaluation corpus and designed a temporal tagger for non-standard TEs in social media. Using a much stricter standard correctness measure than in previous work, our method achieves promising results that are significantly better than two types of baseline. By creating an analogous keyword set, our event date extraction method could be easily adapted to other datasets.

## 8 Acknowledgments

We want to thank Dong Nguyen and Yi-chia Wang, who helped provide the data for this project. The research reported here was supported by National Science Foundation grant IIS-0968485.



## References

- Javier Artiles, Qi Li, Taylor Cassidy, Suzanne Tamang, and Heng Ji. 2011. CUNY BLENDER TACKBP2011 Temporal Slot Filling System Description. In *Proceedings of Text Analysis Conference (TAC)*.
- Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.
- Nathanael Chambers. 2012. Labeling documents with timestamps: Learning from their time expressions. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.
- Angel X. Chang and Christopher D. Manning. 2012. SUTIME: A library for recognizing and normalizing time expressions. In *8th International Conference on Language Resources and Evaluation (LREC 2012)*.
- De Choudhury, M., Counts, S., and Horvitz, E. 2013. Major Life Changes and Behavioral Markers in Social Media: Case of Childbirth. In *Proc. CSCW 2013*.
- Guillermo Garrido, Anselmo Penas, Bernardo Cabaleiro, and Alvaro Rodrigo. 2012. Temporally Anchored Relation Extraction. In *Proceedings of the 50th annual meeting of the association for computational linguistics*.
- Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC 2011 Knowledge Base Population track. In *Proceedings of Text Analysis Conference (TAC)*.
- Hyuckchul Jung, James Allen, Nate Blaylock, Will de Beaumont, Lucian Galescu, and Mary Swift. 2011. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. In *Proceedings of BioNLP 2011*.
- Oleksandr Kolomiyets, Steven Bethard and Marie-Francine Moens. 2011. Model-Portability Experiments for Textual Temporal Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.
- Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics*.
- Xiao Ling and Daniel S Weld. 2010. Temporal information extraction. *Proceedings of the Twenty Fifth National Conference on Artificial Intelligence*.
- David McClosky and Christopher D. Manning. 2012. Learning Constraints for Consistent Timeline Extraction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP2012)*.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th annual meeting of the Association for Computational Linguistics*.
- Heekyong Park and Jinwook Choi. 2012. V-model: a new innovative model to chronologically visualize narrative clinical texts. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems. ACM*.
- Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.
- James Pustejovsky, Jos M. Castao, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *TimeML: Robust specification of event and temporal expressions in text. In New Directions in Question Answering'03*.
- James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev. 2003. The Timebank corpus. In *Corpus Linguistics*.
- Preethi Raghavan, Eric Fosler-Lussier, and Albert M. Lai. 2012. Learning to Temporally Order Medical Events in Clinical Text. In *Proceedings of the 50th annual meeting of the Association for computational Linguistics*.
- Jannik Strotgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *SemEval '10*.
- Jannik Strotgen and Michael Gertz. 2012. Temporal Tagging on Different Domains: Challenges, Strategies, and Gold Standards. In *LREC2012*.
- Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012. Coupled temporal scoping of relational facts. In *Proceedings of the fifth ACM international conference on Web search and data mining. ACM*.
- Naushad UzZaman and James F. Allen. 2010. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.
- Yafang Wang, Bing Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2011. Harvesting facts from textual web sources by constrained label propagation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*.

- Miaomiao Wen, Hyeju Jang, and Carolyn Rosé. 2013. Coding Manual for Illness Event Date Extraction. *Carnegie Mellon University, School of Computer Science, Language Technology Institute*.
- K.-Y. Wen, F. McTavish, G. Kreps, M. Wise, and D. Gustafson. 2012. From diagnosis to death: A case study of coping with breast cancer as seen through online discussion group messages. *Journal of Computer-Mediated Communication*, 16:331-361.
- 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*.
- Li Zhou and George Hripcsak. 2007. Temporal reasoning with medical data—a review with emphasis on medical natural language processing. *Journal of biomedical informatics* 40.2 (2007): 183.

# Multimodal DBN for Predicting High-Quality Answers in cQA portals

Haifeng Hu, Bingquan Liu, Baoxun Wang, Ming Liu, Xiaolong Wang

School of Computer Science and Technology

Harbin Institute of Technology, China

{hfhu, liubq, bxwang, mliu, wangxl}@insun.hit.edu.cn

## Abstract

In this paper, we address the problem for predicting cQA answer quality as a classification task. We propose a multimodal deep belief nets based approach that operates in two stages: First, the joint representation is learned by taking both textual and non-textual features into a deep learning network. Then, the joint representation learned by the network is used as input features for a linear classifier. Extensive experimental results conducted on two cQA datasets demonstrate the effectiveness of our proposed approach.

## 1 Introduction

Predicting the quality of answers in community based Question Answering (cQA) portals is a challenging task. One straightforward approach is to use textual features as a text classification task (Agichtein et al., 2008). However, due to the word over-sparsity and inherent noise of user-generated content, the classical bag-of-words representation, is not appropriate to estimate the quality of short texts (Huang et al., 2011). Another typical approach is to leverage non-textual features to automatically identify high quality answers (Jeon et al., 2006; Zhou et al., 2012). However, in this way, the mining of meaningful textual features usually tends to be ignored.

Intuitively, combining both textual and non-textual information extracted from answers is helpful to improve the performance for predicting the answer quality. However, textual and non-textual features usually have different kinds of representations and the correlations between them are highly non-linear. Previous study (Ngiam et al., 2011) has shown that it is hard for a shallow model to discover the correlations over multiple sources.

To this end, a deep learning approach, called

multimodal deep belief nets (mDBN), is introduced to address the above problems to predict the answer quality. The approach includes two stages: feature learning and supervised training. In the former stage, a specially designed deep network is given to learn the unified representation using both textual and non-textual information. In the latter stage, the outputs of the network are then used as inputs for a linear classifier to make prediction.

The rest of this paper is organized as follows: The related work is surveyed in Section 2. Then, the proposed approach and experimental results are presented in Section 3 and Section 4 respectively. Finally, conclusions and future directions are drawn in Section 5.

## 2 Related Work

The typical way to predict the answer quality is exploring various features and employing machine learning methods. For example, Jeon et al. (2006) have proposed a framework to predict the quality of answers by incorporating non-textual features into a maximum entropy model. Subsequently, Agichtein et al. (2008) and Bian et al. (2009) both leverage a larger range of features to find high quality answers. The deep research on evaluating answer quality has been taken by Shah and Pomerantz (2010) using the logistic regression model. We borrow some of their ideas in this paper.

In deep learning field, extensive studies have been done by Hinton and his co-workers (Hinton et al., 2006; Hinton and Salakhutdinov, 2006; Salakhutdinov and Hinton, 2009), who initially propose the deep belief nets (DBN). Wang et al (2010; 2011) firstly apply the DBNs to model semantic relevance for qa pairs in social communities. Meanwhile, the feature learning for disparate sources has also been the hot research topic. Lee et al. (2009) demonstrate that the hidden representations computed by a convolutional DBN make excellent features for visual recognition.

### 3 Approach

We consider the problem of high-quality answer prediction as a classification task. Figure 1 summarizes the framework of our proposed approach. First, textual features and non-textual features ex-

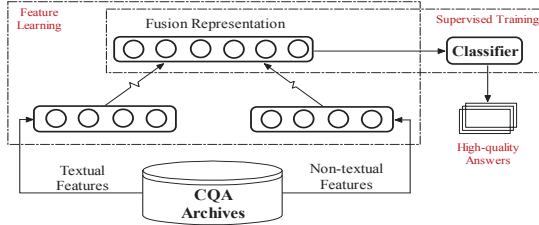


Figure 1: Framework of our proposed approach.

tracted from cQA portals are used to train two DBN models to learn the high-level representations independently for answers. The two high-level representations learned by the deep architectures are then joined together to train a RBM model. Finally, a linear classifier is trained with the final shared representation as input to make prediction.

In this section, a deep network for the cQA answer quality prediction is presented. Textual and non-textual features are typically characterized by distinct statistical properties and the correlations between them are highly non-linear. It is very difficult for a shallow model to discover these correlations and form an informative unified representation. Our motivation of proposing the mDBN is to tackle these problems using an unified representation to enhance the classification performance.

#### 3.1 The Restricted Boltzmann Machines

The basic building block of our feature leaning component is the Restricted Boltzmann Machine (RBM). The classical RBM is a two-layer undirected graphical model with stochastic visible units  $\mathbf{v}$  and stochastic hidden units  $\mathbf{h}$ . The visible layer and the hidden layer are fully connected to the units in the other layer by a symmetric matrix  $\mathbf{w}$ . The classical RBM has been used effectively in modeling distributions over binary-value data. As for real-value inputs, the gaussian RBM (Bengio et al., 2007) can be employed. Different from the former, the hypothesis for the visible unit in the gaussian RBM is the normal distribution.

#### 3.2 Feature Learning

The illustration of the feature learning model is given by Figure 2. Basically, the model consists of two parts.

In the bottom part (i.e.,  $V-H_1$ ,  $H_1-H_2$ ), each data modality is modeled by a two-layer DBN separately. For clarity, we take the textual modality as an example to illustrate the construction of the mDBN in this part. Given a textual input vector  $\mathbf{v}$ , the visible layer generates the hidden vector  $\mathbf{h}$ , by

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

Then the conditional distribution of  $\mathbf{v}$  given  $\mathbf{h}$ , is

$$p(v_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j w_{ij}h_j).$$

where  $\sigma(x) = (1 + e^{-x})^{-1}$  denotes the logistic function. The parameters are updated by performing gradient ascent using Contrastive Divergence (CD) algorithm (Hinton, 2002).

After learning the RBMs in the bottom layer, we treat the activation probabilities of its hidden units driven by the inputs, as the training data for training a new layer. The construction procedures for the non-textual modality are similar to the textual one, except that we use the gaussian RBM to model the real-value inputs in the bottom layer.

Finally, we combine the two models by adding an additional layer,  $H_3$ , on the top of them to form the mDBN. The training method is also similar to the bottom's, but the input vector is the concatenation of the mapped textual vector and the mapped non-textual vector.

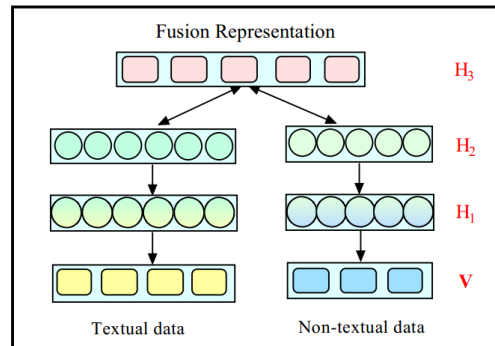


Figure 2: mDBN for Feature Learning

It should be noted in the network, the bottom part is essential to form the joint representation because the correlations between the textual and non-textual features are highly non-linear. It is hard for a RBM directly combining the two disparate sources to learn their correlations.

#### 3.3 Supervised Training and Classification

After the above steps, a deep network for feature learning between textual and non-textual data is established. Classifiers, either support vector machine (SVM) or logistic regression (LR), can then be trained with the unified representation (Ngiam

et al., 2011; Srivastava and Salakhutdinov, 2012). Specifically, the LR classifier is used to make the final prediction in our experiments since it keeps to deliver the best performance.

### 3.4 Basic Features

**Textual Features:** The textual features extract from 1,500 most frequent words in the training dataset after standard preprocessing steps, namely word segmentation, stopwords removal and stemming<sup>1</sup>. As a result, each answer is represented as a vector containing 1,500 distinct terms weighted by binary scheme.

**Non-textual Features:** Referring to the previous work (Jeon et al., 2006; Shah and Pomerantz, 2010), we adopt some features used in theirs and also explore three additional features marked by ‡ sign. The complete list is described in Table 1.

| Features   | Type    |
|--|---------|
| Length of question title (description)           | Integer |
| Length of answer                                 | Integer |
| Number of unique words for the answer ‡          | Integer |
| Ratio of the qa length ‡                         | Float   |
| Answer’s relative position ‡                     | Integer |
| Number of answers for the question               | Integer |
| Number of comments for the question              | Integer |
| Number of questions asked by asker (answerer)    | Integer |
| Number of questions resolved by asker (answerer) | Integer |
| Asker’s (Answerer’s) total points                | Integer |
| Asker’s (Answerer’s) level                       | Integer |
| Asker’s (Answerer’s) total stars                 | Integer |
| Asker’s (Answerer’s) best answer ratio           | Float   |

Table 1: Summary of non-textual features.

## 4 Experiments

### 4.1 Experiment Setup

**Datasets:** We carry out experiments on two datasets. One dataset comes from Baidu Zhidao<sup>2</sup>, which contains 33,740 resolved questions crawled by us from the “travel” category. The other dataset is built by Chen and Nayak (2008) from Yahoo! Answers<sup>3</sup>. We refer to these two datasets as ZHIDAO and YAHOO respectively and randomly sample 10,000 questions from each to form our experimental datasets. According to the user name, we have crawled all the user profile web pages for non-textual feature collection. To alleviate unnecessary noise, we only select those questions with number of answers no less than 3 (one

<sup>1</sup>The stemming step is only used in English corpus.

<sup>2</sup><http://zhidao.baidu.com>

<sup>3</sup><http://answers.yahoo.com>

best answer among them), and those answers at least have 10 tokens. The statistics on the datasets used for experiments are summarized in Table 2.

| Statistics Items          | YAHOO | ZHIDAO |
|---------------------------|-------|--------|
| # of questions            | 6841  | 5368   |
| # of answers              | 74485 | 22435  |
| # of answers per question | 10.9  | 4.1    |
| # of users                | 28812 | 12734  |

Table 2: Statistics on experimental datasets.

**Baselines and Evaluation Metrics:** We compare against the following methods as our baselines. (1) Logistic Regression (LR): We implement the approach used by Shah and Pomerantz (2010) with textual features LR-T, non-textual features LR-N and their simple combination LR-C. (2) DBN: Similar to the mDBN, the outputs of the last hidden layer by the DBN are used as inputs for LR model. Based on the feature sets, we have DBN-T for textual features and DBN-N for non-textual features.

Since we mainly focus on the high quality answers, the *precision*, *recall* and *f1* for positive class and the overall *accuracy* for both classes are employed as our evaluation metrics.

**Model Architecture and Training Details:** To create the mDBN architecture, we use the classical RBM with 1500 visible units followed by 2 hidden layers with 1000 and 800 units respectively for the textual branch, and the gaussian RBM with 20 visible units followed by 2 hidden layers with 100 and 200 units respectively for the non-textual branch. On the joint layer of the network, the layer contains 1000 real-value units.

Each RBM is trained using 1-step CD algorithm. During the training stage, a small weight-cost of 0.0002 is used, and the learning rate for textual data modal is 0.05 while the non-textual data is 0.001. We also adopt a momentum of 0.5 for the first five epochs and 0.9 for the rest epochs. In addition, all non-textual data vectors are normalized to have zero mean and unit standard variance. More details for training the deep architecture can be found in Hinton (2012).

### 4.2 Results and Analysis

In the first experiment, we compare the performance of mDBN with different methods. To make a fair comparison, we use the liblinear toolkit<sup>4</sup> for logistic regression model with L2 regularization and randomly select 70% QA pairs as training data

<sup>4</sup>available at <http://www.csie.ntu.edu.tw/~cjlin/liblinear>

and the rest 30% as testing data. Table 3 and Table 4 summarize the average results of the 5 round experiments on YAHOO and ZHIDAO respectively.

| Methods     | P            | R            | F1           | Accu.        |
|-------------|--------------|--------------|--------------|--------------|
| LR-T        | 0.374        | 0.558        | 0.448        | 0.542        |
| LR-N        | 0.524        | 0.614        | 0.566        | 0.686        |
| LR-C        | 0.493        | 0.557        | 0.523        | 0.662        |
| DBN-T       | 0.496        | 0.571        | 0.531        | 0.663        |
| DBN-N       | 0.505        | 0.578        | 0.539        | 0.670        |
| <b>mDBN</b> | <b>0.534</b> | <b>0.631</b> | <b>0.579</b> | <b>0.694</b> |

Table 3: Comparing results on YAHOO

It is promising to see that the proposed mDBN method notably outperforms almost all the other methods on both datasets over all the metrics as expected, except for the *recall* on ZHIDAO. The main reason for the improvements is that the joint representation learned by mDBN is able to complement each modality perfectly. In addition, the mDBN can extract stronger representation through modeling semantic relationship between textual and non-textual information, which can effectively help distinguish more complicated answers from high quality to low quality.

| Methods     | P            | R            | F1           | Accu.        |
|-------------|--------------|--------------|--------------|--------------|
| LR-T        | 0.380        | 0.540        | 0.446        | 0.553        |
| LR-N        | 0.523        | 0.735        | 0.611        | 0.688        |
| LR-C        | 0.537        | 0.695        | 0.606        | 0.698        |
| DBN-T       | 0.527        | 0.730        | 0.612        | 0.692        |
| DBN-N       | 0.539        | <b>0.760</b> | 0.631        | 0.703        |
| <b>mDBN</b> | <b>0.590</b> | 0.755        | <b>0.662</b> | <b>0.743</b> |

Table 4: Comparing results on ZHIDAO

The classification performance of the textual features are worse on average compared with non-textual features, even when the feature learning strategy is employed. More interestingly, we find the simple combinations of textual and non-textual features don't improve the classification results compared with using non-textual features alone. We conjecture that there are mainly three reasons for the phenomena: First, this is due to the fact that user-generated content is inherently noisy with low word frequency, resulting in the sparsity of employing textual feature. Second, non-textual features (e.g., answer length) usually own strongly statistical properties and feature sparsity problem can be better relieved to some extent. Finally, since correlations between the textual features and non-textual features are highly non-linear, concatenating these features simply sometimes can submerge classification performance. In contrast, mDBN enjoys the advantage of the shared repre-

sentation between textual features and non-textual features using the deep learning architecture.

We also note that neither the mDBN nor other approaches perform very well in predicting answer quality across the two datasets. The best *precision* on ZHIDAO and YAHOO are respectively 59.0% and 53.4%, which means that there are nearly half of the high quality answers not effectively identified. One of the possible reason is that the quality of the corpora influences the result significantly. As shown in Table 2, each question on average receives more than 4 answers on ZHIDAO and more than 10 on YAHOO. Therefore, it is possible that there are several answers with high quality to the same question. Selecting only one as the high quality answer is relatively difficult for our humans, not to mention for the models.

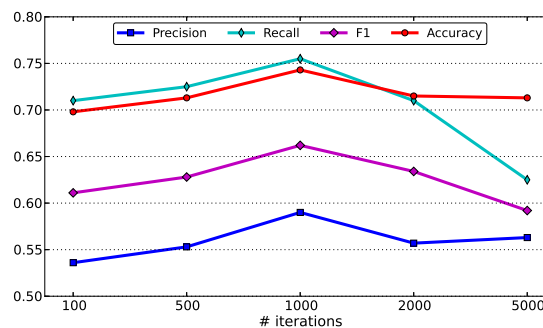


Figure 3: Influences of iterations for mDBN

In the second experiment, we intend to examine the performance of mDBN with different number of iterations. Figure 3 depicts the metrics on ZHIDAO when the iteration number is varied from 100 to 5000. From the result, the first observation is that increasing the number of iterations the performance of mDBN can improve significantly, obtaining the best results for iteration of 1000. This clearly shows the representation power of the mDBN again. However, after a large number of iterations (large than 1000), the mDBN has a detrimental performance. This may be explained by with large number of iterations, the deep learning architecture is easier to be overfitting. The similar trend is also observed on YAHOO.

## 5 Conclusions and Future work

In this paper, we have provided a new perspective to predict the cQA answer quality: learning an informative unified representation between textual and non-textual features instead of concatenating them simply. Specifically, we have proposed a multimodal deep learning framework to

form the unified representation. We compare this with the basic features both in isolation and in combination. Experimental results have demonstrated that our proposed approach can capture the complementarity between textual and non-textual features, which is helpful to improve the performance for cQA answer quality prediction.

For the future work, we plan to explore more semantic analysis to approach the issue for short text quality evaluation. Additionally, more research will be taken to put forward other approaches for learning multimodal representation.

## Acknowledgments

The authors are grateful to the anonymous reviewers for their constructive comments. Special thanks to Chengjie Sun and Deyuan Zhang for insightful suggestions. This work is supported by National Natural Science Foundation of China (NSFC) via grant 61272383 and 61100094.

## References

- E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. 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.
- Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160.
- Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. 2009. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World wide web*, pages 51–60. ACM.
- L. Chen and R. Nayak. 2008. Expertise analysis in a question answer portal for author ranking. In *International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 134–140.
- G.E. Hinton and R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.
- G.E. Hinton, S. Osindero, and Y.W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.
- G.E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.
- G.E. Hinton. 2012. A practical guide to training restricted boltzmann machines. *Lecture Notes in Computer Science*, pages 599–619.
- Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Quality-biased ranking of short texts in microblogging services. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 373–382.
- J. Jeon, W.B. Croft, J.H. Lee, and S. Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 228–235. ACM.
- H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616.
- J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 689–696.
- R. Salakhutdinov and G.E. Hinton. 2009. Deep boltzmann machines. In *Proceedings of the international conference on artificial intelligence and statistics*, volume 5, pages 448–455.
- C. Shah and J. Pomerantz. 2010. Evaluating and predicting answer quality in community qa. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 411–418.
- N. Srivastava and R. Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2231–2239.
- B. Wang, X. Wang, C. Sun, B. Liu, and L. Sun. 2010. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1230–1238. ACL.
- B. Wang, B. Liu, X. Wang, C. Sun, and D. Zhang. 2011. Deep learning approaches to semantic relevance modeling for chinese question-answer pairs. *ACM Transactions on Asian Language Information Processing*, 10(4):21:1–21:16.
- Z.M. Zhou, M. Lan, Z.Y. Niu, and Y. Lu. 2012. Exploiting user profile information for answer ranking in cqa. In *Proceedings of the 21st international conference on World Wide Web*, pages 767–774. ACM.

# Bidirectional Inter-dependencies of Subjective Expressions and Targets and their Value for a Joint Model

Roman Klinger and Philipp Cimiano

Semantic Computing Group

Cognitive Interaction Technology – Center of Excellence (CIT-EC)

Bielefeld University

33615 Bielefeld, Germany

{rklinger, cimiano}@cit-ec.uni-bielefeld.de

## Abstract

Opinion mining is often regarded as a classification or segmentation task, involving the prediction of i) subjective expressions, ii) their target and iii) their polarity. Intuitively, these three variables are bidirectionally interdependent, but most work has either attempted to predict them in isolation or proposing pipeline-based approaches that cannot model the bidirectional interaction between these variables. Towards better understanding the interaction between these variables, we propose a model that allows for analyzing the relation of target and subjective phrases in both directions, thus providing an upper bound for the impact of a joint model in comparison to a pipeline model. We report results on two public datasets (cameras and cars), showing that our model outperforms state-of-the-art models, as well as on a new dataset consisting of Twitter posts.

## 1 Introduction

Sentiment analysis or opinion mining is the task of identifying subjective statements about products, their polarity (*e.g.* positive, negative or neutral) in addition to the particular aspect or feature of the entity that is under discussion, *i.e.*, the so-called *target*. Opinion analysis is thus typically approached as a classification (Täckström and McDonald, 2011; Sayeed et al., 2012; Pang and Lee, 2004) or segmentation (Choi et al., 2010; Johansson and Moschitti, 2011; Yang and Cardie, 2012) task by which fragments of the input are classified or labelled as representing a subjective phrase (Yang and Cardie, 2012), a polarity or a target (Hu and Liu, 2004; Li et al., 2010; Popescu and Etzioni, 2005; Jakob and Gurevych, 2010). As an example, the sentence “I like the low weight of the camera.”

contains a subjective term “like”, and the target “low weight”, which can be classified as a positive statement.

While the three key variables (subjective phrase, polarity and target) intuitively influence each other bidirectionally, most work in the area of opinion mining has concentrated on either predicting one of these variables in isolation (*e.g.* subjective expressions by Yang and Cardie (2012)) or modeling the dependencies uni-directionally in a pipeline architecture, *e.g.* predicting targets on the basis of perfect and complete knowledge about subjective terms (Jakob and Gurevych, 2010). However, such pipeline models do not allow for inclusion of bidirectional interactions between the key variables. In this paper, we propose a model that can include bidirectional dependencies, attempting to answer the following questions which so far have not been addressed but provide the basis for a joint model:

- What is the impact of the performance loss of a non-perfect subjective term extraction in comparison to perfect knowledge?
- Further, how does perfect knowledge about targets influence the prediction of subjective terms?
- How is the latter affected if the knowledge about targets is imperfect, *i.e.* predicted by a learned model?

We study these questions using imperatively defined factor graphs (IDFs, McCallum et al. (2008), McCallum et al. (2009)) to show how these bidirectional dependencies can be modeled in an architecture which allows for further steps towards joint inference. IDFs are a convenient way to define probabilistic graphical models that make structured predictions based on complex dependencies.



## 2 A Model for the Extraction of Target Phrases and Subjective Expressions

This section gives a brief introduction to imperatively defined factor graphs and then introduces our model.

### 2.1 Imperatively Defined Factor Graphs

A factor graph (Kschischang et al., 2001) is a bipartite graph over factors and variables. Let factor graph  $G$  define a probability distribution over a set of output variables  $\mathbf{y}$  conditioned on input variables  $\mathbf{x}$ . A factor  $\Psi_i$  computes a scalar value over the subset of variables  $\mathbf{x}_i$  and  $\mathbf{y}_i$  that are neighbors of  $\Psi_i$  in the graph. Often this real-valued function is defined as the exponential of an inner product over sufficient statistics  $\{f_{ik}(\mathbf{x}_i, \mathbf{y}_i)\}$  and parameters  $\{\theta_{ik}\}$ , where  $k \in [1, K_i]$  and  $K_i$  is the number of parameters for factor  $\Psi_i$ .

A *factor template*  $T_j$  consists of parameters  $\{\theta_{jk}\}$ , sufficient statistic functions  $\{f_{jk}\}$ , and a description of an arbitrary relationship between variables, yielding a set of tuples  $\{(\mathbf{x}_j, \mathbf{y}_j)\}$ . For each of these tuples, the factor template instantiates a factor that shares  $\{\theta_{jk}\}$  and  $\{f_{jk}\}$  with all other instantiations of  $T_j$ . Let  $\mathcal{T}$  be the set of factor templates and  $Z(\mathbf{x})$  be the partition function for normalization. The probability distribution can then be written as  $p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}$

$$\prod_{T_j \in \mathcal{T}} \prod_{(\mathbf{x}_i, \mathbf{y}_i) \in T_j} \exp\left(\sum_{k=1}^{K_j} \theta_{jk} f_{jk}(\mathbf{x}_i, \mathbf{y}_i)\right).$$

FACTORIE<sup>1</sup> (McCallum et al., 2008; McCallum et al., 2009) is an implementation of imperatively defined factor graphs in the context of Markov

<sup>1</sup><http://factorie.cs.umass.edu>

|             | subjective   | target   |
|-------------|--|--|
|             | better   | than CCD shift systems   |
| single span | POS=JJR<br>W=better<br>POS-W=better_JJR  | POS=NN<br>W=shift<br>W=systems<br>POS-W=shift_NN<br>POS-W=systems_NNS<br>POS-SEQ=NN-NNS  |
| inter span  | ONE-EDGE-POS=JJR<br>ONE-EDGE-W=better<br>ONE-EDGE-POS-W=better_JJR<br>ONE-EDGE-POS-SEQ=JJR<br>BOTH-POS=JJR<br>BOTH-W=better<br>BOTH-POS-W=better_JJR<br>BOTH-POS-POS-SEQ=JJR | NO-CLOSE-NOUN<br>ONE-EDGE-POS=NN<br>ONE-EDGE-POS=NNS<br>ONE-EDGE-W=shift<br>ONE-EDGE-W=sensors<br>BOTH-POS=NN<br>BOTH-POS=NNS<br>... |

Figure 1: Example for features extracted for target and subjective expressions (text snippet taken from the camera data set (Kessler et al., 2010)). IOB-like features are merged for simplicity in this depiction.

chain Monte Carlo (MCMC) inference, a common approach for inference in very large graph structures (Culotta and McCallum, 2006; Richardson and Domingos, 2006; Milch et al., 2006). The term imperative is used to denote that actual code in an imperative programming language is written to describe templates and the relationship of tuples they yield. This flexibility is beneficial for modeling inter-dependencies as well as designing information flow in joint models.

### 2.2 Model

Our model is similar to a semi-Markov conditional random field (Sarawagi and Cohen, 2004). It predicts the offsets for target mentions and subjective phrases and can use the information of each other during inference. In contrast to a linear chain conditional random field (Lafferty et al., 2001), this allows for taking distant dependencies of unobserved variables into account and simplifies the design of features measuring characteristics of multi-token phrases. The relevant variables, *i.e.* target and subjective phrase, are modelled via complex span variables of the form  $s = (l, r, c)$  with a left and right offset  $l$  and  $r$ , and a class  $c \in \{\text{target}, \text{subjective}\}$ . These offsets denote the span on a token sequence  $\mathbf{t} = (t_1, \dots, t_n)$ .

We use two different templates to define factors between variables: a *single span* template and an *inter-span* template. The *single span* template defines factors with scores based on features of the tokens in the span and its vicinity. In our model, all features are boolean. As token-based features we use the POS tag, the lower-case representation of the token as well as both in combination. The actual span representation consists of these features prefixed with “I” for all tokens in the span, with “B” for the token at the beginning of the span, and with “E” for the token at the end of the span. In addition, the sequence of POS tags of all tokens in the span is included as a feature.

The inter-span template takes three characteristics of spans into account: Firstly, we measure if a potential target span contains a noun which is the closest noun to a subjective expression. Secondly, we measure for each span if a span of the other class is in the same sentence. A third feature indicates whether there is only one edge in the dependency graph between the tokens contained in spans of a different class. These features are to a great extent inspired by Jakob and Gurevych

(2010). For parsing, we use the Stanford parser (Klein and Manning, 2003).

The features described so far, however, cannot differentiate between a possible aspect mention which is a target of a subjective expression and one which is not. Therefore, the features of the inter-span template are actually built by taking the cross-product of the three described characteristics with all single-span features. Spans which are not in the context of a span of a different class are represented by a ‘negated’ feature (namely No-Close-Noun, No-Single-Edge, and Not-Both-In-Sentence). The example in Figure 1 shows features for two spans which are in context of each other. All of these features representing the text are taken into account for each class, *i. e.*, target and subjective expression.

Inference is performed via Markov Chain Monte Carlo (MCMC) sampling. In each sampling step, only the variables which actually change need to be evaluated, and therefore the sampler directs the process of unrolling the templates to factors. These world changes are necessary to find the maximum a posteriori (MAP) configuration as well as learning the parameters of the model. For each token in the sequence, a span of length one of each class is proposed if no span containing the token exists. For each existing span, it is proposed to change its label, shorten or extend it by one token if possible (all at the beginning and at the end of the span, respectively). Finally, a span can be removed completely.

In order to learn the parameters of our model, we apply SampleRank (Wick et al., 2011). A crucial component in the framework is the objective function which gives feedback about the quality of a sample proposal during training. We use the following objective function  $f(\mathbf{t})$  to evaluate a proposed span  $\mathbf{t}$ :

$$f(\mathbf{t}) = \max_{\mathbf{g} \in \mathbf{s}} \frac{o(\mathbf{t}, \mathbf{g})}{|\mathbf{g}|} - \alpha \cdot p(\mathbf{t}, \mathbf{g}),$$

where  $\mathbf{s}$  is the set of all spans in the gold standard. Further, the function  $o$  calculates the overlap in terms of tokens of two spans and the function  $p$  returns the number of tokens in  $\mathbf{t}$  that are not contained in  $\mathbf{g}$ , *i. e.*, those which are outside the overlap (both functions taking into account the class of the span). Thus, the first part of the objective function represents the fraction of correctly proposed contiguous tokens, while the second part penalizes a

span for containing too many tokens that are outside the best span. Here,  $\alpha$  is a parameter which controls the penalty.

## 3 Results and Discussion

### 3.1 Experimental Setting

We report results on the J.D. Power and Associates Sentiment Corpora<sup>2</sup>, an annotated data set of blog posts in the car and in the camera domain (Kessler et al., 2010). From the rich annotation set, we use subjective terms and entity mentions which are in relation to them as targets. We do not consider *comitter*, *negator*, *neutralizer*, *comparison*, *opo*, or *descriptor* annotations to be subjective expressions. Results on these data sets are compared to Jakob and Gurevych (2010).

In addition, we report results on a Twitter data set<sup>3</sup> for the first time (Spina et al., 2012). Here, we use a Twitter-specific tokenizer and POS tagger<sup>4</sup> (Owoputi et al., 2013) instead of the Stanford parser. Hence, the single-edge-based feature described in Section 2.2 is not used for this dataset. A short summary of the datasets is given in Table 1.

As evaluation metric we use the  $F_1$  measure, the harmonic mean between precision and recall. True positive spans are evaluated in a perfect match and approximate match mode, where the latter regards a span as positive if one token within it is included in a corresponding span in the gold standard. In this case, other predicted spans matching *the same* gold span do not count as false positives. In the objective function,  $\alpha$  is set to 0.01 to prefer spans which are longer than the gold phrase over predicting no span.

Four different experiments are performed (all via 10-fold cross validation): First, predicting subjectivity expressions followed by predicting targets while making use of the previous prediction. Sec-

<sup>2</sup><http://verbs.colorado.edu/jdpacorporus/>

<sup>3</sup><http://nlp.uned.es/~damiano/datasets/entityProfiling ORM Twitter.html>

<sup>4</sup>In version 0.3, <http://www.ark.cs.cmu.edu/TweetNLP/>

|             | Car   | Camera | Twitter |
|-------------|-------|--------|---------|
| Texts       | 457   | 178    | 9238    |
| Targets     | 11966 | 4516   | 1418    |
| Subjectives | 15056 | 5128   | 1519    |

Table 1: Statistics of the data sets.

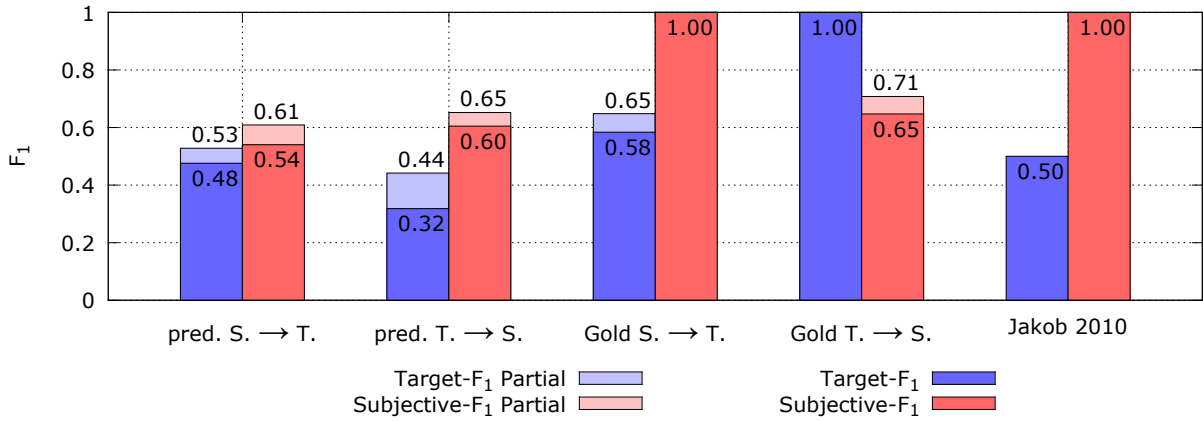


Figure 2: Results for the workflow of first predicting subjective phrases, then targets (pred. S. → T.), and vice versa (pred. T. → S.), as well as in comparison to having perfect information for the first step for the camera data set.

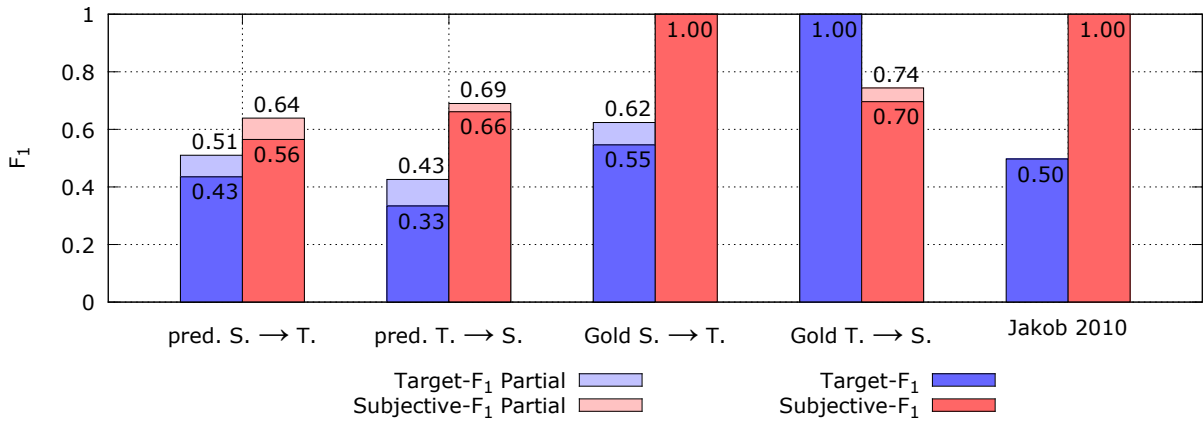


Figure 3: Results for the car data set.

ond, predicting targets followed by predicting subjective expressions. Third, assuming perfect knowledge of subjective expressions when predicting targets, and fourth, assuming perfect knowledge of targets in predicting subjective expressions. This provides us with the information how good a prediction can be with perfect knowledge of the other variable as well as an estimate of how good the prediction can be without any previous knowledge.

### 3.2 Results

Figures 2, 3 and 4 show the results for the four different settings compared to the results by Jakob and Gurevych (2010) for cars and cameras. The darker bars correspond to perfect match, the lighter ones to the increase when taking partial matches into account. In the following we only discuss the perfect match.

Comparing the results (for the car and camera

data sets, Figure 2 and 3) for subjectivity prediction, one can observe a limited performance when targets are not known (0.54  $F_1$  for the camera set, 0.56  $F_1$  for the car set), an upper bound with perfect target information is much higher (0.65  $F_1$ , 0.7  $F_1$ ). When first predicting targets followed by subjective term prediction, we obtain results of 0.6  $F_1$  and 0.66  $F_1$ . The results for target prediction are much lower when not knowing subjective expressions in advance (0.32  $F_1$ , 0.33  $F_1$ ), and clearly increase with predicted subjective expressions (0.48  $F_1$ , 0.43  $F_1$ ) and outperform previous results when compared to Jakob and Gurevych (2010) (0.58  $F_1$ , 0.55  $F_1$  in comparison to their 0.5  $F_1$  on both sets).

The results for the Twitter data set show the same characteristics (in Figure 4). However, they are generally much lower. In addition, the difference between exact and partial match evaluation modes

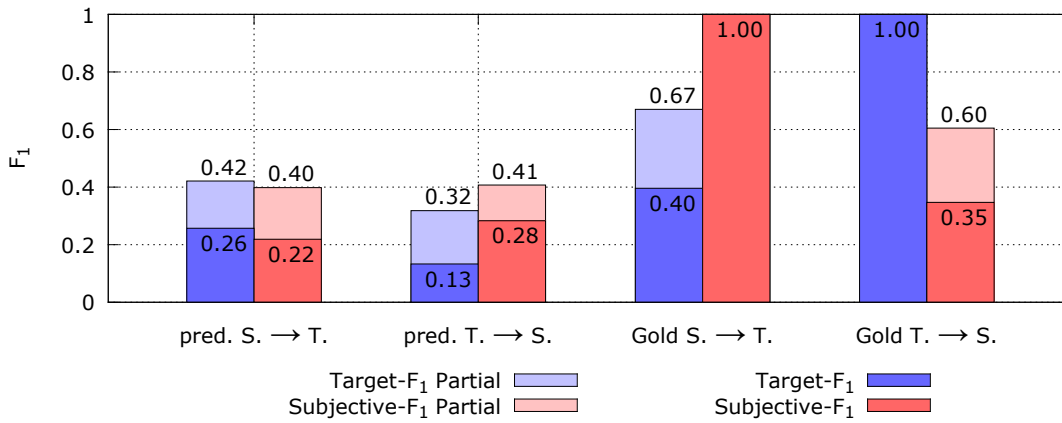


Figure 4: Results for the Twitter data set.

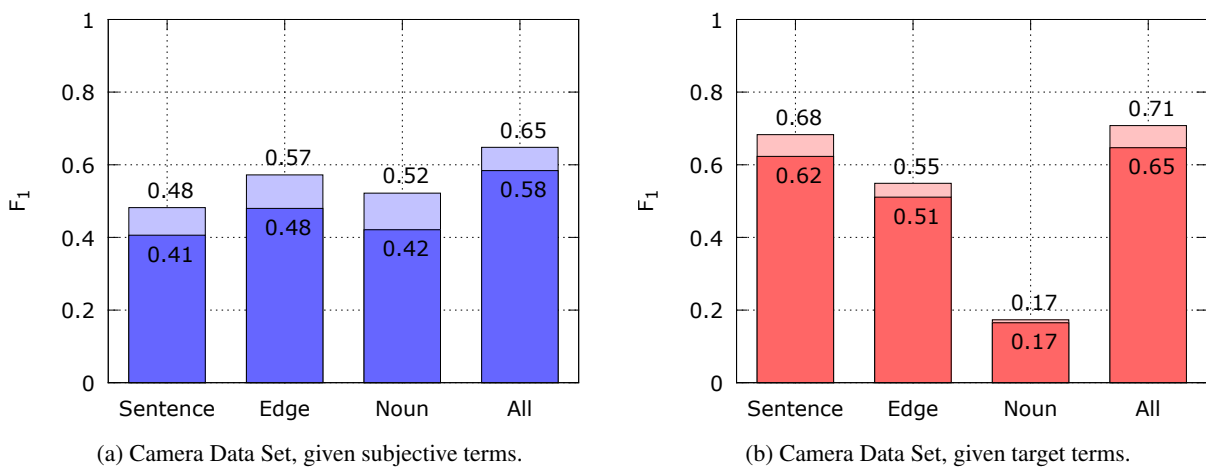


Figure 5: Evaluation of the impact of different features.

is higher. This is due to the existence of many more phrases spanning multiple tokens.

Exemplarily, the impact of the three features in the inter-span templates for the camera data set is depicted in Figure 5 for (a) given subjective terms (b) given targets, respectively. Detecting the closest noun is mainly of importance for target identification and only to a minor extent for detecting subjective phrases. A short path in the dependency graph and detecting if both phrases are in the same sentence have a high positive impact for both subjective and target phrases.

### 3.3 Conclusion and Discussion

The experiments in this paper show that target phrases and subjective terms are clearly interdependent. However, the impact of knowledge about one type of entity for the prediction of the other type of entity has been shown to be asymmetric. The results clearly suggest that the impact of sub-

jective terms on target terms is higher than the other way round. Therefore, if a pipeline architecture is chosen, this order is to be preferred. However, the results with perfect knowledge of the counterpart entity show (in both directions) that the entities influence each other positively. Therefore, the challenge of extracting subjective expressions and their targets is a great candidate for applying supervised, joint inference.

### Acknowledgments

Roman Klinger has been funded by the “It’s OWL” project (“Intelligent Technical Systems Ostwestfalen-Lippe”, <http://www.its-owl.de/>), a leading-edge cluster of the German Ministry of Education and Research. We thank the information extraction and synthesis laboratory (IESL) at the University of Massachusetts Amherst for their support.

## References

- Yoonjung Choi, Seongchan Kim, and Sung-Hyon Myaeng. 2010. Detecting Opinions and their Opinion Targets in NTCIR-8. *Proceedings of NTCIR8 Workshop Meeting*, pages 249–254.
- A. Culotta and A. McCallum. 2006. Tractable Learning and Inference with High-Order Representations. In *ICML Workshop on Open Problems in Statistical Relational Learning*.
- 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, New York, NY, USA. ACM.
- Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Richard Johansson and Alessandro Moschitti. 2011. Extracting opinion expressions and their polarities: exploration of pipelines and joint models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*, pages 101–106, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jason S. Kessler, Miriam Eckert, Lyndsie Clark, and Nicolas Nicolov. 2010. The 2010 ICWSM JDP A Sentiment Corpus for the Automotive Domain. In *4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*.
- D. Klein and Ch. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems]*.
- F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. 2001. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Trans on Information Theory*, 47(2):498–519.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289.
- Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Sentiment analysis with global topics and local dependency. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1371–1376, Atlanta, Georgia, USA.
- A. McCallum, K. Rohanimanesh, M. Wick, K. Schultz, and Sameer Singh. 2008. FACTORIE: Efficient Probabilistic Programming via Imperative Declarations of Structure, Inference and Learning. In *NIPS Workshop on Probabilistic Programming*.
- Andrew McCallum, Karl Schultz, and Sameer Singh. 2009. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*.
- B. Milch, B. Marthi, and S. Russell. 2006. *BLOG: Relational Modeling with Unknown Objects*. Ph.D. thesis, University of California, Berkeley.
- O. Owoputi, B. OConnor, Ch. Dyer, K. Gimpely, N. Schneider, and N. A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, pages 271–278, Barcelona, Spain, July.
- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.
- Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems]*.
- Asad Sayeed, Jordan Boyd-Graber, Bryan Rusk, and Amy Weinberg. 2012. Grammatical structures for word-level sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 667–676, Montréal, Canada, June. Association for Computational Linguistics.
- D. Spina, E. Meij, A. Oghina, M. T. Bui, M. Breuss, and M. de Rijke. 2012. A Corpus for Entity Profiling in Microblog Posts. In *LREC Workshop on Information Access Technologies for Online Reputation Management*.
- Oscar Täckström and Ryan McDonald. 2011. Semi-supervised latent variable models for sentence-level sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages

569–574, Portland, Oregon, USA, June. Association for Computational Linguistics.

M. Wick, K. Rohanimanesh, K. Bellare, A. Culotta, and A. McCallum. 2011. SampleRank: Training factor graphs with atomic gradients. In *International Conference on Machine Learning*.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345, Stroudsburg, PA, USA. Association for Computational Linguistics.

# Identifying Sentiment Words Using an Optimization-based Model without Seed Words

Hongliang Yu<sup>1</sup>, Zhi-Hong Deng<sup>2\*</sup>, Shiyinxue Li<sup>3</sup>

Key Laboratory of Machine Perception (Ministry of Education),  
School of Electronics Engineering and Computer Science,  
Peking University, Beijing 100871, China

<sup>1</sup>yuhongliang324@gmail.com

<sup>2</sup>zhdeng@cis.pku.edu.cn

<sup>3</sup>rachellieinspace@gmail.com

## Abstract

Sentiment Word Identification (SWI) is a basic technique in many sentiment analysis applications. Most existing researches exploit seed words, and lead to low robustness. In this paper, we propose a novel optimization-based model for SWI. Unlike previous approaches, our model exploits the sentiment labels of documents instead of seed words. Several experiments on real datasets show that WEED is effective and outperforms the state-of-the-art methods with seed words.

## 1 Introduction

In recent years, sentiment analysis (Pang et al., 2002) has become a hotspot in opinion mining and attracted much attention. Sentiment analysis is to classify a text span into different sentiment polarities, i.e. positive, negative or neutral. Sentiment Word Identification (SWI) is a basic technique in sentiment analysis. According to (Ku et al., 2006)(Chen et al., 2012)(Fan et al., 2011), SWI can be applied to many fields, such as determining critics opinions about a given product, tweeter classification, summarization of reviews, and message filtering, etc. Thus in this paper, we focus on SWI.

Here is a simple example of how SWI is applied to comment analysis. The sentence below is an movie review in IMDB database:

- **Bored** performers and a **lackluster** plot and script, do not make a **good** action movie.

In order to judge the sentence polarity (thus we can learn about the preference of this user), one must recognize which words are able to express sentiment. In this sentence, “bored” and “lackluster” are negative while “good” should be positive, yet

its polarity is reversed by “not”. By such analysis, we then conclude such movie review is a negative comment. But how do we recognize sentiment words?

To achieve this, previous supervised approaches need labeled polarity words, also called **seed words**, usually manually selected. The words to be classified by their sentiment polarities are called **candidate words**. Prior works study the relations between labeled seed words and unlabeled candidate words, and then obtain sentiment polarities of candidate words by these relations. There are many ways to generate word relations. The authors of (Turney and Littman, 2003) and (Kaji and Kitsuregawa, 2007) use statistical measures, such as point wise mutual information (PMI), to compute similarities in words or phrases. Kanayama and Nasukawa (2006) assume sentiment words successively appear in the text, so one could find sentiment words in the context of seed words (Kanayama and Nasukawa, 2006). In (Hassan and Radev, 2010) and (Hassan et al., 2011), a Markov random walk model is applied to a large word relatedness graph, constructed according to the synonyms and hypernyms in WordNet (Miller, 1995).

However, approaches based on seed words has obvious shortcomings. First, polarities of seed words are not reliable for various domains. As a simple example, “rise” is a neutral word most often, but becomes positive in stock market. Second, manually selection of seed words can be very subjective even if the application domain is determined. Third, algorithms using seed words have low robustness. Any missing key word in the set of seed words could lead to poor performance. Therefore, the seed word set of such algorithms demands high completeness (by containing common polarity words as many as possible).

Unlike the previous research work, we identify sentiment words without any seed words in this paper. Instead, the documents’ bag-of-words in-

\*Corresponding author

formation and their polarity labels are exploited in the identification process. Intuitively, polarities of the document and its most component sentiment words are the same. We call such phenomenon as “sentiment matching”. Moreover, if a word is found mostly in positive documents, it is very likely a positive word, and vice versa.

We present an optimization-based model, called WEED, to exploit the phenomenon of “sentiment matching”. We first measure the importance of the component words in the labeled documents semantically. Here, the basic assumption is that important words are more sentiment related to the document than those less important. Then, we estimate the polarity of each document using its component words’ importance along with their sentiment values, and compare the estimation to the real polarity. After that, we construct an optimization model for the whole corpus to weigh the overall estimation error, which is minimized by the best sentiment values of candidate words. Finally, several experiments demonstrate the effectiveness of our approach. To the best of our knowledge, this paper is the first work that identifies sentiment words without seed words.

## 2 The Proposed Approach

### 2.1 Preliminary

We formulate the sentiment word identification problem as follows. Let  $D = \{d_1, \dots, d_n\}$  denote

document set. Vector  $\vec{l} = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}$  represents their

labels. If document  $d_i$  is a positive sample, then  $l_i = 1$ ; if  $d_i$  is negative, then  $l_i = -1$ . We use the notation  $C = \{c_1, \dots, c_V\}$  to represent candidate word set, and  $V$  is the number of candidate words. Each document is formed by consecutive words in  $C$ . Our task is to predict the sentiment polarity of each word  $c_j \in C$ .

### 2.2 Word Importance

We assume each document  $d_i \in D$  is presented

by a bag-of-words feature vector  $\vec{f}_i = \begin{bmatrix} f_{i1} \\ \vdots \\ f_{iV} \end{bmatrix}$ ,

where  $f_{ij}$  describes the importance of  $c_j$  to  $d_i$ . A high value of  $f_{ij}$  indicates word  $c_j$  contributes a lot to document  $d_i$  in semantic view, and vice versa. Note that  $f_{ij} > 0$  if  $c_j$  appears in  $d_i$ , while

$f_{ij} = 0$  if not. For simplicity, every  $\vec{f}_i$  is normalized to a unit vector, such that features of different documents are relatively comparable.

There are several ways to define the word importance, and we choose normalized *TF-IDF* (Jones, 1972). Therefore, we have  $f_{ij} \propto TF-IDF(d_i, c_j)$ , and  $\|\vec{f}_i\| = 1$ .

### 2.3 Polarity Value

In the above description, the sentiment polarity has only two states, positive or negative. We extend both word and document polarities to polarity values in this section.

**Definition 1** *Word Polarity Value:* For each word  $c_j \in C$ , we denote its **word polarity value** as  $w(c_j)$ .  $w(c_j) > 0$  indicates  $c_j$  is a positive word, while  $w(c_j) < 0$  indicates  $c_j$  is a negative word.  $|w(c_j)|$  indicates the strength of the belief of  $c_j$ ’s polarity. Denote  $w(c_j)$  as  $w_j$ , and the word polar-

ity value vector  $\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_V \end{bmatrix}$ .

For example, if  $w(\text{“bad”}) < w(\text{“greedy”}) < 0$ , we can say “bad” is more likely to be a negative word than “greedy”.

**Definition 2** *Document Polarity Value:* For each document  $d_i$ , **document polarity value** is

$$y(d_i) = \text{cosine}(\vec{f}_i, \vec{w}) = \frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|}. \quad (1)$$

We denote  $y(d_i)$  as  $y_i$  for short.

Here, we can regard  $y_i$  as a polarity estimate for  $d_i$  based on  $\vec{w}$ . To explain this, Table 1 shows an example. “MR1”, “MR2” and “MR3” are three movie review documents, and “compelling” and “boring” are polarity words in the vocabulary. we simply use TF to construct the document feature vectors without normalization. In the table, these three vectors,  $\vec{f}_1$ ,  $\vec{f}_2$  and  $\vec{f}_3$ , are (3, 1), (2, 1) and (1, 3) respectively. Similarly, we can get  $\vec{w} = (1, -1)$ , indicating “compelling” is a positive word while “boring” is negative. After normalizing  $\vec{f}_1$ ,  $\vec{f}_2$  and  $\vec{f}_3$ , and calculating their cosine similarities with  $\vec{w}$ , we obtain  $y_1 > y_2 > 0 > y_3$ . These inequalities tell us the first two reviews are positive, while the last review is negative. Furthermore, we believe that “MR1” is more positive than “MR2”.



|     |              |          |
|-----|--------------|----------|
|     | “compelling” | “boring” |
| MR1 | 3            | 1        |
| MR2 | 2            | 1        |
| MR3 | 1            | 3        |
| $w$ | 1            | -1       |

Table 1: Three rows in the middle shows the feature vectors of three movie reviews, and the last row shows the word polarity value vector  $\vec{w}$ . For simplicity, we use TF value to represent the word importance feature.

## 2.4 Optimization Model

As mentioned above, we can regard  $y_i$  as a polarity estimate for document  $d_i$ . A precise prediction makes the positive document’s estimator close to 1, and the negative’s close to -1. We define the polarity estimate error for document  $d_i$  as:

$$e_i = |y_i - l_i| = \left| \frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|} - l_i \right|. \quad (2)$$

Our learning procedure tries to decrease  $e_i$ . We obtain  $\vec{w}$  by minimizing the overall estimation error of all document samples  $\sum_{i=1}^n e_i^2$ . Thus, the optimization problem can be described as

$$\min_{\vec{w}} \sum_{i=1}^n \left( \frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|} - l_i \right)^2. \quad (3)$$

After solving this problem, we not only obtain the polarity of each word  $c_j$  according to the sign of  $w_j$ , but also its polarity belief based on  $|w_j|$ .

## 2.5 Model Solution

We use normalized vector  $\vec{x}$  to substitute  $\frac{\vec{w}}{\|\vec{w}\|}$ , and derive an equivalent optimization problem:

$$\begin{aligned} \min_{\vec{x}} \quad & E(\vec{x}) = \sum_{i=1}^n (\vec{f}_i^T \cdot \vec{x} - l_i)^2 \\ \text{s.t.} \quad & \|\vec{x}\| = 1. \end{aligned} \quad (4)$$

The equality constraint of above model makes the problem non-convex. We relax the equality constraint to  $\|\vec{x}\| \leq 1$ , then the problem becomes convex. We can rewrite the objective function as the form of least square regression:  $E(\vec{x}) = \|\mathbf{F} \cdot \vec{x} - \vec{l}\|^2$ , where  $\mathbf{F}$  is the feature matrix, and

$$\text{equals to } \begin{bmatrix} \vec{f}_1^T \\ \vdots \\ \vec{f}_n^T \end{bmatrix}.$$

Now we can solve the problem by convex optimization algorithms (Boyd and Vandenberghe, 2004), such as gradient descend method. In each iteration step, we update  $\vec{x}$  by  $\Delta \vec{x} = \eta \cdot (-\nabla E) = 2\eta \cdot (\mathbf{F}^T \vec{l} - \mathbf{F}^T \mathbf{F} \vec{x})$ , where  $\eta > 0$  is the learning rate.

## 3 Experiment

### 3.1 Experimental Setup

We leverage two widely used document datasets. The first dataset is the Cornell Movie Review Data<sup>1</sup>, containing 1,000 positive and 1,000 negative processed reviews. The other is the Stanford Large Dataset<sup>2</sup> (Maas et al., 2011), a collection of 50,000 comments from IMDB, evenly divided into training and test sets.

The ground-truth is generated with the help of a sentiment lexicon, MPQA subjective lexicon<sup>3</sup>. We randomly select 20% polarity words as the seed words, and the remaining are candidate ones. Here, the seed words are provided for the baseline methods but not for ours. In order to increase the difficulty of our task, several non-polarity words are added to the candidate word set. Table 2 shows the word distribution of two datasets.

| Dataset  | Word Set  | pos | neg  | non  | total |
|----------|-----------|-----|------|------|-------|
| Cornell  | seed      | 135 | 201  | -    | 336   |
|          | candidate | 541 | 806  | 1232 | 2579  |
| Stanford | seed      | 202 | 343  | -    | 545   |
|          | candidate | 808 | 1370 | 2566 | 4744  |

Table 2: Word Distribution

In order to demonstrate the effectiveness of our model, we select two baselines, *SO-PMI* (Turney and Littman, 2003) and *COM* (Chen et al., 2012). Both of them need seed words.

### 3.2 Top-K Test

In face of the long lists of recommended polarity words, people are only concerned about the top-ranked words with the highest sentiment value. In this experiment we consider the accuracy of the top  $K$  polarity words. The quality of a polarity word list is measured by  $p@K = \frac{N_{right,K}}{K}$ , where  $N_{right,K}$  is the number of top- $K$  words which are correctly recommended.

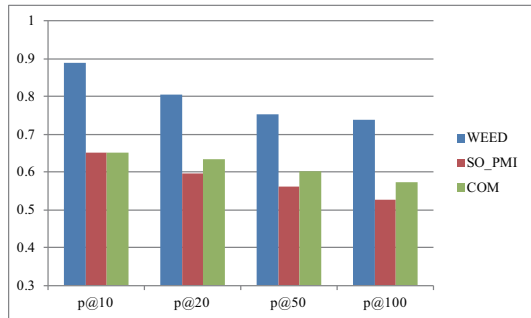
<sup>1</sup><http://www.cs.cornell.edu/people/pabo/movie-review-data/>

<sup>2</sup><http://ai.stanford.edu/amaas/data/sentiment/>

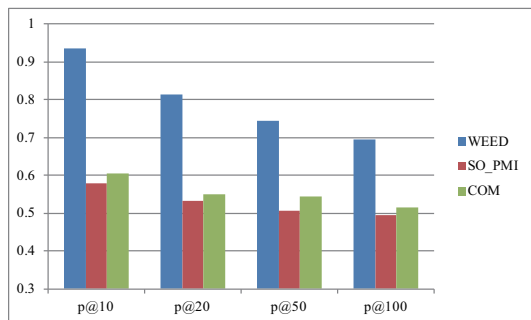
<sup>3</sup><http://www.cs.pitt.edu/mpqa/>

| WEED                         |                          | SO-PMI                        |                            | COM                |                          |
|------------------------------|--------------------------|-------------------------------|----------------------------|--------------------|--------------------------|
| positive words               | negative words           | positive words                | negative words             | positive words     | negative words           |
| <b>great excellent</b>       | <b>bad stupid</b>        | destiny lush                  | cheap worst                | <b>best great</b>  | <b>ridiculous bad</b>    |
| <b>perfect perfectly</b>     | <b>worst mess</b>        | <b>brilliant skillfully</b>   | <b>ridiculous annoying</b> | will star          | plot evil                |
| <b>terrific best</b>         | <b>boring ridiculous</b> | courtesy courtesy             | <b>damn pathetic</b>       | bad fun            | star garish              |
| <b>true wonderfully</b>      | awful plot               | <b>gorgeous magnificent</b>   | inconsistencies fool       | <b>better</b> plot | <b>dreadfully stupid</b> |
| <b>brilliant outstanding</b> | <b>worse terrible</b>    | temptation <b>marvelously</b> | <b>desperate giddy</b>     | love horror        | pretty fun               |

Table 3: Case Study



(a) Cornell Dataset



(b) Stanford Dataset

Figure 1: Top-K Test

Figure 1 shows the final result of  $p@K$ , which is the average score of the positive and negative list. We can see that in both datasets, our approach highly outperforms two baselines, and the precision is 14.4%-33.0% higher than the best baseline.  $p@10$ s of WEED for Cornell and Stanford datasets reach to 93.5% and 89.0%, and it shows the top 10 words in our recommended list is exceptionally reliable. As the size of  $K$  increases, the accuracy of all methods falls accordingly. This shows three approaches rank the most probable polarity words in the front of the word list. Compared with the small dataset, we obtain a better result with large  $K$  on the Stanford dataset.

### 3.3 Case Study

We conduct an experiment to illustrate the characteristics of three methods. Table 3 shows top-10 positive and negative words for each method,

where the bold words are the ones with correct polarities. From the first two columns, we can see the accuracy of WEED is very high, where positive words are absolutely correct and negative word list makes only one mistake, “plot”. The other columns of this table shows the baseline methods both achieve reasonable results but do not perform as well as WEED.

Our approach is able to identify frequently used sentiment words, which are vital for the applications without prior sentiment lexicons. The sentiment words identified by SO-PMI are not so representative as WEED and COM. For example, “skillfully” and “giddy” are correctly classified but they are not very frequently used. COM tends to assign wrong polarities to the sentiment words although these words are often used. In the 5<sup>th</sup> and 6<sup>th</sup> columns of Table 3, “bad” and “horror” are recognized as positive words, while “pretty” and “fun” are recognized as negative ones. These concrete results show that WEED captures the generality of the sentiment words, and achieves a higher accuracy than the baselines.

## 4 Conclusion and Future Work

We propose an effective optimization-based model, WEED, to identify sentiment words from the corpus without seed words. The algorithm exploits the sentiment information provided by the documents. To the best of our knowledge, this paper is the first work that identifies sentiment words without any seed words. Several experiments on real datasets show that WEED outperforms the state-of-the-art methods with seed words.

Our work can be considered as the first step of building a domain-specific sentiment lexicon. Once some sentiment words are obtained in a certain domain, our future work is to improve WEED by utilizing these words.

## Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Grant No. 61170091).

## References

- S. Boyd and L. Vandenberghe. 2004. *Convex optimization*. Cambridge university press.
- L. Chen, W. Wang, M. Nagarajan, S. Wang, and A.P. Sheth. 2012. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 50–57.
- Wen Fan, Shutao Sun, and Guohui Song. 2011. Probability adjustment naïve bayes algorithm based on nondomain-specific sentiment and evaluation word for domain-transfer sentiment analysis. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, volume 2, pages 1043–1046. IEEE.
- A. Hassan and D. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics.
- A. Hassan, A. Abu-Jbara, R. Jha, and D. Radev. 2011. Identifying the semantic orientation of foreign words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 592–597.
- K.S. Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- N. Kaji and M. Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 1075–1083.
- H. Kanayama and T. Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics.
- Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006 spring symposium on computational approaches to analyzing weblogs*, volume 2001.
- A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational Linguistics (acL-2011)*.
- Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- 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*, pages 79–86. Association for Computational Linguistics.
- P. Turney and M.L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association.

# Detecting Turnarounds in Sentiment Analysis: Thwarting

**Ankit Ramteke**

Dept. of Computer Science & Engg.,  
Indian Institute of Technology  
Bombay, Mumbai, India.  
ankitr@cse.iitb.ac.in

**Akshat Malu**

Dept. of Computer Science & Engg.,  
Indian Institute of Technology  
Bombay, Mumbai, India.  
akshatmalu@cse.iitb.ac.in

**Pushpak Bhattacharyya**

Dept. of Computer Science & Engg.,  
Indian Institute of Technology  
Bombay, Mumbai, India.  
pb@cse.iitb.ac.in

**J. Saketha Nath**

Dept. of Computer Science & Engg.,  
Indian Institute of Technology  
Bombay, Mumbai, India.  
saketh@cse.iitb.ac.in

## Abstract

Thwarting and sarcasm are two uncharted territories in sentiment analysis, the former because of the lack of training corpora and the latter because of the enormous amount of world knowledge it demands. In this paper, we propose a working definition of thwarting amenable to machine learning and create a system that detects if the document is thwarted or not. We focus on identifying thwarting in product reviews, especially in the camera domain. An ontology of the camera domain is created. *Thwarting is looked upon as the phenomenon of polarity reversal at a higher level of ontology compared to the polarity expressed at the lower level.* This notion of thwarting defined with respect to an ontology is novel, to the best of our knowledge. A rule based implementation building upon this idea forms our baseline. We show that machine learning with annotated corpora (thwarted/non-thwarted) is more effective than the rule based system. Because of the skewed distribution of thwarting, we adopt the Area-under-the-Curve measure of performance. To the best of our knowledge, this is the first attempt at the difficult problem of thwarting detection, which we hope will at

least provide a baseline system to compare against.

## 1 Credits

The authors thank the lexicographers at Center for Indian Language Technology (CFILT) at IIT Bombay for their support for this work.

## 2 Introduction

Although much research has been done in the field of sentiment analysis (Liu *et al.*, 2012), *thwarting* and *sarcasm* are not addressed, to the best of our knowledge. Thwarting has been identified as a common phenomenon in sentiment analysis (Pang *et al.*, 2002, Ohana *et al.*, 2009, Brooke, 2009) in various forms of texts but no previous work has proposed a solution to the problem of identifying thwarting. We focus on identifying thwarting in product reviews.

The definition of an opinion as specified in Liu (2012) is

“An opinion is a quintuple,  $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ , where  $e_i$  is the name of an entity,  $a_{ij}$  is an aspect of  $e_i$ ,  $s_{ijkl}$  is the sentiment on aspect  $a_{ij}$  of entity  $e_i$ ,  $h_k$  is the opinion holder, and  $t_l$  is the time when the opinion is expressed by  $h_k$ .”

If the sentiment towards the entity or one of its important attribute contradicts the sentiment towards all other attributes, we can say that the document is thwarted.

A domain ontology is an ontology of various features pertaining to a domain, arranged in a hierarchy. Subsumption in this hierarchy implies that the child is a part or feature of the parent. Domain ontology has been used by various works in NLP (Saggion *et al.*, 2007 and Polpinij *et al.*, 2008). In our work, we use domain ontology of camera. We look upon thwarting as the phenomenon of *reversal of polarity* from the lower level of the ontology to the higher level. At the higher level of ontology the entities mentioned are the whole product or a large critical part of the product. So while statements about entities at the lower level of the ontology are on “details”, statements about entities at higher levels are on the “big picture”. **Polarity reversal from details to the big picture is at the heart of thwarting.**

The motivation for our study on thwarting comes from the fact that: a) Thwarting is a challenging NLP problem and b) Special ML machinery is needed in view of the fact that the training data is so skewed. Additionally large amount of world and domain knowledge maybe called for to solve the problem. In spite of the relatively fewer occurrence of the thwarting phenomenon the problem poses an intellectually stimulating exercise. We may also say that in the limit, thwarting approaches the very difficult problem of sarcasm detection (Tsur *et al.* 2010).

We start by defining and understanding the problem of thwarting in section 2. In section 3, we describe a method to create the domain ontology. In section 4, we propose a naïve rule based approach to detect thwarting. In section 5 we discuss a machine learning based approach which could be used to identify whether a document is thwarted or not. This is followed by experimental results in section 6. Section 7 draws conclusions and points to future work.

### 3 Definition

Thwarting is defined by Pang *et al.*, (2008) as follows:

*“Thwarted expectations basically refer to the phenomenon wherein the author of the text first builds up certain expectations for the topic, only to produce a deliberate contrast to the earlier discussion.”*

For our computational purposes, we define thwarting as:

*“The phenomenon wherein the overall polarity of the document is in contrast with the polarity of majority of the document.”*

This definition emphasizes thwarting as piggy-backing on sentiment analysis to improve the latter’s performance. The current work however only addresses the problem of whether a document is thwarted or not and does not output the sentiment of the document. The basic block diagram for our system is shown in figure 1.

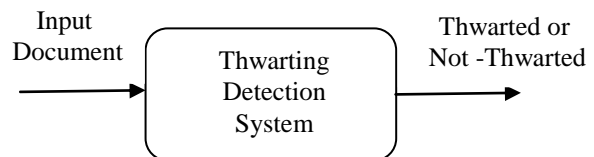


Figure 1: Basic Block Diagram

An example of a thwarted document is:

*“I love the sleek design. The lens is impressive. The pictures look good but, somehow this camera disappoints me. I do not recommend it.”*

While thwarting occurs in various forms of sentiment bearing texts, it is not a very frequent one. It accounts for hardly 1-2% of any given corpus. Thus, it becomes hard to find sufficient number of examples of thwarting to train a classifier.

Since thwarting is a complex natural language phenomenon we require basic NLP tools and resources, whose accuracy in turn can affect the overall performance of a thwarting detection system.

### 4 Building domain ontology

Domain ontology comprises of features and entities from the domain and the relationships between them. The process thus has two steps, *viz.* (a) identify the features and entities, and (b) connect them in the form of a hierarchy. We decided to use a combination of review corpora mining and manual means for identifying key features. Our approach to building the domain ontology is as follows:

**Step 1:** We use Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003) on a corpus containing reviews of a particular product (camera, in our case) to identify key features from the domain. The output is then analyzed manually to finally select the key features. Some additional features get added by human annotator to increase the coverage of the ontology. For Example, in the camera domain, the corpus may include words

like *memory, card, gb, etc.* but, may not contain the word *storage*. The abstract concept of *storage* is contributed by the human annotator through his/her world knowledge.

**Step 2:** The features thus obtained are arranged in the form of a hierarchy by a human annotator.

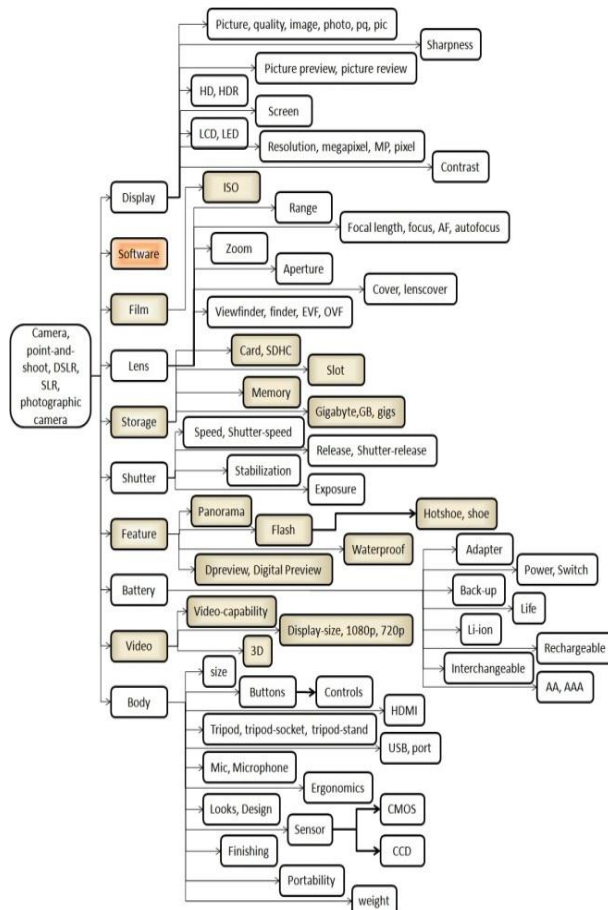


Figure 2: Ontology for the camera domain

## 5 A rule based approach to thwarting recognition

As per the definition of thwarting, most of the thwarted document carries a single sentiment; however, a small but critical portion of the text, carrying the contrary sentiment, actually decides the overall polarity. The critical statement, thus, should be strongly polar (either positive or negative), and it should be on some critical feature of the product.

*From the perspective of the domain ontology, the sentiment towards the overall product or towards some critical feature mentioned near the root of the ontology should be opposite to the sentiment towards features near the leaves.*

Based on these observations we propose the following naïve approach to thwarting detection:

For each sentence in a review to be tested

1. Get the dependency parse of the sentence. This step is essential. It makes explicit the adjective noun dependencies, which in turn uncovers the sentiment on a specific part or feature of the product.

2. Identify the polarities towards all nouns, using the dependency parse and sentiment lexicons.

3. If a domain feature, identified using the domain ontology, exists in the sentence, annotate/update the ontology node, containing the feature, using the polarity obtained.

Once the entire review is processed, we obtain the domain ontology, with polarity marking on nodes, for the corresponding review.

*The given review is thwarted if there is a contradiction of sentiment among different levels of the domain ontology with polarity marking on nodes.*

The sentiment lexicons used are SentiWordNet (Esuli et al., 2006), Taboada (Taboada et al., 2004), BL lexicon (Hu et al., 2004) and Inquirer (Stone et al., 1966).

The procedure is illustrated by an example.

*“I love the sleek design. The lens is impressive. The pictures look good but, somehow this camera disappoints me. I do not recommend it.”*

A part of the ontology, with polarity marking on nodes, for this example is shown in figure 3.

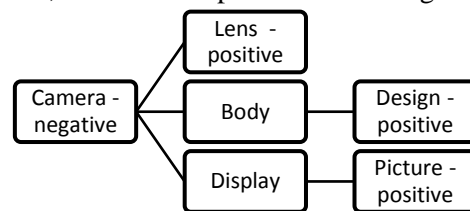


Figure 3: ontology with polarity marking on nodes: example

Based on this ontology we see that there is an opposition of sentiment between the root (“camera”) and the lower nodes. We thus determine that this document is thwarted.

However, since the nodes, within the same level, might have different weighting based upon the product under consideration, this method fails to perform well. For example, the *body* and *video capability* might be subjective whereas any fault in the *lens* or the *battery* will render the camera useless, hence they are more critical. We thus see a need for relative weighting among all features in the ontology.

## 6 A Machine Learning based approach

Manual fixing of relative weightages for the features of the product is possible, but that would be *ad hoc*. We now propose a machine learning based approach to detect thwarting in documents. It uses the domain ontology to identify key features related to the domain. The approach involves two major steps namely learning the weights and building a model that classifies the reviews using the learnt weights.

### 6.1 Learning Weights

The weights are learnt using the loss-regularization framework. The key idea is that the overall polarity of the document is determined by the polarities of individual words in the document. Since, we need to find the weights for the nodes in the domain ontology; we consider only the words belonging to the ontology for further processing. Thus, if  $P$  is the polarity of the review and  $p_i$  is the polarity associated with word  $i$  then  $P = \sum_i w_i p_i$  gives the linear model. The word  $i$  should belong to the ontology as well as the review. Similarly, the hinge loss is given by  $\max(0, 1 - P \cdot w^T x)$  where  $w$  is the weight vector and  $x$  is the feature vector consisting of  $p_i$ 's.

Based on the intuition, that every word contributes some polarity to its parent node in the domain ontology, we also learnt weights on the ontology by percolating polarities towards the root. We experimented with complete percolation, wherein the polarity at a node is its polarity in the document summed with the polarities of all its descendants. We also define controlled percolation, wherein the value added for a particular descendant is a function of its distance from the node. We halved the polarity value percolated, for each edge between the two nodes. Thus, for the example in figure 2, the polarity value of camera would be

$$P_{camera} = p_{camera} + \frac{p_{lens}}{2} + \frac{p_{body}}{2} + \frac{p_{display}}{2} + \frac{p_{design}}{4} + \frac{p_{picture}}{4}$$

Where  $P_{camera}$  is the final polarity for camera and  $p_{word}$  is the polarity of the word  $\epsilon$  {camera, body, display, design, picture}.

### 6.2 Classifier

We use the SVM classifier with features generated using the following steps. We first create a vector of weighted polarity values for each review. This is constructed by generating a value

for each word in the domain ontology encountered while reading the review sequentially. The value is calculated by multiplying the weight, found in the previous step (5.1), with the polarity of the word as determined from the sentence. Since, these vectors will be of different dimensionality for each review, we extract features from these reviews. These features are selected based on our understanding of the problem and the fact that thwarting is a function of the change of polarity values and also the position of change.

#### The Features extracted are:

Document polarity, number of flips of sign (*i.e.* change of polarity from positive to negative and vice versa), the maximum and minimum values in a sequence, the length of the longest contiguous subsequence of positive values (LCSP), the length of the longest contiguous subsequence of negative values (LCSN), the mean of all values, total number of positive values in the sequence, total number of negative values in the sequence, the first and the last value in the sequence, the variance of the moving averages, the difference in the means of LCSP and LCSN.

## 7 Results

Experiments were performed on a dataset obtained by crawling product reviews from Amazon<sup>1</sup>. We focused on the camera domain. We obtained 1196 reviews from this domain. The reviews were annotated for thwarting, *i.e.*, thwarted or non-thwarted as well as polarity. The reviews crawled were given to three different annotators. The instructions given for annotation were as follows:

1. Read the entire review and try to form a mental picture of how sentiment in the document is distributed. Ignore anything that is not the opinion of the writer.
2. Try to determine the overall polarity of the document. The star rating of the document can be used for this purpose.
3. If the overall polarity of the document is negative but, most of the words in the document indicate positive sentiment, or vice versa, then consider the document as thwarted.

Since, identifying thwarting is a difficult task even for humans, we calculated the Cohen's kappa score (Cohen 1960) in order to determine the inter annotator agreement. It was found out to

---

<sup>1</sup>Reviews crawled from <http://www.amazon.com/>

be **0.7317**. The annotators showed high agreement (98%) in the non-thwarted class whereas they agreed on 70% of the thwarted documents.

Out of the 1196 reviews, exactly 21 were thwarted documents, agreed upon by all annotators. We used the Stanford Core NLP tools<sup>2</sup> (Klein *et al.*, 2003, Toutanova *et al.*, 2003) for basic NL processing. The system was tested on the entire dataset.

Since, the data is highly skewed; we used Area under the Curve (AUC) for the ROC curve as the measure of evaluation (Ling *et al.*, 2003). The AUC for a random baseline is expected to be 50%, and the rule based approach is close to the baseline (**56.3%**).

Table 1 shows the results for the experiments with the machine learning model. We used the CVX<sup>3</sup> library in Matlab to solve the optimization problem for learning weights and the LIBSVM<sup>4</sup> library to implement the svm classifier. In order to account for the data skew, we assign a class weight of 50 (determined empirically) to the thwarted instances and 1 for non-thwarted instances in the classifier. All results were obtained using a 10 fold cross validation. The same dataset was used for this set of experiments.

| Loss type for weights | Percolation type for weights | AUC value for classification |
|-----------------------|------------------------------|------------------------------|
| Linear                | Complete                     | 73%                          |
|                       | Controlled                   | <b>81%</b>                   |
| Hinge                 | Complete                     | 70%                          |
|                       | Controlled                   | 76%                          |

Table 1: Results of the machine learning based approach to thwarting detection

We see that the overall system for identification of thwarting performs well for the weights obtained using the linear model with a controlled percolation of polarity values in the ontology. The system outperforms both the random baseline as well as the rule based system. These results though great are to be taken with a pinch of salt. The basic objective for creating a thwarting detection system was to include such a module in the general sentiment analysis framework. Thus, using document polarity as a feature contradicts the objective of sentiment analysis, which is to *find* the document polarity. Without the docu-

ment polarity feature, the values drop by 10% which is not acceptable.

## 8 Conclusions and Future Work

We have described a system for detecting thwarting, based on polarity reversal between opinion on most parts of the product and opinion on the overall product or a critical part of the product. The parts of the product are related to one another through an ontology. This ontology guides a rule based approach to thwarting detection, and also provides features for an SVM based learning system. The ML based system scores over the rule based system. Future work consists in trying out the approach across products and across domains, doing better ontology harnessing from the reviews and investing and searching for distributions and learning algorithms more suitable for the problem.

## References

- Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet allocation. In *the Journal of machine Learning research*, 3, pages 993-1022.
- Brooke, J. 2009. *A Semantic Approach to Automated Text Sentiment Analysis*. Ph.D. thesis, Simon Fraser University.
- Chang, C. C., and Lin, C. J. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.
- Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, no. 1, pages 37-46.
- Esuli, A. and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, Volume 6, pages 417-422.
- Hu, M. and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168-177. ACM.
- Klein, D. and Manning, C. D. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423-430.
- Ling, C. X., Huang, J. and Zhang, H. 2003. AUC: A better measure than accuracy in comparing learning algorithms. In *Advances in Artificial Intelligence*, pages 329-341, Springer Berlin Heidelberg.

<sup>2</sup><http://nlp.stanford.edu/software/corenlp.shtml>

<sup>3</sup><http://cvxr.com/cvx>

<sup>4</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>



- Liu, B., and Zhang, L. 2012. A survey of opinion mining and sentiment analysis. In *Mining Text Data* (pp. 415-463). Springer US.
- Liu B., 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1-167.
- Ohana, B. and Tierney, B. 2009. Sentiment classification of reviews using SentiWordNet. In *9th. IT & T Conference*, page 13.
- Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2), 1-135.
- Pang, B., Lee, L. and Vaithyanathan S. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of EMNLP* pages 79-86).
- Polpinij, J. and Ghose, A. K. 2008. An ontology-based sentiment classification methodology for online consumer reviews. In *Web Intelligence and Intelligent Agent Technology*.
- Taboada, M. and Grieve, J. 2004. Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS# 04# 07)*, Stanford University, CA, pages. 158-161. AAAI Press.
- Toutanova, K., Klein, D., Manning, C. D. and Singer Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pages 252-259.
- Tsur, O., Davidov, D., & Rappoport, A. 2010. ICWSM—A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the fourth international AAAI conference on weblogs and social media*, pages. 162-169.
- Saggion, H., Funk, A., Maynard, D. and Bontcheva, K. 2007. Ontology-based information extraction for business intelligence. In *The Semantic Web* pages 843-856, Springer Berlin Heidelberg.
- 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.

# Explicit and Implicit Syntactic Features for Text Classification

Matt Post<sup>1</sup> and Shane Bergsma<sup>1,2</sup>

<sup>1</sup>Human Language Technology Center of Excellence

<sup>2</sup>Center for Language and Speech Processing

Johns Hopkins University

Baltimore, MD

## Abstract

Syntactic features are useful for many text classification tasks. Among these, tree kernels (Collins and Duffy, 2001) have been perhaps the most robust and effective syntactic tool, appealing for their empirical success, but also because they do not require an answer to the difficult question of *which* tree features to use for a given task. We compare tree kernels to different explicit sets of tree features on five diverse tasks, and find that explicit features often perform as well as tree kernels on accuracy and always in orders of magnitude less time, and with smaller models. Since explicit features are easy to generate and use (with publicly available tools), we suggest they should always be included as baseline comparisons in tree kernel method evaluations.

## 1 Introduction

Features computed over parse trees are useful for a range of discriminative tasks, including authorship attribution (Baayen et al., 1996), parse reranking (Collins and Duffy, 2002), language modeling (Cherry and Quirk, 2008), and native-language detection (Wong and Dras, 2011). A major distinction among these uses of syntax is how the features are represented. The **implicit approach** uses tree kernels (Collins and Duffy, 2001), which make predictions with inner products between tree pairs. These products can be computed efficiently with a dynamic program that produces weighted counts of all the shared tree fragments between a pair of trees, essentially incorporating all fragments without representing any of them explicitly. Tree kernel approaches

have been applied successfully in many areas of NLP (Collins and Duffy, 2002; Moschitti, 2004; Pighin and Moschitti, 2009).

Tree kernels were inspired in part by ideas from Data-Oriented Parsing (Scha, 1990; Bod, 1993), which was in turn motivated by uncertainty about which fragments to include in a grammar. However, manual and automatic approaches to inducing tree fragments have recently been found to be useful in an **explicit approach** to text classification, which employs specific tree fragments as features in standard classifiers (Post, 2011; Wong and Dras, 2011; Swanson and Charniak, 2012). These feature sets necessarily represent only a small subset of all possible tree patterns, leaving open the question of what further gains might be had from the unused fragments.

Somewhat surprisingly, explicit and implicit syntactic features have been explored largely independently. Here, we compare them on a range of classification tasks: (1,2) grammatical classification (is a sentence written by a human?), (3) question classification (what type of answer is sought by this question?), and (4,5) native language prediction (what is the native language of a text's author?).

Our main contribution is to show that an explicit syntactic feature set performs as well or better than tree kernels on each tested task, and in orders of magnitude less time. Since explicit features are simple to generate (with publicly available tools) and flexible to use, we recommend they be included as baseline comparisons in tree kernel method evaluations.

## 2 Experimental setup

We used the following feature sets:

**N-grams** All unigrams and bigrams.<sup>1</sup>

<sup>1</sup>Experiments with trigrams did not show any im-

**CFG rules** Counts of depth-one context-free grammar (CFG) productions obtained from the Berkeley parser (Petrov et al., 2006).

**C&J features** The parse-tree reranking feature set of Charniak and Johnson (2005), extracted from the Berkeley parse trees.

**TSG features** We also parsed with a Bayesian tree substitution grammar (Post and Gildea, 2009, TSG)<sup>2</sup> and extracted fragment counts from Viterbi derivations.

We build classifiers with LIBLINEAR<sup>3</sup> (Fan et al., 2008). We divided each dataset into training, dev, and test sets. We then trained an L2-regularized L1-loss support vector machine (`-s 3`) with a bias parameter of 1 (`-B 1`), optimizing the regularization parameter (`-c`) on the dev set over the range  $\{0.0001 \dots 100\}$  by multiples of 10. The best model was then used to classify the test set. A sentence length feature was included for every sentence.

For tree kernels, we used SVM-light-TK<sup>4</sup> (Moschitti, 2004; Moschitti, 2006) with the default settings (`-t 5 -D 1 -L 0.4`),<sup>5</sup> which also solves an L2-regularized L1-loss SVM optimization problem. We tuned the regularization parameter (`-c`) on the dev set in the same manner as described above, providing 4 GB of memory to the kernel cache (`-m 4000`).<sup>6</sup> We used *subset tree kernels*, which compute the similarity between two trees by implicitly enumerating all possible fragments of the trees (in contrast with *subtree kernels*, where all fragments fully extend to the leaves).

### 3 Tasks

Table 1 summarizes our datasets.

#### 3.1 Coarse grammatical classification

Our first comparison is coarse grammatical classification, where the goal is to distinguish between human-written sentences and “pseudo-negative” sentences sampled from a trigram language model constructed from in-

provement.

<sup>2</sup>[github.com/mjpost/dptsg](https://github.com/mjpost/dptsg)

<sup>3</sup>[www.csie.ntu.edu.tw/~cjlin/liblinear/](http://www.csie.ntu.edu.tw/~cjlin/liblinear/)

<sup>4</sup>[disi.unitn.it/moschitti/Tree-Kernel.htm](http://disi.unitn.it/moschitti/Tree-Kernel.htm)

<sup>5</sup>Optimizing SVM-TK’s decay parameter (`-L`) did not improve test-set accuracy, but did increase training time (squaring the number of hyperparameter combinations to evaluate), so we stuck with the default.

<sup>6</sup>Increased from the default of 40 MB, which halves the running time.

|  | train   | dev    | test   |
|--|---------|--------|--------|
| <i>Coarse grammaticality (BLLIP)</i>       |         |        |        |
| sentences                                  | 100,000 | 6,000  | 6,000  |
| <i>Fine grammaticality (PTB)</i>           |         |        |        |
| sentences                                  | 79,664  | 3,978  | 3,840  |
| <i>Question classification (TREC-10)</i>   |         |        |        |
| sentences                                  | 4,907   | 545    | 500    |
| <i>Native language (ICLE; 7 languages)</i> |         |        |        |
| documents                                  | 490     | 105    | 175    |
| sentences                                  | 17,715  | 3,968  | 6,777  |
| <i>Native language (ACL; 5 languages)</i>  |         |        |        |
| documents                                  | 987     | 195    | 185    |
| sentences                                  | 146,257 | 28,139 | 28,403 |

Table 1: Datasets.

| system | accuracy    | CPU time |
|--------|-------------|----------|
| Chance | 50.0        | -        |
| N-gram | 68.4        | minutes  |
| CFG    | 86.3        | minutes  |
| TSG    | 89.8        | minutes  |
| C&J    | <b>92.9</b> | an hour  |
| SVM-TK | 91.0        | a week   |

Table 2: Coarse grammaticality. CPU time is for classifier setup, training, and testing.

domain data (Okanohara and Tsujii, 2007). Cherry and Quirk (2008) first applied syntax to this task, learning weighted parameters for a CFG with a latent SVM. Post (2011) found further improvements with fragment-based representations (TSGs and C&J) with a regular SVM. Here, we compare their results to kernel methods. We repeat Post’s experiments on the BLLIP dataset,<sup>7</sup> using his exact data splits (Table 2). To our knowledge, tree kernels have not been applied to this task.

#### 3.2 Fine grammatical classification

Real-world grammaticality judgments require much finer-grained distinctions than the coarse ones of the previous section (for example, marking dropped determiners or wrong verb inflections). For this task, we too positive examples from all sentences of sections 2–21 of the WSJ portion of the Penn Treebank (Marcus et al., 1993). Negative examples were created by inserting one or two errors

<sup>7</sup>LDC Catalog No. LDC2000T43

| system      | accuracy    | CPU time |
|-------------|-------------|----------|
| Wong & Dras | 60.6        | -        |
| Chance      | 50.0        | -        |
| N-gram      | 61.4        | minutes  |
| CFG         | 64.5        | minutes  |
| TSG         | 67.0        | minutes  |
| C&J         | <b>71.9</b> | an hour  |
| SVM-TK      | 67.8        | weeks    |

Table 3: Fine-grained classification accuracy (the Wong and Dras (2010) score is the highest score from the last column of their Table 3).

| system             | accuracy    | CPU time    |
|--------------------|-------------|-------------|
| Pighin & Moschitti | 86.6        | -           |
| Bigram             | 73.2        | seconds     |
| CFG                | <b>90.0</b> | seconds     |
| TSG                | 85.6        | seconds     |
| C&J                | 89.6        | minutes     |
| SVM-TK             | 87.7        | twenty min. |

Table 4: Question classification (6 classes).

into the parse trees from the positive data using GenERRate (Foster and Andersen, 2009). An example sentence pair is *But the ballplayers disagree[ing]*, where the negative example incorrectly inflects the verb. Wong and Dras (2010) reported good results with parsers trained separately on the positive and negative sides of the training data and classifiers built from comparisons between the CFG productions of those parsers. We obtained their data splits (described as *NoisyWSJ* in their paper) and repeat their experiments here (Table 3).

### 3.3 Question Classification

We look next at question classification (QC). Li and Roth (2002) introduced the TREC-10 dataset,<sup>8</sup> a set of questions paired with labels that categorize the question by the type of answer it seeks. The labels are organized hierarchically into six (coarse) top-level labels and fifty (fine) refinements. An example question from the ENTY/animal category is *What was the first domesticated bird?*. Table 4 contains results predicting just the coarse labels. We compare to Pighin and Moschitti (2009), and also repeat their experiments, finding a slightly better result for them.

<sup>8</sup>[cogcomp.cs.illinois.edu/Data/QA/QC/](http://cogcomp.cs.illinois.edu/Data/QA/QC/)

| system      | sent.       | voting | whole       |
|-------------|-------------|--------|-------------|
| Wong & Dras | -           | -      | 80.0        |
| Style       | 42.0        | 75.3   | <b>86.8</b> |
| CFG         | 39.5        | 73.2   | 83.7        |
| TSG         | 38.7        | 72.1   | 83.2        |
| C&J         | <b>42.9</b> | 76.3   | 86.3        |
| SVM-TK      | 40.7        | 69.5   | -           |
| Style       | 42.5        | 65.3   | 83.7        |
| CFG         | 39.2        | 52.6   | <b>86.3</b> |
| TSG         | 40.4        | 56.8   | 84.7        |
| C&J         | <b>49.2</b> | 66.3   | 81.1        |
| SVM-TK      | 42.1        | 52.6   | -           |

Table 5: Accuracy on ICLE (7 languages, top) and ACL (five, bottom) datasets at the sentence and document levels. All documents were signature-stylized (§3.4).

We also experimented with the refined version of the task, where we directly predict one of the fifty refined categories, and found nearly identical relative results, with the best explicit feature set (CFG) returning an accuracy of 83.6% (in seconds), and the tree kernel system 69.8% (in an hour). For reference, Zhang and Lee (2003) report 80.2% accuracy when training on the full training set (5,500 examples) with an SVM and bag-of-words features.<sup>9</sup>

### 3.4 Native language identification

Native language identification (NLI) is the task of determining a text’s author’s native language. This is usually cast as a document-level task, since there are often not enough cues to identify native languages at smaller granularities. As such, this task presents a challenge to tree kernels, which are defined at the level of a single parse tree and have no obvious document-level extension. Table 5 therefore presents three evaluations: (a) sentence-level accuracy, and document-level accuracy from (b) sentence-level voting and (c) direct, whole-document classification.

We perform these experiments on two datasets. In order to mitigate topic bias<sup>10</sup> and other problems that have been reported with

<sup>9</sup>Pighin and Moschitti (2009) did not report results on this version of the task.

<sup>10</sup>E.g., when we train with all words, the keyword ‘Japanese’ is a strong indicator for Japanese authors, while ‘Arabic’ is a strong indicator for English ones.

the ICLE dataset (Tetreault et al., 2012),<sup>11</sup> we preprocessed each dataset into two signature-stylized versions by replacing all words not in a stopwords list.<sup>12</sup> The first version replaces non-stopwords with word classes computed from surface-form signatures,<sup>13</sup> and the second with POS tags.<sup>14</sup> N-gram features are then taken from both stylized versions of the corpus.

Restricting the feature representation to be topic-independent is standard-practice in stylistometric tasks like authorship attribution, gender identification, and native-language identification (Mosteller and Wallace, 1984; Koppel et al., 2003; Tomokiyo and Jones, 2001).

### 3.4.1 ICLE v.2

The first dataset is a seven-language subset of the *International Corpus of Learner English, Version 2* (ICLE) (Granger et al., 2009), which contains 3.7 million words of English documents written by people with sixteen different native languages. Table 1 contains scores, including one reported by Wong and Dras (2011), who used the CFG and C&J features, and whose data splits we mirror.<sup>15</sup>

### 3.4.2 ACL Anthology Network

We also experimented with native language classification on scientific documents using a version of the ACL Anthology Network (Radev et al., 2009, AAN) annotated for experiments in stylistometric tasks, including a native/non-native author judgment (Bergsma et al., 2012). For NLI, we further annotated this dataset in a semi-automatic fashion for the five most-common native languages of ACL authors in our training era: English, Japanese, German, Chinese, and French. The annotation heuristics, designed to favor precision over recall, provided annotations for 1,959 of 8,483 papers (23%) in the 2001–2009 AAN.<sup>16</sup>

<sup>11</sup>Including prompts, characters, and special tokens that correlate strongly with particular outcomes.

<sup>12</sup>The stopwords list contains the set of 524 SMART-system stopwords used by Tomokiyo and Jones (2001), plus punctuation and Latin abbreviations.

<sup>13</sup>For example, suffix and capitalization.

<sup>14</sup>Via CRFTagger (Phan, 2006).

<sup>15</sup>Tetreault et al. reported accuracies up to 90.1 in a cross-validation setting that isn’t directly comparable.

<sup>16</sup>Details and data at [old-site.clsp.jhu.edu/~sbergsma/Stylo/](http://old-site.clsp.jhu.edu/~sbergsma/Stylo/).

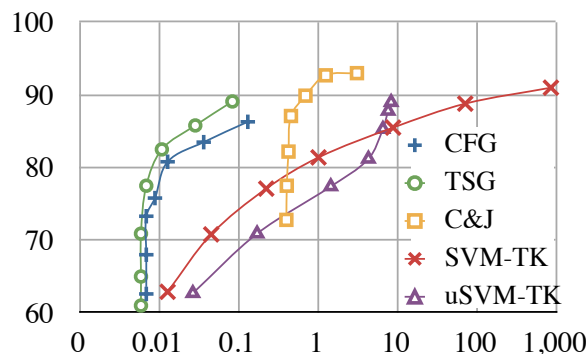


Figure 1: Training time (1000s of seconds) vs. test accuracy for coarse grammaticality, plotting test scores from models trained on 100, 300, 1k, 3k, 10k, 30k, and 100k instances.

## 4 Discussion

Syntactic features improve upon the n-gram baseline for all tasks except whole-document classification for ICLE. Tree kernels are often among the best, but always trail (by orders of magnitude) when runtime is considered. Constructing the multi-class SVM-TK models for the NLI tasks in particular was computationally burdensome, requiring cpu-months of time. The C&J features are similarly often the best, but incur a runtime cost due to the large models. CFG and TSG features balance performance, model size, and runtime. We now compare these approaches in more depth.

### 4.1 Training time versus accuracy

Tree kernel training is quadratic in the size of the training data, and its empirical slowness is known. It is informative to examine learning curves to see how the time-accuracy trade-offs extrapolate. We compared models trained on the first 100, 300, 1k, 3k, 10k, 30k, and 100k data points of the coarse grammaticality dataset, split evenly between positive and negative examples (Figure 1). SVM-TK improves over the TSG and CFG models in the limit, but at an extraordinary cost in training time: 100k training examples is already pushing the bounds of practicality for tree kernel learning, and generating curve’s next point would require several *months* of time. Kernel methods also produce large models that result in slow *test-time* performance, a problem dubbed the “curse of kernelization” (Wang et al., 2010).

Approximate kernel methods designed to scale to large datasets address this (Severyn

and Moschitti, 2010). We investigated the uSVM-TK toolkit,<sup>17</sup> which enables tuning the tradeoff between training time and accuracy. While faster than SVM-TK, its performance was never better than explicit methods along both dimensions (time and accuracy).

## 4.2 Overfitting

Overfitting is also a problem for kernel methods. The best models often had a huge number of support vectors, achieving near-perfect accuracy on the training set but making many errors on the dev. and test sets. On the ICLE task, close to 75% of all the training examples were used as support vectors. We found only half as many support vectors used for the explicit representations, implying less error (Vapnik, 1998), and saw much lower variance between training and testing performance.

## 4.3 Which fragments?

Our findings support the observations of Cumby and Roth (2003), who point out that kernels introduce a large number of irrelevant features that may be especially harmful in small-data settings, and that, when possible, it is often better to have a set of explicit, relevant features. In other words, it is better to have the *right* features than *all* of them. Tree kernels provide a robust, efficiently-computable measure of comparison, but they also skirt the difficult question, *Which fragments?*

So what are the “right” features? Table 6) presents an intuitive list from the coarse grammaticality task: phenomena such as balanced parenthetical phrases and quotations are associated with grammaticality, while small, flat, abstract rules indicate samples from the n-gram model. Similar intuitive results hold for the other tasks. The immediate interpretability of the explicit formalisms is another advantage, although recent work has shown that weights on the implicit features can also be obtained after a kind of linearization of the tree kernel (Pighin and Moschitti, 2009).

Ultimately, which features matter is task-dependent, and skirting the question is advantageous in many settings. But it is also encouraging that methods for selecting fragments and other tree features work so well,

```
(TOP (S “ S , ” NP (VP (VBZ says) ADVP) .))
(FRAG (X SYM) VP .)
(PRN (-LRB- -LRB-) S (-RRB- -RRB-))
(PRN (-LRB- -LRB-) NP (-RRB- -RRB-))
(S NP VP .)
-----
(NP (NP DT CD (NN %)) PP)
(NP DT)
(PP (IN of))
(TOP (NP NP PP PP .))
(NP DT JJ NNS)
```

Table 6: The highest- and lowest-weighted TSG features (coarse grammaticality).

yielding quick, light-weight models that contrast with the heavy machinery of tree kernels.

## 5 Conclusion

Tree kernels provide a robust measure of comparison between trees, effectively making use of all fragments. We have shown that for some tasks, it is sufficient (and advantageous) to instead use an explicitly-represented subset of them. In addition to their flexibility and interpretability, explicit syntactic features often outperformed tree kernels in accuracy, and even where they did not, the cost was *multiple orders of magnitude increase* in both training and testing time. These results were consistent across a range of task types, dataset sizes, and classification arities (binary and multiclass).

There are a number of important caveats. We explored a range of data settings, but there are many others where tree kernels have been proven useful, such as parse tree reranking (Collins and Duffy, 2002; Shen and Joshi, 2003), sentence subjectivity (Suzuki et al., 2004), pronoun resolution (Yang et al., 2006), relation extraction (Culotta and Sorensen, 2004), machine translation evaluation (Liu and Gildea, 2005), predicate-argument recognition, and semantic role labeling (Pighin and Moschitti, 2009). There are also tree kernel variations such as dependency tree kernels (Culotta and Sorensen, 2004) and shallow semantic tree kernels (Moschitti et al., 2007). These variables provide a rich environment for future work; in the meantime, we take these results as compelling motivation for the continued development of explicit syntactic features (both manual and automatically induced), and suggest that such features should be part of the baseline systems on applicable discriminative NLP tasks.

<sup>17</sup>[disi.unitn.it/~severyn/code.html](http://disi.unitn.it/~severyn/code.html)

## References

- Harald Baayen, Hans Van Halteren, and Fiona Tweedie. 1996. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121.
- Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proc. of NAACL-HLT*, pages 327–337, Montréal, Canada, June. Association for Computational Linguistics.
- Rens Bod. 1993. Using an annotated corpus as a stochastic grammar. In *Proc. of ACL*, Columbus, Ohio, USA, June.
- Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. of ACL*, pages 173–180, Ann Arbor, Michigan, USA, June.
- Colin Cherry and Chris Quirk. 2008. Discriminative, syntactic language modeling through latent SVMs. In *Proc. of AMTA*, Waikiki, Hawaii, USA, October.
- Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proc. of NIPS*.
- Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proc. of ACL*, pages 173–180, Philadelphia, Pennsylvania, USA, July.
- Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. of ACL*, pages 423–429.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Jennifer Foster and Øistein E. Andersen. 2009. GenERRate: Generating errors for use in grammatical error detection. In *Proceedings of the fourth workshop on innovative use of NLP for building educational applications*, pages 82–90.
- Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. The International Corpus of Learner English. Version 2. Handbook and CD-Rom.
- Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2003. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412.
- Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proc. of COLING*, pages 1–7.
- Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):330.
- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proc. of ACL*, pages 776–783, Prague, Czech Republic, June.
- Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proc. of ACL*.
- Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proc. of EACL*, volume 6, pages 113–120.
- Frederick Mosteller and David L. Wallace. 1984. *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. Springer-Verlag.
- Daisuke Okanohara and Jun’ichi Tsujii. 2007. A discriminative language model with pseudo-negative samples. In *Proc. of ACL*, Prague, Czech Republic, June.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL*, Sydney, Australia, July.
- Xuan-Hieu Phan. 2006. CRFTagger: CRF English POS Tagger. [crftagger.sourceforge.net](http://crftagger.sourceforge.net).
- Daniele Pighin and Alessandro Moschitti. 2009. Reverse engineering of tree kernel feature spaces. In *Proc. of EMNLP*, pages 111–120, Singapore, August.
- Matt Post and Daniel Gildea. 2009. Bayesian learning of a tree substitution grammar. In *Proc. of ACL (short paper track)*, Suntec, Singapore, August.
- Matt Post. 2011. Judging grammaticality with tree substitution grammar derivations. In *Proc. of ACL*, Portland, Oregon, USA, June.
- Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The ACL anthology network corpus. In *Proc. of ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, pages 54–61.
- Remko Scha. 1990. Taaltheorie en taaltechnologie; competence en performance. In R. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de nederlandse taalwetenschap*, pages 7–22, Almere, the Netherlands. De Vereniging.

- Aliaksei Severyn and Alessandro Moschitti. 2010. Large-scale support vector learning with structural kernels. In *Proc. of ECML/PKDD*, pages 229–244.
- Libin Shen and Aravind K. Joshi. 2003. An SVM-based voting algorithm with application to parse reranking. In *Proc. of CoNLL*, pages 9–16.
- Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. 2004. Convolution kernels with feature selection for natural language processing tasks. In *Proc. of ACL*, pages 119–126.
- Benjamin Swanson and Eugene Charniak. 2012. Native language detection with tree substitution grammars. In *Proc. of ACL (short papers)*, pages 193–197, Jeju Island, Korea, July.
- Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In *Proc. of COLING*, pages 2585–2602, Mumbai, India, December.
- Laura Mayfield Tomokiyo and Rosie Jones. 2001. You’re not from ’round here, are you? Naive Bayes detection of non-native utterances. In *Proc. of NAACL*.
- Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.
- Zhuang Wang, Koby Crammer, and Slobodan Vucetic. 2010. Multi-class pegasos on a budget. In *ICML*, pages 1143–1150.
- Sze-Meng Jojo Wong and Mark Dras. 2010. Parser features for sentence grammaticality classification. In *Proceedings of the Australasian Language Technology Association Workshop*, Melbourne, Australia, December.
- Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting parse structures for native language identification. In *Proc. of EMNLP*, pages 1600–1610, Edinburgh, Scotland, UK., July.
- Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. In *Proc. of Coling-ACL*, pages 41–48.
- Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, SIGIR ’03*, pages 26–32, New York, NY, USA. ACM.



# Does Korean defeat phonotactic word segmentation?

**Robert Daland**

Department of Linguistics  
University of California, Los Angeles  
3125 Campbell Hall, Box 951543  
Los Angeles, CA 90095-1543, USA  
r.daland@gmail.com

**Kie Zuraw**

Department of Linguistics  
University of California, Los Angeles  
3125 Campbell Hall, Box 951543  
Los Angeles, CA 90095-1543, USA  
kie@ucla.edu

## Abstract

Computational models of infant word segmentation have not been tested on a wide range of languages. This paper applies a phonotactic segmentation model to Korean. In contrast to the undersegmentation pattern previously found in English and Russian, the model exhibited more oversegmentation errors and more errors overall. Despite the high error rate, analysis suggested that lexical acquisition might not be problematic, provided that infants attend only to frequently segmented items.

## 1 Introduction

The process by which infants learn to parse the acoustic signal into word-sized units—word segmentation—is an active area of research in developmental psychology (Polka and Sundara 2012; Saffran et al. 1996) and cognitive modeling (Daland and Pierrehumbert 2011 [DP11], Goldwater et al. 2009 [GGJ09]). Word segmentation is a classic bootstrapping problem: to learn words, infants must segment the input, because around 90% of the novel word types they hear are never uttered in isolation (Aslin et al. 1996; van de Weijer 1998). However, in order to segment infants must know some words, or generalizations about the properties of words. How can infants form generalizations about words before learning words themselves?

### 1.1 DiBS

Two approaches in the literature might be termed *lexical* and *phonotactic*. Under the lexical approach, exemplified by GGJ09, infants are assumed to exploit the Zipfian distribution of lan-

guage, identifying frequently recurring and mutually predictive sequences as words. In the phonotactic approach, infants are assumed to leverage universal and/or language-specific knowledge about the phonological *content* of sequences to infer the optimal segmentation. The present study focuses on the phonotactic approach outlined in DP11, termed DiBS. For other examples of approaches that use phonotactics, see Fleck 2008, Blanchard et al. 2010.

A (Di)phone-(B)ased (S)egmentation model consists of an inventory of segment-segment sequences, with an estimated probability that a word boundary falls between the two segments. For example, when [pd] occurs in English, the probability of an intervening word boundary is very high:  $\Pr(\# \mid [pd]) \approx 1$ . These probabilities are the parameters of the model to be learned. In the supervised setting (*baseline* model), these parameters may be estimated directly from data in which the word boundaries are labeled:  $\Pr(\# \mid pd) = \text{Fr}(\# \wedge pd) / (\text{Fr}(\# \wedge pd) + \text{Fr}(-\# \wedge pd))$  where  $\text{Fr}(\# \wedge pd)$  is the number of [pd] sequences separated by a word boundary, and  $\text{Fr}(-\# \wedge pd)$  the number of [pd]'s not separated by a word boundary. For assessment purposes, these probabilities are converted to hard decisions.

DP11 describe an unsupervised learning algorithm for DiBS that exploits a positional independence assumption, treating phrase edges as a proxy for word edges (*phrasal* model). This learning model's performance on English is on par with state-of-the-art lexical models (GGJ09), reflecting the high positional informativeness of diphones in English. We apply the baseline and phrasal models to Korean.

### 1.2 Linguistic properties of Korean

Korean is unrelated to languages previously modeled (English, Dutch, French, Spanish, Ara-

bic, Greek, Russian), and it is an interesting test case for both phonotactic and lexical approaches.

Korean syntax and morphology (Sohn 1999) present a particular challenge for unsupervised learning. Most noun phrases are marked with a limited set of case suffixes, and clauses generally end in a verb, inflected with suffixes ending in a limited set of sounds ([a,ʌ,i,jɔ]). Thus, the phrase-final distribution may not reflect the overall word-final distribution—problematic for some phonotactic approaches. Similarly, the high frequency and positional predictability of affixes could lead a lexical model to treat them as words. A range of phonological processes apply in Korean, even across word boundaries (Sohn 1999), yielding extensive allomorphy. Phonotactic models may be robust to this kind of variation, but it is challenging for current lexical models (see DP11).

Korean consonantal phonology gives diphones several informative properties, including:

- Various consonant clusters (obstruent-lenis, lenis-nasal, *et al.*) are possible only if they span a word boundary
- Various consonants cannot precede a word boundary
- [ŋ] cannot follow a word boundary

Conversely, unlike in previously studied languages, vowel-vowel sequences are common word-internally. This is likely to be problematic for phonotactic models, but not for lexical ones.

## 2 Methods

We obtained a *phonetic corpus* representing Korean speech by applying a grapheme-to-phonetic converter to a text corpus. First, we conducted an analysis of this phonetic corpus, with results in Table 1. Next, for comparability with previous studies, two 750,000-word samples (representing approximately one month of child input each) were randomly drawn from the phonetic corpus—the *training* and *test* corpora. The phrasal and baseline DiBS models described above were trained and tested on these corpora; results are reported in Table 2. Finally, we inspected one ‘day’ worth of segmentations, and offer a qualitative assessment of errors.

### 2.1 Corpus and phonetic conversion

The Korean Advanced Institute of Science and Technology Raw Corpus, available from the Semantic Web Research Center, semantic-

web.kaist.ac.kr/home/index.php/KAIST\_Corpus contains approximately 70,000,000 words from speeches, novels, newspapers, and more. The corpus was preprocessed to supply phrase breaks at punctuation marks and strip XML.

The grapheme-to-phonetic conversion system of Kim et al. (2002) was generously shared by its creators. It includes morphosyntactic processing, phrase-break detection, and a dictionary of phonetic exceptions. It applies regular and lexically-conditioned phonological rules, but not optional rules. Kim et al. reported per-grapheme accuracy of 99.7% in one corpus and 99.98% in another.

An example of original text and the phonetic conversion is given below, with phonological changes in bold:

orthographic: 경기도<sub>1</sub> 여주에서<sub>2</sub> 출생<sub>3</sub>.  
중앙대<sub>4</sub> 문예창작학과를<sub>5</sub> 졸업했다<sub>6</sub>.

phonetic: ㄱ ɰ ɔ ㄱ ɰ | ㄷ ɰ 1 ㄱ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ 2  
ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ 3 # ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ 4  
ㄴ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ 5  
ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ ㅈ 6

IPA: k jʌ ŋ k i t o<sub>1</sub> jʌ tɕ u e s ʌ<sub>2</sub> tɕ<sup>h</sup> u l s<sup>\*</sup> ε  
ŋ<sub>3</sub> # tɕ u ŋ a ŋ t ε<sub>4</sub> m u n e tɕ<sup>h</sup> a ŋ tɕ a k<sup>h</sup> a k  
k<sup>\*</sup> w a l i l<sub>5</sub> tɕ o l ʌ p<sup>h</sup> ε t t<sup>\*</sup> a<sub>6</sub>

(the \* diacritic indicates tense consonants)

gloss: Born<sub>3</sub> in Yeaju<sub>2</sub>, Gyeonggi-do<sub>1</sub>.  
Graduated<sub>6</sub> from Jungang University<sub>4</sub>  
Department of Creative Writing<sub>5</sub>.

We relied on spaces in the corpus to indicate word boundaries, although, as in all languages, there can be inconsistencies in written Korean.

### 2.2 Error analysis

An under-researched issue is the nature of the errors that segmentation algorithms make. For a given input word in the test corpus, we defined the *output projection* as the minimal sequence of segmented words containing the entire input word. For example, if *the#kitty* were segmented as *thekitty*, then *thekitty* would be the output projection for both *the* and *kitty*. Similarly, for a posited word in the segmentation/output of the test corpus, we defined the *input projection*. For example, if *the#kitty* were segmented as *theki#tty*, then the *the#kitty* would be the input projection of both *theki* and *tty*. For each word, we examined the input-output relationship. Several questions were of interest. Are highly frequent items segmented frequently enough that the child is likely to be able to learn them? Is it

the case that all or most items which are segmented frequently are themselves words? Are there predicted errors which seem especially serious or difficult to overcome?

### 3 Results and discussion

The 1350 distinct diphones found in the phonetic corpus were grouped into phonological classes. Table 1 indicates the probabilities (percentage) that a word boundary falls inside the diphone; when the class contains 3 or more diphones, the median and range are shown. Because of various phonological processes, some sequences cannot exist (blank cells), some can occur only word-internally (marked *int*), and some can occur only across word boundaries (marked *span*). For example, the velar nasal [ŋ] cannot begin a word, so diphones of the form *Xŋ* must be word-internal. Conversely, a lenis-/h/ sequence indicates a word boundary, because within a word a lenis stop merges with following /h/ to become an aspirated stop. If all diphones in a cell have a spanning rate above 90%, the cell says *span\**, and if below 10%, *int\**. This means that all the diphones in that class are highly informative; other classes contain a mix of more and less informative diphones.

The performance of the DiBS models is shown in Table 2. An undersegmentation error is a true word boundary which the segmentation algorithm fails to find (*miss*), while an oversegmentation error is a falsely posited boundary (*false alarm*). The under- and over-segmentation

error rates are defined as the number of such errors per word (percent). We also report the precision, recall, and F scores for boundary detection, word token segmentation, and type segmentation (for details see DP11, GGJ09).

| <i>model</i>           | <b>baseline</b> | <b>phrasal</b> |
|------------------------|-----------------|----------------|
| under (errs per wd)    | 43.4            | 72.5           |
| over (errs per wd)     | 17.7            | 22.0           |
| prec (bdry/tok/type)   | 68/36/34        | 28/11/12       |
| recall (bdry/tok/type) | 46/27/29        | 11/6/8         |
| F (bdry/tok/type)      | 55/31/31        | 15/8/9         |

Table 2: Results of DiBS models

On the basis of the fact that the oversegmentation error rate in English and Russian was consistently below 10% (<1 error/10 wds), DP11 conjectured that phonotactic segmenters will, cross-linguistically, avoid significant oversegmentation. The results in Table 2 provide a counterexample: oversegmentation is distinctly higher than in English and Russian. Indeed, Korean is a more challenging language for purely phonotactic segmentation.

#### 3.1 Phonotactic cues to word segmentation

Because phonological processes are more likely to apply word-internally, word-internal sequences are more predictable (Aslin et al. 1996; DP11; GGJ09; Saffran et al. 1996; van de Weijer 1998). The phonology of Korean is a potentially

| seg. 2<br>seg. 1  | lenis<br>stop | lenis<br>non-stop | tense       | asp.        | h          | n             | m          | ŋ   | liquid        | vowel       | diphth.     |
|-------------------|---------------|-------------------|-------------|-------------|------------|---------------|------------|-----|---------------|-------------|-------------|
| lenis stop        | span          | 100<br>4-100      | int*        | 27<br>5-53  | span       | 100<br>98-100 | span       |     | 100<br>10-100 | int*        | 7<br>0-100  |
| lenis<br>non-stop |               |                   |             |             |            |               |            |     |               | int         | int         |
| tense             |               |                   |             |             |            |               |            |     |               | int         | int         |
| aspirated         |               |                   |             |             |            |               |            |     |               | int         | int         |
| h                 |               |                   |             |             |            |               |            |     |               | int         | int         |
| n                 | 65<br>29-66   | 46, 57            | 38<br>18-82 | 45<br>32-67 | 35         | 32            | 61         |     | span*         | 12<br>1-37  | 53<br>20-99 |
| m                 | 19<br>14-21   | 18, 18            | 14<br>4-57  | 14<br>12-26 | 14         | int*          | 21         |     | span          | int*        | 12<br>1-92  |
| ŋ                 | 12<br>11-13   | 10, 12            | 9<br>6-55   | 11<br>10-15 | int*       | int*          | 10         |     | span          | 6<br>0-64   | 18<br>4-86  |
| liquid            | 55<br>43-63   | 84, 88            | 71<br>6-90  | 53<br>17-68 | 42         | 90            | 53         |     | int*          | 3<br>0-14   | 39<br>7-95  |
| vowel             | 16<br>6-87    | 32<br>12-82       | 36<br>4-97  | 18<br>3-88  | 38<br>9-84 | 5<br>1-31     | 13<br>2-70 | int | int*          | 44<br>1-90  | 51<br>3-100 |
| diphthong         | 10<br>0-79    | 12<br>0-55        | 21<br>0-100 | 11<br>0-87  | 16<br>0-88 | 3<br>0-15     | 19<br>0-74 | int | int*          | 26<br>0-100 | 31<br>0-100 |

Table 1: Diphone behavior

rich source of information for word segmentation: obstruent-initial diphones are generally informative as to the presence/absence of word boundaries. However, as we suspected, vowel-vowel sequences are problematic, since they occur freely both within words and across word boundaries. Korean differs from English in that most English diphones occur nearly exclusively within words, or nearly exclusively across word boundaries (DP11), while in Korean most sonorant-obstruent sequences occur both within and across words.

### 3.2 Errors and word-learning

It seems reasonable to assume that word-learning is best facilitated by seeing multiple occurrences of a word. A segmentation that is produced only once might be ignored; thus we defined an input or output projection as frequent if it occurred more than once in the test sample.

A word learner relying on a phonotactic model could expect to successfully identify many frequent words. For 73 of the 100 most frequent input words, the only frequent output projection in the baseline model was the input word itself, meaning that the word was segmented correctly in most contexts. For 20 there was no frequent output projection, meaning that the word was not segmented consistently across contexts, which we assume is noise to the learner. In the phrasal model, for 16 items the most frequent output projection was the input word itself and for 64 there was no frequent output projection.

Conversely, of the 100 most frequent potential words identified by the baseline model, in 26 cases the most frequent input projection was the output word itself: a real word was correctly identified. In 26 cases there was no frequent input projection, and in 48 another input projection was at least as frequent as the output word. One such example is [mjʌn] ‘cotton’, frequently segmented out when it was a bound morpheme (‘if’ or ‘how many’). The most frequently segmented item was [ke], which can be a freestanding word (‘there/thing’), but was often segmented out from words suffixed with [-ke] ‘-ly/to’ and [-eke] ‘to’.

What do these results mean for a child using a phonotactic strategy? First, many of the types segmented in a day would be experienced only once (and presumably ignored). Second, infants would not go far astray if they learned frequently-segmented items as words.

### 3.3 Phrase edges and independence

We suspected the reason that the phrasal DiBS model performed so much worse than baseline was its assumption that phrase-edge distributions approximate word-edge distributions. Phrase beginnings were a good proxy for word beginnings, but there were mismatches phrase-finally. For example, [a] is much more frequent phrase-finally than word-finally (because of common verb suffixes ending in [a]), while [n] is much more frequent word-finally (because of non-sentence-final suffixes ending in [n]). The positional independence assumption is too strong.

## 4 Conclusion

This paper extends previous studies by applying a computational learning model of phonotactic word segmentation to Korean. Various properties of Korean led us to believe it would challenge both unsupervised phonotactic and lexical approaches.

Phonological and morphological analysis of errors yielded novel insights. For example, the generally greater error rate in Korean is partly caused by a high tolerance for vowel-vowel sequences within words. Interactions between morphology and word order result in violations of a key positional independence assumption.

Phonotactic segmentation was distinctly worse than in previous languages (English, Russian), particularly for oversegmentation errors. This implies the segmentation of simplistic diphone models is not cross-linguistically stable, a finding that aligns with other cross-linguistic comparisons of segmentation algorithms. In general, distinctly worse performance is found for languages other than English (Sesotho: Blanchard et al. 2010; Arabic and Spanish: Fleck 2008). These facts suggest that the successful segmentation model must incorporate richer phonotactics, or integrate some lexical processing. On the bright side, we found that frequently segmented items were mostly words, so a high segmentation error rate does not necessarily translate to a high error rate for word-learning.

## References

- Aslin, R. N., Woodward, J. Z., LaMendola, N. P., & Bever, T. G. (1996). Models of word segmentation in fluent maternal speech to infants. In J. L. Morgan & K. Demuth (Eds.). *Signal to syntax*. Mahwah, NJ: LEA, pp. 117–134.
- Blanchard, D., Heinz, J., & Golinkoff, R. (2010). Modeling the contribution of phonotactic cues to

- the problem of word segmentation. *Journal of Child Language* 37(3), 487-511.
- Daland, R. & Pierrehumbert, J.B. (2011). Learnability of diphone-based segmentation. *Cognitive Science* 35(1), 119-155.
- Fleck, M. (2008). Lexicalized phonotactic word segmentation. *Proceedings of ACL-08: HLT*, 130-138.
- Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition* 112(1), 21-54.
- Kim, B., Lee, G., & Lee, J.-H. (2002). Morpheme-based grapheme to phoneme conversion using phonetic patterns and morphophonemic connectivity information. *ACM Trans. Asian Lang. Inf. Process.* 1(1), 65-82.
- Polka, L. & Sundara, M. (2012). Word segmentation in monolingual infants acquiring Canadian-English and Canadian-French: Native language, cross-language and cross-dialect comparisons. *Infancy* 17(2), 198-232.
- Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science* 275(5294), 1926-1928.
- Sohn, H.-M. (1999). *The Korean Language*. Cambridge: Cambridge University Press.
- van de Weijer, J. (1998). Language input for word discovery. *MPI series in psycholinguistics (No. 9)*.

# Word surprisal predicts N400 amplitude during reading

Stefan L. Frank<sup>1,2</sup>   Leun J. Otten<sup>3</sup>   Giulia Galli<sup>3</sup>   Gabriella Vigliocco<sup>2</sup>

{s.frank, l.otten, g.galli, g.vigliocco}@ucl.ac.uk

<sup>1</sup>Centre for Language Studies, Radboud University Nijmegen

<sup>2</sup>Department of Cognitive, Perceptual and Brain Sciences, University College London

<sup>3</sup>Institute of Cognitive Neuroscience, University College London

## Abstract

We investigated the effect of word surprisal on the EEG signal during sentence reading. On each word of 205 experimental sentences, surprisal was estimated by three types of language model: Markov models, probabilistic phrase-structure grammars, and recurrent neural networks. Four event-related potential components were extracted from the EEG of 24 readers of the same sentences. Surprisal estimates under each model type formed a significant predictor of the amplitude of the N400 component only, with more surprising words resulting in more negative N400s. This effect was mostly due to content words. These findings provide support for surprisal as a generally applicable measure of processing difficulty during language comprehension.

## 1 Introduction

Many studies of human language comprehension measure the brain's electrical activity during reading. Such electroencephalography (EEG) experiments have revealed that the EEG signal displays systematic variation in response to the appearance of each word. The different components that can be observed in this signal are known as event-related potentials (ERPs). Probably the most reliably observed (and most studied) of these components is a negative-going deflection at centroparietal electrodes that peaks at around 400 ms after word onset and is therefore referred to as the N400 component.

It is well known that the N400 increases in amplitude (i.e., becomes more negative) when the word leads to comprehension difficulty. To study the general relation between word predictability and the N400, Dambacher et al. (2006) obtained

subjective word-probability estimates (so-called *cloze* probabilities) by asking participants to predict the upcoming word at each point in a large number of sentences. A different group of subjects read these same sentences while their EEG signal was recorded. Results showed a correlation between N400 amplitude and cloze probability: Less predictable words yielded stronger N400s.

We investigated whether similar results can be obtained using more objective, model-based word probabilities. For each word in a collection of English sentences, estimates of its *surprisal* (i.e., its negative log-transformed conditional probability:  $-\log P(w_t|w_1, \dots, w_{t-1})$ ) were generated by three types of language model: Markov (i.e., *n*-gram) models, phrase-structure grammars (PSGs), and recurrent neural networks (RNNs). Next, EEG signals of participants reading the same sentences were recorded. A comparison of word surprisal to different ERP components revealed that, indeed, N400 amplitude was predicted by surprisal values: More surprising words resulted in more negative N400s, at least for content words.

## 2 Language models

A range of models of each type was trained, allowing to investigate whether models that capture the language statistics more accurately also yield better predictions of ERP size. Such a relation is generally found in studies that use word-reading time as the dependent variable (Fernandez Monsalve et al., 2012; Frank and Bod, 2011; Frank and Thompson, 2012), providing additional support that these psychological data are indeed explained by the surprisal values and not by some confounding variable.

### 2.1 Corpus data

All models were trained on sentences from the written texts in the British National Corpus (BNC). First, the 10,000 word types with highest

frequency were selected from the BNC. Next, all sentences were extracted that contained only those words. This resulted in a training corpus of 1.06 million sentences (12.6 million word tokens).

Each trained model estimated a surprisal value for each word of the 205 sentences (1931 word tokens) for which eye-tracking data are available in the UCL corpus of reading times (Frank et al., in press). These sentences, which were selected from three unpublished novels, only contained words from the 10,000 high-frequency word list.

## 2.2 Markov models

Markov models were trained with modified Kneser-Ney smoothing (Chen and Goodman, 1999) as implemented in SRILM (Stolcke, 2002). Model order was varied:  $n = 2, 3, 4$ . No unigram model was computed because word frequency was factored out during data analysis (see Section 4.2).

## 2.3 Recurrent neural networks

The RNN model architecture has been thoroughly described elsewhere (Fernandez Monsalve et al., 2012; Frank, in press) so it is not discussed here. The only difference with previous versions was that the current RNN was trained on a substantially larger data set with more word types. A range of RNN models was obtained by training on nine increasingly large subsets of the BNC data, comprising 2K, 5K, 10K, 20K, 50K, 100K, 200K, 400K, and all 1.06M sentences. In addition, the network was trained on the full set twice, making a total of ten instantiations of the RNN model.

## 2.4 Phrase-structure grammars

To prepare data for PSG training, the selected BNC sentences were parsed by the Stanford parser (Klein and Manning, 2003). The resulting treebank was divided into nine increasingly large subsets, equal to those used for RNN training.<sup>1</sup> Grammars were induced from these subsets using the algorithm by Roark (2001) with its standard settings. Next, surprisal values on the experimental sentences were generated by Roark's incremental parser. Since increasing the parser's beam width has been shown to improve both word-probability estimates and the fit to word-reading times (Frank, 2009), the parser's 'base beam threshold' parameter was reduced to  $10^{-20}$ .

<sup>1</sup>Because not all experimental sentences could be parsed when the treebank comprised only 2K sentences, 1K sentences were added to the smallest subset.

## 3 EEG data collection

Twenty-four healthy, adult volunteers from the UCL Psychology subject pool took part in the reading study. Their EEG was recorded continuously from 32 channels during the presentation of 5 practice sentences and the 205 experimental items. Participants were asked to minimise blinks, eye movements, and head movements during sentence presentation.

Each sentence was preceded by a centrally presented fixation cross. As soon as the participant pressed a key, the cross was replaced by the sentence's first word, which was then automatically replaced by each subsequent word. Word presentation duration (in milliseconds) equalled  $190 + 20k$ , where  $k$  is the number of characters in the word (including any attached punctuation). After the word disappeared, there was a 390 ms interval before the next word appeared.

The sentences were presented in random order, one word at a time, always centrally located on the monitor. One-hundred and ten of the experimental sentences were followed by a yes/no-comprehension question, to ensure that participants tried to understand the sentences. All participants answered at least 80% of the comprehension questions correctly.

## 4 Data analysis

### 4.1 ERP components

Four ERP components of interest were identified from the literature on EEG and sentence reading: Early Left Anterior Negativity (ELAN), P200, N400, and a post-N400 positivity (PNP). Table 1 lists the corresponding time windows and approximate electrode sites.<sup>2</sup> For each component, the average electrode potential over the corresponding time window and electrodes was computed. These average ERP amplitudes served as the four dependent variables for data analysis.

The ELAN component is generally thought of as indicative of difficulty with constructing syntactic phrase structure (Friederici et al., 1999; Gunter et al., 1999; Neville et al., 1991). Hence, if any of the model types predicts ELAN size, we would expect this to be the PSG.

Dambacher et al. (2006) found effects of word frequency or length (which are strongly correlated

<sup>2</sup>The P600 component (Osterhout and Holcomb, 1992) was not included because the shortest interval between consecutive word onsets was only 600 ms.

| Component | Time window | Location       |
|-----------|-------------|----------------|
| ELAN      | 125–175 ms  | left anterior  |
| P200      | 140–200 ms  | frontocentral  |
| N400      | 300–500 ms  | centroparietal |
| PNP       | 400–600 ms  | frontopolar    |

Table 1: Investigated ERP components, their time windows, and approximate scalp locations.

and therefore difficult to tease apart) on the P200 amplitude. Since we factor out these two lexical factors in the analysis, we expect no additional effect of surprisal on P200.

If any of the components is sensitive to word surprisal, this is most likely to be the N400 as many studies have already shown that N400 amplitude depends on subjective word predictability (Dambacher et al., 2006; Kutas and Hillyard, 1984; Moreno et al., 2002). Whether an effect will appear on the PNP is more doubtful. Van Petten and Luka (2012) argue that word expectations that are confirmed result in reduced N400 size, whereas expectations that are *disconfirmed* increase the PNP. However, in a probabilistic setting, expectations are not all-or-nothing so there is no strict distinction between confirmation and disconfirmation. Nevertheless, surprisal effects on PNP may occur. Since the PNP has received relatively little attention, the component may not be such a reliable index of comprehension difficulty as the N400 has proven to be.

## 4.2 Regression analysis

Data were discarded on words attached to a comma, clitics, sentence-initial, and sentence-final words. Moreover, artifacts in the EEG data (mostly due to eye blinks) were identified and removed, leaving 32,010 analysed data points per investigated ERP component. For each data point and ERP component, a baseline potential was determined by averaging over the component’s electrodes in the 100 ms leading up to word onset.

In order to quantify the fit of surprisal to ERP size, a linear mixed-effects regression model was fitted to each of the four ERPs, using the predictors: baseline potential, log-transformed word frequency, word length (number of characters), word position in the sentence, and sentence position in the experiment.<sup>3</sup> Also, all significant

<sup>3</sup>For word and sentence position, both linear and squared factors were included in order to capture possible non-linear

two-way interactions were included (main effects were removed if they were not significant and did not appear in any interaction). In addition, there were by-subject and by-item random intervals, as well as significant by-subject and by-item random slopes. Parameters for the correlation between random intercept and slope were also estimated, if they significantly contributed to model fit.

When the surprisal estimates by a particular language model are included in the analysis, the regression model’s deviance decreases. The size of this decrease is the  $\chi^2$ -statistic of a likelihood-ratio test for significance of the surprisal effect, and was taken as the measure of the surprisal values’ fit to the ERP data.<sup>4</sup> Negative values will be used to indicate effects in the negative direction, that is, when higher surprisal results in more negative (or less positive) going ERP deflections.

## 5 Results

### 5.1 Surprisal effects

Figure 1 plots the fit of each model’s surprisal estimates to ERP amplitude as a function of the average natural  $\log P(w_t|w_1, \dots, w_{t-1})$ , which quantifies to what extent the model has acquired accurate language statistics.<sup>5</sup> For the ELAN, P200 and PNP components, there were no significant effects after correcting for multiple comparisons. In contrast, effects on the N400 were highly significant.

### 5.2 Model comparison

Table 2 shows results of pairwise comparisons between the best models of each type (i.e., those whose surprisal estimates fit the N400 data best). Clearly, RNN-based surprisal explains variance over and above each of the other two models whereas neither the  $n$ -gram nor the PSG model outperforms the RNN. Moreover, the RNN’s surprisals explain a marginally significant ( $\chi^2 = 3.47; p < .07$ ) amount of variance over and above the *combined* PSG and  $n$ -gram surprisals.

changes over the course of the sentence or experiment.

<sup>4</sup>This definition equals what Frank and Bod (2011) call ‘psychological accuracy’ in an analysis of reading times.

<sup>5</sup>This measure, which Frank and Bod (2011) call ‘linguistic accuracy’, equals the negative logarithm of the model’s perplexity. Increasing the amount of training data (or the value of  $n$ ) resulted in higher linguistic accuracy, except for the three PSG models trained on the smallest amounts of data. This shows that the models did not suffer from overfitting.



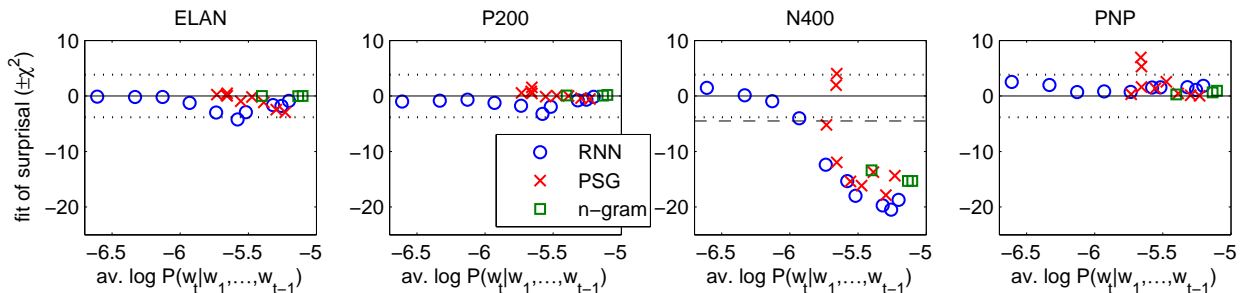


Figure 1: Fit to surprisal of ERP amplitude (for ELAN, P200, N400, and PNP components) as a function of average  $\log P(w_t|w_1, \dots, w_{t-1})$ . Each plotted point corresponds to predictions by one of the trained models. Dotted lines indicate  $\chi^2 = \pm 3.84$ , beyond which effects are statistically significant ( $p < .05$ ) if no correction for multiple comparisons is applied. The dashed line indicates the level below which effects are significant after applying the correction proposed by Benjamini and Hochberg (1995), on each ERP component separately because of our prior expectation that effects would occur mostly (if not exclusively) on the N400 component.

| Model          | <i>n</i> -gram               | RNN                         | PSG                          |
|----------------|------------------------------|-----------------------------|------------------------------|
| <i>n</i> -gram |                              | $\chi^2 = 1.34$<br>$p > .2$ | $\chi^2 = 1.66$<br>$p > .1$  |
| RNN            | $\chi^2 = 6.52$<br>$p < .02$ |                             | $\chi^2 = 4.78$<br>$p < .05$ |
| PSG            | $\chi^2 = 4.20$<br>$p < .05$ | $\chi^2 = 2.14$<br>$p > .1$ |                              |

Table 2: Pairwise comparisons between surprisal estimates by the best models of each type. Shown are the results of likelihood-ratio tests for the effect of one set of surprisal estimates (rows) over and above the other (columns).

### 5.3 Comparing word classes

N400 effects are nearly exclusively investigated on content (i.e., open-class) words. Dambacher et al. (2006), too, investigated the relation between ERP amplitudes and cloze probabilities on content words only. When running separate analyses on content and function words (constituting 53.2% and 46.8% of the data, respectively), we found that the N400 effect of Figure 1 is nearly fully driven by content words (see Figure 2). None of the models' surprisal estimates formed a significant predictor of N400 amplitude on function words, after correction for multiple comparisons.

## 6 Discussion

We demonstrated a clear effect of word surprisal, as estimated by different language models, on the EEG signal: The larger a (content) word's surprisal value, the more negative the resulting N400.

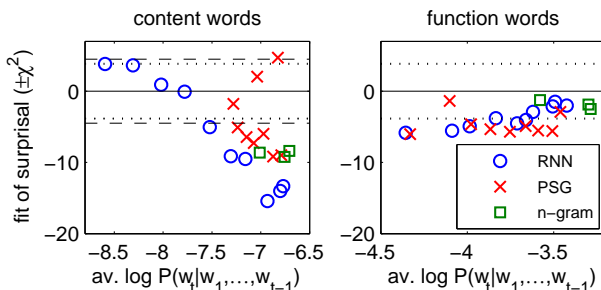


Figure 2: Fit to surprisal of N400 amplitude, for content words (left) and function words (right). Dotted lines indicate  $\chi^2 = \pm 3.84$ , beyond which effects are statistically significant ( $p < .05$ ) without correcting for multiple comparisons. Dashed lines indicates the levels beyond which effects are significant after multiple-comparison correction (Benjamini and Hochberg, 1995).

The N400 component is generally viewed as indicative of lexical rather than syntactic processing (Kaan, 2007), which may explain why surprisal under the PSG model did not have any significant explanatory value over and above RNN-based surprisal. The relatively weak performance of our Markov models is most likely due to their strict (and cognitively unrealistic) limit on the size of the prior context upon which word-probability estimates are conditioned.

Unlike the ELAN, P200, and PNP components, the N400 is known to be sensitive to the cloze probability of content words. The fact that surprisal effects were found on the N400 only, therefore suggests that subjective predictability scores and model-based surprisal estimates form opera-

tionalisations of one and the same underlying cognitive factor. Needless to say, our statistical models fail to capture many information sources, such as semantics and discourse, that do affect cloze probabilities. However, it is possible in principle to integrate these into probabilistic language models (Dubey et al., 2011; Mitchell et al., 2010).

To the best of our knowledge, only one other published study relates language model predictions to the N400: Parviz et al. (2011) found that surprisal estimates (corrected for word frequency) from an  $n = 4$  Markov model predicted N400 size as measured by magnetoencephalography (rather than EEG). Although their PSG-based surprisals did not correlate with N400 size, a related measure derived from the PSG –lexical entropy– did. However, Parviz et al. (2011) only looked at effects on the sentence-final content word of items constructed for a speech perception experiment (Kalikow et al., 1977), rather than investigating surprisal’s general predictive value across words of naturally occurring sentences, as we did here.

Our experimental design was parametric rather than factorial, which allowed us to study the effect of surprisal over a sample of English sentences rather than carefully manipulating surprisal while holding other factors constant. This has the advantage that our findings are likely to generalise to other sentence stimuli, but it can also raise a possible concern: The N400 effect may not be due to surprisal itself, but to an unknown confounding variable that was not included in the regression analysis. However, this seems unlikely because of two additional findings that only follow naturally if surprisal is indeed the relevant predictor: Significant results only appeared where they were most expected a priori (i.e., on N400 but not on other components) and there was a nearly monotonic relation between the models’ word-prediction accuracy and their ability to account for N400 size.

## 7 Conclusion

Although word surprisal has often been shown to be predictive of word-reading time (Fernandez Monsalve et al., 2012; Frank and Thompson, 2012; Smith and Levy, in press), a general effect on the EEG signal has not before been demonstrated. Hence, these results provide additional evidence in support of surprisal as a reliable measure of cognitive processing difficulty during sentence comprehension (Hale, 2001; Levy, 2008).

## Acknowledgments

The research presented here was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant number 253803. The authors acknowledge the use of the UCL *Leigion* High Performance Computing Facility, and associated support services, in the completion of this work.

## References

- Y. Benjamini and Y. Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57:289–300.
- S. F. Chen and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394.
- M. Dambacher, R. Kliegl, M. Hofmann, and A. M. Jacobs. 2006. Frequency and predictability effect on event-related potentials during reading. *Brain Research*, 1084:89–103.
- A. Dubey, F. Keller, and P. Sturt. 2011. A model of discourse predictions in human sentence processing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 304–312. Edinburgh, UK: Association for Computational Linguistics.
- I. Fernandez Monsalve, S. L. Frank, and G. Vigliocco. 2012. Lexical surprisal as a general predictor of reading time. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 398–408. Avignon, France: Association for Computational Linguistics.
- S. L. Frank and R. Bod. 2011. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological Science*, 22:829–834.
- S. L. Frank and R. L. Thompson. 2012. Early effects of word surprisal on pupil size during reading. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pages 1554–1559. Austin, TX: Cognitive Science Society.
- S. L. Frank, I. Fernandez Monsalve, R. L. Thompson, and G. Vigliocco. in press. Reading time data for evaluating broad-coverage models of English sentence processing. *Behavior Research Methods*.
- S. L. Frank. 2009. Surprisal-based comparison between a symbolic and a connectionist model of sentence processing. In N. A. Taatgen and H. van Rijn, editors, *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 1139–1144. Austin, TX: Cognitive Science Society.

- S. L. Frank. in press. Uncertainty reduction as a measure of cognitive processing load in sentence comprehension. *Topics in Cognitive Science*.
- A. D. Friederici, K. Steinhauer, and S. Frisch. 1999. Lexical integration: sequential effects of syntactic and semantic information. *Memory & Cognition*, 27:438–453.
- T. C. Gunter, A. D. Friederici, and A. Hahne. 1999. Brain responses during sentence reading: Visual input affects central processes. *NeuroReport*, 10:3175–3178.
- J. T. Hale. 2001. A probabilistic Early parser as a psycholinguistic model. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics*, volume 2, pages 159–166. Pittsburgh, PA: Association for Computational Linguistics.
- E. Kaan. 2007. Event-related potentials and language processing: a brief overview. *Language and Linguistics Compass*, 1:571–591.
- D. N. Kalikow, K. N. Stevens, and L. L. Elliott. 1977. Development of a test of speech intelligibility in noise using sentence materials with controlled word predictability. *Journal of the Acoustical Society of America*, 61:1337–1351.
- D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430. Sapporo, Japan: Association for Computational Linguistics.
- M. Kutas and S. A. Hillyard. 1984. Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307:161–163.
- R. Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106:1126–1177.
- J. Mitchell, M. Lapata, V. Demberg, and F. Keller. 2010. Syntactic and semantic factors in processing difficulty: An integrated measure. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 196–206. Uppsala, Sweden: Association for Computational Linguistics.
- E. M. Moreno, K. D. Federmeier, and M. Kutas. 2002. Switching languages, switching *palabras* (words): an electrophysiological study of code switching. *Brain and Language*, 80:188–207.
- H. Neville, J. L. Nicol, A. Barss, K. I. Forster, and M. F. Garrett. 1991. Syntactically based sentence processing classes: evidence from event-related brain potentials. *Journal of Cognitive Neuroscience*, 3:151–165.
- L. Osterhout and P. J. Holcomb. 1992. Event-related brain potentials elicited by syntactic anomaly. *Journal of Memory and Language*, 31:785–806.
- M. Parviz, M. Johnson, B. Johnson, and J. Brock. 2011. Using language models and Latent Semantic Analysis to characterise the N400m neural response. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 38–46. Canberra, Australia.
- B. Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27:249–276.
- N. J. Smith and R. Levy. in press. The effect of word predictability on reading time is logarithmic. *Cognition*.
- A. Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904. Denver, Colorado.
- C. Van Petten and B. J. Luka. 2012. Prediction during language comprehension: benefits, costs, and ERP components. *International Journal of Psychophysiology*, 83:176–190.

# Computerized Analysis of a Verbal Fluency Test

James O. Ryan<sup>1</sup>, Serguei Pakhomov<sup>1</sup>, Susan Marino<sup>1</sup>,  
Charles Bernick<sup>2</sup>, and Sarah Banks<sup>2</sup>

<sup>1</sup> College of Pharmacy, University of Minnesota

<sup>2</sup> Lou Ruvo Center for Brain Health, Cleveland Clinic

{ryanx765, pakh0002, marin007}@umn.edu

{bernicc, bankss2}@ccf.org

## Abstract

We present a system for automated phonetic clustering analysis of cognitive tests of phonemic verbal fluency, on which one must name words starting with a specific letter (e.g., ‘F’) for one minute. Test responses are typically subjected to manual phonetic clustering analysis that is labor-intensive and subject to inter-rater variability. Our system provides an automated alternative. In a pilot study, we applied this system to tests of 55 novice and experienced professional fighters (boxers and mixed martial artists) and found that experienced fighters produced significantly longer chains of phonetically similar words, while no differences were found in the total number of words produced. These findings are preliminary, but strongly suggest that our system can be used to detect subtle signs of brain damage due to repetitive head trauma in individuals that are otherwise unimpaired.

## 1 Introduction

The neuropsychological test of phonemic verbal fluency (PVF) consists of asking the patient to generate as many words as he or she can in a limited time (usually 60 seconds) that begin with a specific letter of the alphabet (Benton et al., 1989). This test has been used extensively as part of larger cognitive test batteries to study cognitive impairment resulting from a number of neurological conditions, including Parkinson’s and Huntington’s diseases, various forms of dementia, and traumatic brain injury (Troyer et al., 1998a,b; Raskin et al., 1992; Ho et al., 2002). Patients with these disorders tend to generate significantly fewer words on this test than do healthy individuals. Prior studies have also found that clustering (the degree

to which patients generate groups of phonetically similar words) and switching (transitioning from one cluster to the next) behaviors are also sensitive to the effects of these neurological conditions.

Contact sports such as boxing, mixed martial arts, football, and hockey are well known for high prevalence of repetitive head trauma. In recent years, the long-term effects of repetitive head trauma in athletes has become the subject of intensive research. In general, repetitive head trauma is a known risk factor for chronic traumatic encephalopathy (CTE), a devastating and untreatable condition that ultimately results in permanent disability and premature death (Omalu et al., 2010; Gavett et al., 2011). However, little is currently known about the relationship between the amount of exposure to head injury and the magnitude of risk for developing these conditions. Furthermore, the development of new behavioral methods aimed at detection of subtle early signs of brain impairment is an active area of research.

The PVF test is an excellent target for this research because it is very easy to administer and has been shown to be sensitive to the effects of acute traumatic brain injury (Raskin and Rearick, 1996). However, a major obstacle to using this test widely for early detection of brain impairment is that clustering and switching analyses needed to detect these subtle changes have to be done manually. These manual approaches are extremely labor-intensive, and are therefore limited in the types of clustering analyses that can be performed. Manual methods are also not scalable to large numbers of tests and are subject to inter-rater variability, making the results difficult to compare across subjects, as well as across different studies. Moreover, traditional manual clustering and switching analyses rely primarily on word orthography to determine phonetic similarity (e.g., by comparing the first two letters of two words), rather than phonetic representations, which would be prohibitively time-

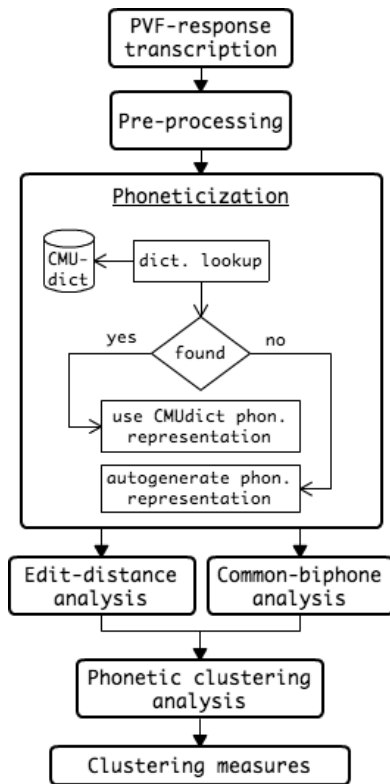


Figure 1: High-level system architecture and workflow.

consuming to obtain by hand.

Phonetic similarity has been investigated in application to a number of research areas, including spelling correction (Toutanova and Moore, 2002), machine translation (Knight and Graehl, 1998; Kondrak et al., 2003), cross-lingual information retrieval (Melamed, 1999; Fujii and Ishikawa, 2001), language acquisition (Somers, 1998), historical linguistics (Raman et al., 1997), and social-media informatics (Liu et al., 2012); we propose a novel clinical application.

Our objective was to develop and pilot-test a relatively simple, but robust, system for automatic identification of word clusters, based on phonetic content, that uses the CMU Pronouncing Dictionary, a decision tree-based algorithm for generating pronunciations for out-of-dictionary words, and two different approaches to calculating phonetic similarity between words.

We first describe the system architecture and our phonetic-similarity computation methods, and then present the results of a pilot study, using data from professional fighters, demonstrating the utility of this system for early detection of subtle signs of brain impairment.

## 2 Automated Clustering Analysis

Figure 1 shows the high-level architecture and workflow of our system.

### 2.1 Pronunciation Dictionary

We use a dictionary developed for speech recognition and synthesis applications at the Carnegie Mellon University (CMUdict). CMUdict contains phonetic transcriptions, using a phone set based on ARPABET (Rabiner and Juang, 1993), for North American English word pronunciations (Weide, 1998). We used the latest version, *cmudict.0.7a*, which contains 133,746 entries.

From the full set of entries in CMUdict, we removed alternative pronunciations for each word, leaving a single phonetic representation for each heteronymous set. Additionally, all vowel symbols were stripped of numeric stress markings (e.g., AH1 → AH), and all multicharacter phone symbols were converted to arbitrary single-character symbols, in lowercase to distinguish these symbols from the original single-character ARPABET symbols (e.g., AH → c). Finally, whitespace between the symbols constituting each phonetic representation was removed, yielding compact phonetic-representation strings suitable for computing our similarity measures.

To illustrate, the CMUdict pronunciation entry for the word *phonetic*, [F AH0 N EH1 T IH0 K], would be represented as FcNiTmK.

### 2.2 Similarity Computation

Our system uses two methods for determining phonetic similarity: edit distance and a common-biphone check. Each of these methods gives a measure of similarity for a pair of phonetic representations, which we respectively call a *phonetic-similarity score* (PSS) and a *common-biphone score* (CBS).

For PSS, we first compute the Levenshtein distance (Levenshtein, 1966) between compact phonetic-representation strings and normalize that to the length of the longer string; then, that value is subtracted from 1. PSS values range from 0 to 1, with higher scores indicating greater similarity. The CBS is binary, with a score of 1 given for two phonetic representations that have a common initial and/or final biphone, and 0 for two strings that have neither in common.

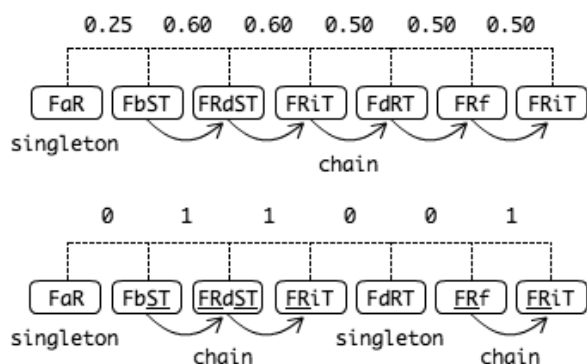


Figure 2: Phonetic chain and common-biphphone chain (below) for an example PVF response.

### 2.3 Phonetic Clustering

We distinguish between two ways of defining phonetic clusters. Traditionally, any sequence of  $n$  words in a PVF response is deemed to form a cluster if all pairwise word combinations for that sequence are determined to be phonetically similar by some metric. In addition to this method, we developed a less stringent approach in which we define *chains* instead of clusters.

A chain comprises a sequence for which the phonetic representation of each word is similar to that of the word immediately prior to it in the chain (unless it is chain-initial) and the word subsequent to it (unless it is chain-final). Lone words that do not belong to any cluster constitute *singleton* clusters. We call chains based on the edit-distance method *phonetic chains*, and chains based on the common-biphphone method *common-biphphone chains*; both are illustrated in Figure 2.

Unlike the binary CBS method, the PSS method produces continuous edit-distance values, and therefore requires a threshold for categorizing a word pair as similar or dissimilar. We determine the threshold empirically for each letter by taking a random sample of 1000 words starting with that letter in CMUdict, computing PSS scores for each pairwise combination ( $n = 499,500$ ), and then setting the threshold as the value separating the upper quintile of these scores. With the common-biphphone method, two words are considered phonetically similar simply if their CBS is 1.

### 2.4 System Overview

Our system is written in Python, and is available online.<sup>1</sup> The system accepts transcriptions of a

<sup>1</sup><http://rxinformatics.umn.edu/downloads.html>

PVF response for a specific letter and, as a pre-processing step, removes any words that do not begin with that letter. After pre-processing, all words are phoneticized by dictionary lookup in our modified CMUdict. For out-of-dictionary words, we automatically generate a phonetic representation with a decision tree-based grapheme-to-phoneme algorithm trained on the CMUdict (Pagel et al., 1998).

Next, PSSs and CBSs are computed sequentially for each pair of contiguous phonetic representations, and are used in their respective methods to compute the following measures: mean pairwise similarity score (MPSS), mean chain length (MCL), and maximum chain length (MXCL). Singletons are included in these calculations as chains of length 1.

We also calculate equivalent measures for clusters, but do not present these results here due to space limitations, as they are similar to those for chains. In addition to these measures, our system produces a count of the total number of words that start with the letter specified for the PVF test (WCNT), and a count of repeated words (RCNT).

## 3 Pilot Study

### 3.1 Participants

We used PVF tests from 55 boxers and mixed martial artists (4 women, 51 men; mean age 27.7 y.o., SD 6.0) that participated in the Professional Fighters Brain Health Study (PFBH). The PFBH is a longitudinal study of unarmed active professional fighters, retired professional fighters, and age/education matched controls (Bernick et al., in press). It is designed to enroll over 400 participants over the next five years. The 55 participants in our pilot represent a sample from the first wave of assessments, conducted in summer of 2012. All 55 participants were fluent speakers of English and were able to read at at least a 4th-grade level. None of these participants fought in a professional or amateur competition within 45 days prior to testing.

### 3.2 Methods

Each participant's professional fighting history was used to determine his or her total number of pro fights and number of fights per year. These figures were used to construct a composite fight-exposure index as a summary measure of cumulative traumatic exposure, as follows.

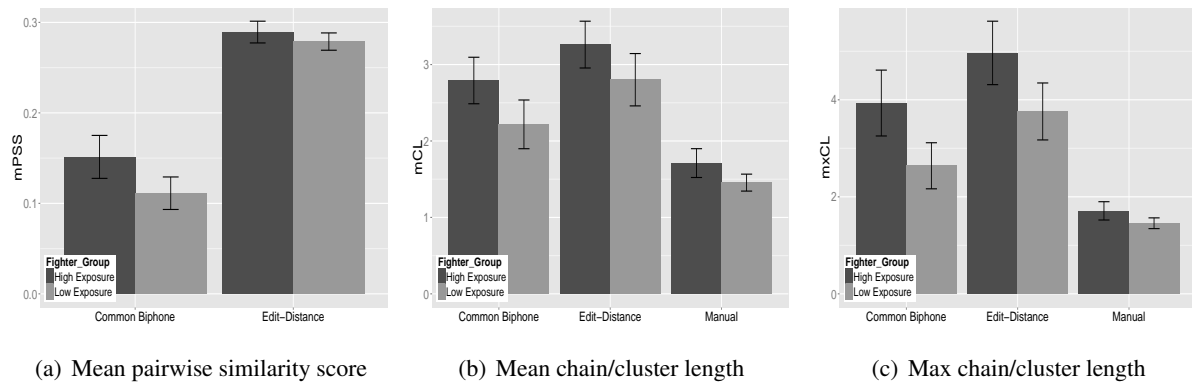


Figure 3: Computation-method and exposure-group comparisons showing significant differences between the low- and high-exposure fighter groups on MPSS, MCL, and MXCL measures. Error bars represent 95% confidence intervals around the means.

Fighters with zero professional fights were assigned a score of 0; fighters with between 1 and 15 total fights, but only one or fewer fights per year, were assigned a score of 1; fighters with 1-15 total fights, and more than one fight per year, got a score of 2; fighters with more than 15 total fights, but only one or fewer fights per year, got a score of 3; remaining fighters, with more than 15 fights and more than one fight per year, were assigned the highest score of 4.

Due to the relatively small sample size in our pilot study, we combined groups with scores of 0 and 1 to constitute the *low-exposure* group ( $n = 25$ ), and the rest were assigned to the *high-exposure* group ( $n = 30$ ).

All participants underwent a cognitive test battery that included the PVF test (letter ‘F’). Their responses were processed by our system, and means for our chaining variables of interest, as well as counts of total words and repetitions, were compared across the low- and high-exposure groups. Additionally, all 55 PVF responses were subjected to *manual* phonetic clustering analysis, following the methodology of Troyer et al. (1997). With this approach, clusters are used instead of chains, and two words are considered phonetically similar if they meet any of the following conditions: they begin with the same two orthographic letters; they rhyme; they differ by only a vowel sound (e.g., *flip* and *flop*); or they are homophones.

For each clustering method, the differences in means between the groups were tested for statistical significance using one-way ANOVA adjusted for the effects of age and years of education. Spearman correlation was used to test for associ-

ations between continuous variables, due to non-linearity, and to directly compare manually determined clustering measures with corresponding automatically determined chain measures.

## 4 Results

The results of comparisons between the clustering methods, as well as between the low- and high-exposure groups, are illustrated in Figure 3.<sup>2</sup>

We found a significant difference ( $p < 0.02$ ) in MPSS between the high- and low-exposure groups using the common-biphone method (0.15 vs. 0.11), while with edit distance the difference was small (0.29 vs. 0.28) and not significant (Figure 3a). Due to infeasibility, MPSS was not calculated manually.

Mean chain sizes determined by the common-biphone method correlated with manually determined cluster sizes more strongly than did chain sizes determined by edit distance ( $\rho = 0.73$ ,  $p < 0.01$  vs.  $\rho = 0.48$ ,  $p < 0.01$ ). Comparisons of maximum chain and cluster sizes showed a similar pattern ( $\rho = 0.71$ ,  $p < 0.01$  vs.  $\rho = 0.39$ ,  $p < 0.01$ ).

Both automatic methods showed significant differences ( $p < 0.01$ ) between the two groups in MCL and MXCL, with each finding longer chains in the high-exposure group (Figure 3b, 3c); however, slightly larger differences were observed using the common-biphone method (MCL: 2.79 vs. 2.21 by common-biphone method, 3.23 vs. 2.80 by edit-distance method; MXCL: 3.94 vs. 2.64 by

<sup>2</sup>Clustering measures rely on chains for our automatic methods, and on clusters for manual analysis.

common biphone, 4.94 vs. 3.76 by edit distance). Group differences for manually determined MCL and MXCL were also significant ( $p < 0.05$  and  $p < 0.02$ , respectively), but less so (MCL: 1.71 vs. 1.46; MXCL: 4.0 vs. 3.04).

## 5 Discussion

While manual phonetic clustering analysis yielded significant differences between the low- and high-exposure fighter groups, our automatic approach, which utilizes phonetic word representations, appears to be more sensitive to these differences; it also appears to produce less variability on clustering measures. Furthermore, as discussed above, automatic analysis is much less labor-intensive, and thus is more scalable to large numbers of tests. Moreover, our system is not prone to human error during analysis, nor to inter-rater variability.

Of the two automatic clustering methods, the common-biphone method, which uses binary similarity values, found greater differences between groups in MPSS, MCL, and MXCL; thus, it appears to be more sensitive than the edit-distance method in detecting group differences. Common-biphone measures were also found to better correlate with manual measures; however, both automated methods disagreed with the manual approach to some extent. The fact that the automated common-biphone method shows significant differences between group means, while having less variability in measurements, suggests that it may be a more suitable measure of phonetic clustering than the traditional manual method.

These results are particularly important in light of the difference in WCNT means between low- and high-exposure groups being small and not significant (WCNT: 17.6, SD 5.1 vs. 18.7, SD 4.7;  $p = 0.24$ ). Other studies that used manual clustering and switching analyses reported significantly more switches for healthy controls than for individuals with neurological conditions (Troyer et al., 1997). These studies also reported differences in the total number of words produced, likely due to investigating already impaired individuals.

Our findings show that the low- and high-exposure groups produced similar numbers of words, but the high-exposure group tended to produce longer sequences of phonetically similar words. The latter phenomenon may be interpreted as a mild form of perseverative (*stuck-in-set/repetitive*) behavior that is characteristic of dis-

orders involving damage to frontal and subcortical brain structures.

To test this interpretation, we correlated MCL and MXCL, the two measures with greatest differences between low- and high-exposure fighters, with the count of repeated words (RCNT). The resulting correlations were 0.41 ( $p = 0.01$ ) and 0.48 ( $p < 0.001$ ), respectively, which supports the perseverative-behavior interpretation of our findings.

Clearly, these findings are preliminary and need to be confirmed in larger samples; however, they plainly demonstrate the utility of our fully automated and quantifiable approach to characterizing and measuring clustering behavior on PVF tests. Pending further clinical validation, this system may be used for large-scale screening for subtle signs of certain types of brain damage or degeneration not only in contact-sports athletes, but also in the general population.

## 6 Acknowledgements

We thank the anonymous reviewers for their insightful feedback.

## References

- Atsushi Fujii and Tetsuya Ishikawa. 2001. Japanese/English cross-language information retrieval: Exploration of query translation and transliteration. In *Computers and the Humanities* 35.4.
- A.L. Benton, K.D. Hamsher, and A.B. Sivan. 1989. Multilingual aphasia examination.
- C. Bernick, S.J. Banks, S. Jones, W. Shin, M. Phillips, M. Lowe, M. Modic. In press. Professional Fighters Brain Health Study: Rationale and methods. In *American Journal of Epidemiology*.
- Brandon E. Gavett, Robert A. Stern, and Ann C. McKee. 2011. Chronic traumatic encephalopathy: A potential late effect of sport-related concussive and subconcussive head trauma. In *Clinics in Sports Medicine* 30, no. 1.
- Aileen K. Ho, Barbara J. Sahakian, Trevor W. Robbins, Roger A. Barker, Anne E. Rosser, and John R. Hodges. 2002. Verbal fluency in Huntington's disease: A longitudinal analysis of phonemic and semantic clustering and switching. In *Neuropsychologia* 40, no. 8.
- Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, vol. 10.



- 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*. Association for Computational Linguistics.
- Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. In *Computational Linguistics 24.4*.
- Grzegorz Kondrak, Daniel Marcu, and Kevin Knight. 2003. Cognates can improve statistical translation models. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003*. Association for Computational Linguistics.
- I. Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. In *Computational Linguistics 25.1*.
- Bennet I. Omalu, Julian Bailes, Jennifer Lynn Hammers, and Robert P. Fitzsimmons. 2010. Chronic traumatic encephalopathy, suicides and parasuicides in professional American athletes: The role of the forensic pathologist. In *The American Journal of Forensic Medicine and Pathology 31, no. 2*.
- Vincent Pagel, Kevin Lenzo, and Alan Black. 1998. Letter to sound rules for accented lexicon compression.
- Lawrence Rabiner and Bing-Hwang Juang. 1993. Fundamentals of speech recognition.
- Anand Raman, John Newman, and Jon Patrick. 1997. A complexity measure for diachronic Chinese phonology. In *Proceedings of the SIGPHON97 Workshop on Computational Linguistics at the ACL97/EACL97*.
- Sarah A. Raskin, Martin Sliwinski, and Joan C. Borod. 1992. Clustering strategies on tasks of verbal fluency in Parkinson's disease. In *Neuropsychologia 30, no. 1*.
- Sarah A. Raskin and Elizabeth Rearick. 1996. Verbal fluency in individuals with mild traumatic brain injury. In *Neuropsychology 10, no. 3*.
- Harold L. Somers. 1998. Similarity metrics for aligning children's articulation data. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics.
- Kristina Toutanova and Robert C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Angela K. Troyer, Morris Moscovitch, and Gordon Winocur. 1997. Clustering and switching as two components of verbal fluency: Evidence from younger and older healthy adults. In *Neuropsychology, 11*.
- Angela K. Troyer, Morris Moscovitch, Gordon Winocur, Michael P. Alexander, and Don Stuss. 1998a. Clustering and switching on verbal fluency: The effects of focal frontal- and temporal-lobe lesions. In *Neuropsychologia*.
- Angela K. Troyer, Morris Moscovitch, Gordon Winocur, Larry Leach, and Morris Freedman. 1998b. Clustering and switching on verbal fluency tests in Alzheimer's and Parkinson's disease. In *Journal of the International Neuropsychological Society 4, no. 2*.
- Robert Weide. 2008. Carnegie Mellon Pronouncing Dictionary, v. 0.7a. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.

# A New Set of Norms for Semantic Relatedness Measures

**Sean Szumlanski**

Department of EECS  
University of Central Florida  
seansz@cs.ucf.edu

**Fernando Gomez**

Department of EECS  
University of Central Florida  
gomez@eeecs.ucf.edu

**Valerie K. Sims**

Department of Psychology  
University of Central Florida  
Valerie.Sims@ucf.edu

## Abstract

We have elicited human quantitative judgments of semantic relatedness for 122 pairs of nouns and compiled them into a new set of relatedness norms that we call Rel-122. Judgments from individual subjects in our study exhibit high average correlation to the resulting relatedness means ( $r = 0.77$ ,  $\sigma = 0.09$ ,  $N = 73$ ), although not as high as Resnik's (1995) upper bound for expected average human correlation to similarity means ( $r = 0.90$ ). This suggests that human perceptions of relatedness are less strictly constrained than perceptions of similarity and establishes a clearer expectation for what constitutes human-like performance by a computational measure of semantic relatedness.

We compare the results of several WordNet-based similarity and relatedness measures to our Rel-122 norms and demonstrate the limitations of WordNet for discovering general indications of semantic relatedness. We also offer a critique of the field's reliance upon similarity norms to evaluate relatedness measures.

## 1 Introduction

Despite the well-established technical distinction between semantic similarity and relatedness (Agirre et al., 2009; Budanitsky and Hirst, 2006; Resnik, 1995), comparison to established similarity norms from psychology remains part of the standard evaluative procedure for assessing computational measures of semantic relatedness. Because similarity is only one particular type of relatedness, comparison to similarity norms fails to give a complete view of a relatedness measure's efficacy.

In keeping with Budanitsky and Hirst's (2006) observation that "comparison with human judgments is the ideal way to evaluate a measure of similarity or relatedness," we have undertaken the creation of a new set of relatedness norms.

## 2 Background

The similarity norms of Rubenstein and Goode-nough (1965; henceforth R&G) and Miller and Charles (1991; henceforth M&C) have seen ubiquitous use in evaluation of computational measures of semantic similarity and relatedness.

R&G established their similarity norms by presenting subjects with 65 slips of paper, each of which contained a pair of nouns. Subjects were directed to read through all 65 noun pairs, then sort the pairs "according to amount of 'similarity of meaning.'" Subjects then assigned similarity scores to each pair on a scale of 0.0 (completely dissimilar) to 4.0 (strongly synonymous).

The R&G results have proven to be highly replicable. M&C repeated R&G's study using a subset of 30 of the original word pairs, and their resulting similarity norms correlated to the R&G norms at  $r = 0.97$ . Resnik's (1995) subsequent replication of M&C's study similarly yielded a correlation of  $r = 0.96$ . The M&C pairs were also included in a similarity study by Finkelstein et al. (2002), which yielded correlation of  $r = 0.95$  to the M&C norms.

### 2.1 WordSim353

WordSim353 (Finkelstein et al., 2002) has recently emerged as a potential surrogate dataset for evaluating relatedness measures. Several studies have reported correlation to WordSim353 norms as part of their evaluation procedures, with some studies explicitly referring to it as a collection of human-assigned relatedness scores (Gabrilovich and Markovitch, 2007; Hughes and Ramage, 2007; Milne and Witten, 2008).

Yet, the instructions presented to Finkelstein et al.'s subjects give us pause to reconsider WordSim353's classification as a set of relatedness norms. They repeatedly framed the task as one in which subjects were expected to assign word similarity scores, although participants were instructed to extend their definition of similarity to include antonymy, which perhaps explains why the authors later referred to their data as "relatedness" norms rather than merely "similarity" norms.

Jarmasz and Szpakowicz (2003) have raised further methodological concerns about the construction of WordSim353, including: (a) similarity was rated on a scale of 0.0 to 10.0, which is intrinsically more difficult for humans to manage than the scale of 0.0 to 4.0 used by R&G and M&C, and (b) the inclusion of proper nouns introduced an element of cultural bias into the dataset (e.g., the evaluation of the pair *Arafat-terror*).

Cognizant of the problematic conflation of similarity and relatedness in WordSim353, Agirre et al. (2009) partitioned the data into two sets: one containing noun pairs exhibiting similarity, and one containing pairs of related but dissimilar nouns. However, pairs in the latter set were not assessed for scoring distribution validity to ensure that strongly related word pairs were not penalized by human subjects for being dissimilar.<sup>1</sup>

### 3 Methodology

In our experiments, we elicited human ratings of semantic relatedness for 122 noun pairs. In doing so, we followed the methodology of Rubenstein and Goodenough (1965) as closely as possible: participants were instructed to read through a set of noun pairs, sort them by how strongly related they were, and then assign each pair a relatedness score on a scale of 0.0 ("completely unrelated") to 4.0 ("very strongly related").

We made two notable modifications to the experimental procedure of Rubenstein and Goodenough. First, instead of asking participants to judge "amount of 'similarity of meaning,'" we asked them to judge "how closely related in meaning" each pair of nouns was. Second, we used a Web interface to collect data in our study; instead of reordering a deck of cards, participants were presented with a grid of cards that they were able

<sup>1</sup>Perhaps not surprisingly, the highest scores in WordSim353 (all ratings from 9.0 to 10.0) were assigned to pairs that Agirre et al. placed in their similarity partition.

to rearrange interactively with the use of a mouse or any touch-enabled device, such as a tablet PC.<sup>2</sup>

### 3.1 Experimental Conditions

Each participant in our study was randomly assigned to one of four conditions. Each condition contained 32 noun pairs for evaluation.

Of those pairs, 10 were randomly selected from from WordNet++ (Ponzetto and Navigli, 2010) and 10 from SGN (Szumlanski and Gomez, 2010)—two semantic networks that categorically indicate strong relatedness between WordNet noun senses. 10 additional pairs were generated by randomly pairing words from a list of all nouns occurring in Wikipedia. The nouns in the pairs we used from each of these three sources were matched for frequency of occurrence in Wikipedia.

We manually selected two additional pairs that appeared across all four conditions: *leaves-rake* and *lion-cage*. These control pairs were included to ensure that each condition contained examples of strong semantic relatedness, and potentially to help identify and eliminate data from participants who assigned random relatedness scores. Within each condition, the 32 word pairs were presented to all subjects in the same random order. Across conditions, the two control pairs were always presented in the same positions in the word pair grid.

Each word pair was subjected to additional scrutiny before being included in our dataset. We eliminated any pairs falling into one or more of the following categories: (a) pairs containing proper nouns, (b) pairs in which one or both nouns might easily be mistaken for adjectives or verbs, (c) pairs with advanced vocabulary or words that might require domain-specific knowledge in order to be properly evaluated, and (d) pairs with shared stems or common head nouns (e.g., *first cousin-second cousin* and *sinner-sinning*). The latter were eliminated to prevent subjects from latching onto superficial lexical commonalities as indicators of strong semantic relatedness without reflecting upon meaning.

### 3.2 Participants

Participants in our study were recruited from introductory undergraduate courses in psychology and computer science at the University of Central Florida. Students from the psychology courses

<sup>2</sup>Online demo: <http://www.cs.ucf.edu/~seansz/rel-122>

participated for course credit and accounted for 89% of respondents.

92 participants provided data for our study. Of these, we identified 19 as outliers, and their data were excluded from our norms to prevent interference from individuals who appeared to be assigning random scores to noun pairs. We considered an outlier to be any individual whose numeric ratings fell outside two standard deviations from the means for more than 10% of the word pairs they evaluated (i.e., at least four word pairs, since each condition contained 32 word pairs).

For outlier detection, means and standard deviations were computed using leave-one-out sampling. That is, data from individual  $J$  were not incorporated into means or standard deviations when considering whether to eliminate  $J$  as an outlier.<sup>3</sup>

Of the 73 participants remaining after outlier elimination, there was a near-even split between males (37) and females (35), with one individual declining to provide any demographic data. The average age of participants was 20.32 ( $\sigma = 4.08$ ,  $N = 72$ ). Most students were freshmen (49), followed in frequency by sophomores (16), seniors (4), and juniors (3). Participants earned an average score of 42% on a standardized test of advanced vocabulary ( $\sigma = 16\%$ ,  $N = 72$ ) (Test I – V-4 from Ekstrom et al. (1976)).

## 4 Results

Each word pair in Rel-122 was evaluated by at least 20 human subjects. After outlier removal (described above), each word pair retained evaluations from 14 to 22 individuals. The resulting relatedness means are available online.<sup>4</sup>

An excerpt of the Rel-122 norms is shown in Table 1. We note that the highest rated pairs in our dataset are not strictly similar entities; exactly half of the 10 most strongly related nouns in Table 1 are dissimilar (e.g., *digital camera–photographer*).

Judgments from individual subjects in our study exhibited high average correlation to the elicited relatedness means ( $r = 0.769$ ,  $\sigma = 0.09$ ,  $N = 73$ ). Resnik (1995), in his replication of the

<sup>3</sup>We used this sampling method to prevent extreme outliers from masking their own aberration during outlier detection, which is potentially problematic when dealing with small populations. Without leave-one-out-sampling, we would have identified fewer outliers (14 instead of 19), but the resulting means would still have correlated strongly to our final relatedness norms ( $r = 0.991$ ,  $p < 0.01$ ).

<sup>4</sup><http://www.cs.ucf.edu/~seansz/rel-122>

| #    | Word Pair      |                | $\mu$ |
|------|----------------|----------------|-------|
| 1.   | underwear      | lingerie       | 3.94  |
| 2.   | digital camera | photographer   | 3.85  |
| 3.   | tuition        | fee            | 3.85  |
| 4.   | leaves         | rake           | 3.82  |
| 5.   | symptom        | fever          | 3.79  |
| 6.   | fertility      | ovary          | 3.78  |
| 7.   | beef           | slaughterhouse | 3.78  |
| 8.   | broadcast      | commentator    | 3.75  |
| 9.   | apparel        | jewellery      | 3.72  |
| 10.  | arrest         | detention      | 3.69  |
|      | ...            |                |       |
| 122. | gladiator      | plastic bag    | 0.13  |

Table 1: Excerpt of Rel-122 norms.

M&C study, reported average individual correlation of  $r = 0.90$  ( $\sigma = 0.07$ ,  $N = 10$ ) to similarity means elicited from a population of 10 graduate students and postdoctoral researchers. Presumably Resnik’s subjects had advanced knowledge of what constitutes semantic similarity, as he established  $r = 0.90$  as an upper bound for expected human correlation on that task.

The fact that average human correlation in our study is weaker than in previous studies suggests that human perceptions of relatedness are less strictly constrained than perceptions of similarity, and that a reasonable computational measure of relatedness might only approach a correlation of  $r = 0.769$  to relatedness norms.

In Table 2, we present the performance of a variety of relatedness and similarity measures on our new set of relatedness means.<sup>5</sup> Coefficients of correlation are given for Pearson’s product-moment correlation ( $r$ ), as well as Spearman’s rank correlation ( $\rho$ ). For comparison, we include results for the correlation of these measures to the M&C and R&G similarity means.

The generally weak performance of the WordNet-based measures on this task is not surprising, given WordNet’s strong disposition toward codifying semantic similarity, which makes it an impoverished resource for discovering general semantic relatedness. We note that the three WordNet-based measures from Table 2 that are regarded in the literature as relatedness measures (Banerjee and Pedersen, 2003; Hirst and St-Onge, 1998; Patwardhan and Pedersen, 2006)

<sup>5</sup>Results based on standard implementations in the WordNet::Similarity Perl module of Pedersen et al. (2004) (v2.05).

| Measure                          | Rel-122      |              | M&C          |              | R&G          |              |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                  | $r$          | $\rho$       | $r$          | $\rho$       | $r$          | $\rho$       |
| * Szumlanski and Gomez (2010)    | <b>0.654</b> | <b>0.534</b> | 0.852        | 0.859        | 0.824        | <b>0.841</b> |
| * Patwardhan and Pedersen (2006) | 0.341        | 0.364        | <b>0.865</b> | <b>0.906</b> | 0.793        | 0.795        |
| Path Length                      | 0.225        | 0.183        | 0.755        | 0.715        | 0.784        | 0.783        |
| * Banerjee and Pedersen (2003)   | 0.210        | 0.258        | 0.356        | 0.804        | 0.340        | 0.718        |
| Resnik (1995)                    | 0.203        | 0.182        | 0.806        | 0.741        | 0.822        | 0.757        |
| Jiang and Conrath (1997)         | 0.188        | 0.133        | 0.473        | 0.663        | 0.575        | 0.592        |
| Leacock and Chodorow (1998)      | 0.173        | 0.167        | 0.779        | 0.715        | <b>0.839</b> | 0.783        |
| Wu and Palmer (1994)             | 0.187        | 0.180        | 0.764        | 0.732        | 0.797        | 0.768        |
| Lin (1998)                       | 0.145        | 0.148        | 0.739        | 0.687        | 0.726        | 0.636        |
| * Hirst and St-Onge (1998)       | 0.141        | 0.160        | 0.667        | 0.782        | 0.726        | 0.797        |

Table 2: Correlation of similarity and relatedness measures to Rel-122, M&C, and R&G. Starred rows (\*) are considered relatedness measures. All measures are WordNet-based, except for the scoring metric of Szumlanski and Gomez (2010), which is based on lexical co-occurrence frequency in Wikipedia.

| #   | Noun Pair |            | Sim. | Rel. | #   | Noun Pair |           | Sim. | Rel. |
|-----|-----------|------------|------|------|-----|-----------|-----------|------|------|
| 1.  | car       | automobile | 3.92 | 4.00 | 16. | lad       | brother   | 1.66 | 2.68 |
| 2.  | gem       | jewel      | 3.84 | 3.98 | 17. | journey   | car       | 1.16 | 3.00 |
| 3.  | journey   | voyage     | 3.84 | 3.97 | 18. | monk      | oracle    | 1.10 | 2.54 |
| 4.  | boy       | lad        | 3.76 | 3.97 | 19. | cemetery  | woodland  | 0.95 | 1.69 |
| 5.  | coast     | shore      | 3.70 | 3.97 | 20. | food      | rooster   | 0.89 | 2.59 |
| 6.  | asylum    | madhouse   | 3.61 | 3.91 | 21. | coast     | hill      | 0.87 | 1.59 |
| 7.  | magician  | wizard     | 3.50 | 3.58 | 22. | forest    | graveyard | 0.84 | 2.01 |
| 8.  | midday    | noon       | 3.42 | 4.00 | 23. | shore     | woodland  | 0.63 | 1.63 |
| 9.  | furnace   | stove      | 3.11 | 3.67 | 24. | monk      | slave     | 0.55 | 1.31 |
| 10. | food      | fruit      | 3.08 | 3.91 | 25. | coast     | forest    | 0.42 | 1.89 |
| 11. | bird      | cock       | 3.05 | 3.71 | 26. | lad       | wizard    | 0.42 | 2.12 |
| 12. | bird      | crane      | 2.97 | 3.96 | 27. | chord     | smile     | 0.13 | 0.68 |
| 13. | tool      | implement  | 2.95 | 2.86 | 28. | glass     | magician  | 0.11 | 1.30 |
| 14. | brother   | monk       | 2.82 | 2.89 | 29. | rooster   | voyage    | 0.08 | 0.63 |
| 15. | crane     | implement  | 1.68 | 0.90 | 30. | noon      | string    | 0.08 | 0.14 |

Table 3: Comparison of relatedness means to M&C similarity means. Correlation is  $r = 0.91$ .

have been hampered by their reliance upon WordNet. The disparity between their performance on Rel-122 and the M&C and R&G norms suggests the shortcomings of using similarity norms for evaluating measures of relatedness.

## 5 (Re-)Evaluating Similarity Norms

After establishing our relatedness norms, we created two additional experimental conditions in which subjects evaluated the relatedness of noun pairs from the M&C study. Each condition again had 32 noun pairs: 15 from M&C and 17 from Rel-122. Pairs from M&C and Rel-122 were uniformly distributed between these two new condi-

tions based on matched normative similarity or relatedness scores from their respective datasets.

Results from this second phase of our study are shown in Table 3. The correlation of our relatedness means on this set to the similarity means of M&C was strong ( $r = 0.91$ ), but not as strong as in replications of the study that asked subjects to evaluate similarity (e.g.  $r = 0.96$  in Resnik’s (1995) replication and  $r = 0.95$  in Finkelstein et al.’s (2002) M&C subset).

That the synonymous M&C pairs garner high relatedness ratings in our study is not surprising; strong similarity is, after all, one type of strong relatedness. The more interesting result from

our study, shown in Table 3, is that relatedness norms for pairs that are related but dissimilar (e.g., *journey-car* and *forest-graveyard*) deviate significantly from established similarity norms. This indicates that asking subjects to evaluate “similarity” instead of “relatedness” can significantly impact the norms established in such studies.

## 6 Conclusions

We have established a new set of relatedness norms, Rel-122, that is offered as a supplementary evaluative standard for assessing semantic relatedness measures.

We have also demonstrated the shortcomings of using similarity norms to evaluate such measures. Namely, since similarity is only one type of relatedness, comparison to similarity norms fails to provide a complete view of a measure’s ability to capture more general types of relatedness. This is particularly problematic when evaluating WordNet-based measures, which naturally excel at capturing similarity, given the nature of the WordNet ontology.

Furthermore, we have found that asking judges to evaluate “relatedness” of terms, rather than “similarity,” has a substantive impact on resulting norms, particularly with respect to the M&C similarity dataset. Correlation of individual judges’ ratings to resulting means was also significantly lower on average in our study than in previous studies that focused on similarity (e.g., Resnik, 1995). These results suggest that human perceptions of relatedness are less strictly constrained than perceptions of similarity and validate the need for new relatedness norms to supplement existing gold standard similarity norms in the evaluation of relatedness measures.

## 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 *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 19–27.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 805–810.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Ruth B. Ekstrom, John W. French, Harry H. Harman, and Diran Dermen. 1976. *Manual for Kit of Factor-Referenced Cognitive Tests*. Educational Testing Service, Princeton, NJ.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems (TOIS)*, 20(1):116–131.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Thad Hughes and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 581–589, Prague, Czech Republic, June. Association for Computational Linguistics.

Mario Jarmasz and Stan Szpakowicz. 2003. Roget’s thesaurus and semantic similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 212–219.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics (ROCLING)*, pages 19–33.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–283. MIT Press.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, pages 296–304.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

- David Milne and Ian H. Witten. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI)*, pages 25–30.
- Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics Workshop on Making Sense of Sense*, pages 1–8.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – Measuring the relatedness of concepts. In *Proceedings of the 5th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 38–41.
- Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1522–1531.
- Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453.
- Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
- Sean Szumlanski and Fernando Gomez. 2010. Automatically acquiring a semantic network of related concepts. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 19–28.
- Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 133–139.





# Author Index

- Abu-Jbara, Amjad, 572, 829  
Addanki, Karteek, 375  
Adel, Heike, 206  
Agić, Željko, 784  
Ahn, Byung-Gyu, 406  
Alexe, Bogdan, 804  
Almeida, Miguel, 617  
Aly, Mohamed, 494  
Ambati, Bharat Ram, 604  
Ananthakrishnan, Sankaranarayanan, 697  
Andreas, Jacob, 47  
Andrews, Nicholas, 63  
Androutopoulos, Ion, 561  
Arase, Yuki, 238  
Ashwell, Peter, 516  
Atiya, Amir, 494  
Augenstein, Isabelle, 289
- Bai, Yalong, 312  
Baird, Henry S., 479  
Bakhshaei, Somayeh, 318  
Baldwin, Tyler, 804  
Banks, Sarah, 884  
Baroni, Marco, 53  
Bazrafshan, Marzieh, 419  
Beck, Daniel, 543  
Bedini, Claudia, 92  
Bekki, Daisuke, 273  
Bel, Núria, 725  
Belinkov, Yonatan, 1  
Beller, Charley, 63  
Bellot, Patrice, 148  
Beloborodov, Alexander, 262  
Ben, Guosheng, 382  
Berg, Alexander, 790  
Berg, Tamara, 790  
Bergen, Leon, 115  
Bergsma, Shane, 866  
Bernardi, Raffaella, 53  
Bernick, Charles, 884  
Bertomeu Castelló, Núria, 92  
Bhatt, Brijesh, 268  
Bhattacharyya, Pushpak, 268, 346, 538, 860  
Bhingardive, Sudha, 538
- Biran, Or, 69  
Bird, Steven, 634  
Blomqvist, Eva, 289  
Boella, Guido, 532  
Bouamor, Dhouha, 759  
Braslavski, Pavel, 262
- Cai, Shu, 748  
Callison-Burch, Chris, 63, 702  
Carbonell, Jaime, 765  
Cardie, Claire, 217  
Carl, Michael, 346  
Cha, Young-rok, 201  
Chang, Kai-Chun, 446  
Chao, Lidia S., 171  
Chen, Emily, 143  
Chen, Hsin-Hsi, 446  
Chen, Junwen, 58  
Chen, Lei, 278  
Chen, Ruey-Cheng, 166  
Chen, Zheng, 810  
Chng, Eng Siong, 233  
Choi, Yejin, 790  
Choi, Yoonjung, 120  
Chong, Tze Yuang, 233  
Choudhury, Pallavi, 222  
Cimiano, Philipp, 848  
Ciravegna, Fabio, 289  
Clark, Jonathan H., 690  
Clark, Peter, 159, 702  
Clark, Stephen, 47  
Cleuziou, Guillaume, 153  
Cohen, William W., 35  
Cohn, Trevor, 543  
Collet, Christophe, 328  
Conroy, John M., 131  
Cook, Paul, 634  
Coppola, Greg, 610  
Cui, Lei, 340  
Curiel, Arturo, 328  
Curran, James R., 98, 516, 671
- Dagan, Ido, 283, 451  
Daland, Robert, 873

Dang, Hoa Trang, 131  
Darwish, Kareem, 1  
Das, Dipanjan, 92  
Dasigi, Pradeep, 549  
de Souza, José G.C., 771  
Deng, Lingjia, 120  
Deng, Xiaotie, 24  
Deng, Zhi-Hong, 855  
Deoskar, Tejaswini, 604  
Derczynski, Leon, 645  
Deveaud, Romain, 148  
DeYoung, Jay, 63  
Di Caro, Luigi, 532  
Diab, Mona, 456, 549, 829  
Dias, Gaël, 153  
Dinu, Georgiana, 53  
Doucet, Antoine, 243  
Dredze, Mark, 63  
Duan, Nan, 41, 424  
Duh, Kevin, 678  
Dunietz, Jesse, 765  
Duong, Long, 634  
Durrani, Nadir, 399  
Dyer, Chris, 777

E. Banchs, Rafael, 233  
El Kholy, Ahmed, 412  
Elfardy, Heba, 456  
Elluru, Naresh Kumar, 196  
Elluru, Raghavendra, 196  
Esplà-Gomis, Miquel, 771

Farkas, Richárd, 255  
Farra, Noura, 549  
Faruqui, Manaal, 777  
Finch, Andrew, 393  
Fossati, Marco, 742  
Frank, Stefan L., 878  
Fraser, Alexander, 399  
Fukumoto, Fumiyo, 474

Gaizauskas, Robert, 645  
Galli, Giulia, 878  
Ganchev, Kuzman, 92  
Ganesalingam, Mohan, 440  
Gao, Dehong, 567  
Gao, Wei, 58  
Garain, Utpal, 126  
Gentile, Anna Lisa, 289  
Georgi, Ryan, 306  
Gibson, Edward, 115  
Gilbert, Nathan, 81

Gildea, Daniel, 419  
Giuliano, Claudio, 742  
Glavaš, Goran, 797  
Goldberg, Yoav, 92, 628  
Goldberger, Jacob, 283  
Goldwasser, Dan, 462  
Gomez, Fernando, 890  
Govindaraju, Vidhya, 658  
Goyal, Kartik, 467  
Grishman, Ralph, 665  
Guo, Weiwei, 143  
Gurevych, Iryna, 451  
Guzmán, Francisco, 12

Habash, Nizar, 412, 549  
Habibi, Maryam, 651  
Hagan, Susan, 719  
Hagiwara, Masato, 183  
Hall, Keith, 92  
Hasan, Kazi Saidul, 816  
He, Yulan, 58  
He, Zhengyan, 30, 177  
Heafield, Kenneth, 690  
Herbelot, Aurélie, 440  
Hewavitharana, Sanjika, 697  
Hieber, Felix, 323  
Hirao, Tsutomu, 212  
Hoang, Hieu, 399  
Hoffmann, Raphael, 665  
Hovy, Eduard, 467  
Hu, Haifeng, 843  
Hu, Xuelei, 521  
Huang, Chu-ren, 511  
Huang, Fei, 387  
Huang, Hen-Hsen, 446  
Huang, Liang, 628  
Huang, Xuanjing, 434  
Hwang, Seung-won, 201

Iwata, Tomoharu, 212

Jang, Hyeju, 836  
Jauhar, Sujay Kumar, 467  
Jehl, Laura, 323  
Jha, Rahul, 249, 572  
Jiang, Minghu, 484  
Jiang, Wenbin, 591  
Jiang, Xiaorui, 822  
Jurafsky, Dan, 74, 499

Kaneko, Kimi, 273  
Khadivi, Shahram, 318  
Khalilov, Maxim, 262

Kim, Jinhan, 201  
King, Ben, 249, 829  
Klein, Dan, 98  
Klinger, Roman, 848  
Knight, Kevin, 748  
Koehn, Philipp, 352, 399, 690  
Komachi, Mamoru, 238, 708  
Koprinska, Irena, 516  
Korhonen, Anna, 736  
Kummerfeld, Jonathan K., 98  
Kuznetsova, Polina, 790

Labutov, Igor, 489  
Lampouras, Gerasimos, 561  
Langlais, Phillippe, 684  
Lavrenko, Victor, 753  
Lee, Jungmee, 92  
Leung, Cheung-Chi, 190  
Leusch, Gregor, 412  
Levin, Lori, 765  
Levy, Omer, 451  
Lewis, William D., 306  
Li, Binyang, 58  
Li, Haizhou, 190, 233  
Li, Huayi, 24  
Li, Huiying, 467  
Li, Jiwei, 217, 556  
Li, Li, 177  
Li, Miao, 278  
Li, Mu, 30, 340  
Li, Peifeng, 511  
Li, Qing, 24  
Li, Shiyngxue, 855  
Li, Shoushan, 511, 521  
Li, Sujian, 217, 556  
Li, Tingting, 393  
Li, Wenjie, 567  
Li, Xiang, 591  
Li, Yunyao, 804  
Ligozat, Anne-Laure, 429  
Lim, Lian Tze, 294  
Lim, Tek Yong, 294  
Lin, Shouxun, 358  
Lipson, Hod, 489  
Liu, Bing, 24  
Liu, Bingquan, 843  
Liu, Ding, 484  
Liu, Huanhuan, 511  
Liu, Ming, 843  
Liu, Qun, 358, 364, 382, 591  
Liu, Shujie, 30, 340  
Lo, Chi-kiu, 375

Lu, Xiaoming, 190  
Lü, Yajuan, 364, 382, 591

Ma, Bin, 190  
Ma, Ji, 110  
Ma, Xuezhe, 585  
Malu, Akshat, 860  
Mankoff, Robert, 249  
Marelli, Marco, 53  
Marino, Susan, 884  
Martínez Alonso, Héctor, 725  
Martins, Andre, 617  
Matsumoto, Yuji, 708  
Matsuyoshi, Suguru, 474  
Matthews, David, 228  
Matusov, Evgeny, 412  
McCarthy, Diana, 736  
McDonald, Ryan, 92  
McKeown, Kathleen, 69  
Mehay, Dennis, 697  
Melamud, Oren, 283  
Mishra, Abhijit, 346  
Miyao, Yusuke, 273  
Moran, Sean, 753  
Moreno, Jose G., 153  
Moschitti, Alessandro, 714  
Movshovitz-Attias, Dana, 35  
Mukherjee, Arjun, 24  
Murthy, Hema, 196

Nagata, Masaaki, 18, 212  
Nagy T., István, 255  
Nakov, Preslav, 12  
Nalisnick, Eric T., 479  
Natarajan, Prem, 697  
Nath, J. Saketha, 860  
Negri, Matteo, 771  
Nenkova, Ani, 131  
Neubig, Graham, 678  
Ng, Vincent, 816  
Nicosia, Massimo, 714  
Nivre, Joakim, 92

O'Donnell, Timothy J., 115  
O'Keefe, Tim, 516  
Ofrazier, Kemal, 719  
Ordonez, Vicente, 790  
Osborne, Miles, 753  
Otten, Leun J., 878

Padó, Sebastian, 731, 784  
Pakhomov, Serguei, 884  
Passonneau, Rebecca J., 143

Pecina, Pavel, 634  
Pendus, Cezar, 387  
Penstein Rosé, Carolyn, 836  
Perin, Dolores, 143  
Petrov, Slav, 92  
Petrović, Saša, 228  
Poddar, Lahari, 268  
Popescu-Belis, Andrei, 651  
Post, Matt, 866  
Potts, Christopher, 74  
Pouzyrevsky, Ivan, 690  
Prahallad, Kishore, 196

Qiu, Xipeng, 434  
Quirk, Chris, 7, 222  
Quirnbach-Brundage, Yvonne, 92

Radev, Dragomir, 249, 572, 829  
Radford, Will, 671  
Ramteke, Ankit, 860  
Ranaivo-Malançon, Bali, 294  
Rankel, Peter A., 131  
Razmara, Majid, 334  
Ré, Christopher, 658  
Reschke, Kevin, 499  
Riezler, Stefan, 323  
Riloff, Ellen, 81  
Roth, Dan, 462  
Roth, Ryan, 549  
Ryan, James O., 884

Sachan, Mrinmaya, 467  
Saers, Markus, 375  
Sajjad, Hassan, 1  
Sakaguchi, Keisuke, 238  
Salama, Ahmed, 719  
Salavati, Shahin, 300  
Sandford Pedersen, Bolette, 725  
SanJuan, Eric, 148  
Sarkar, Anoop, 334  
Sawaf, Hassan, 412  
Sawai, Yu, 708  
Schmid, Helmut, 399  
Schultz, Tanja, 206  
Sekine, Satoshi, 183  
Semmar, Nasredine, 759  
Senapati, Apurbalal, 126  
Severyn, Aliaksei, 714  
Søgaard, Anders, 640  
Shaikh, Samiulla, 538  
Sharoff, Serge, 262  
Sheykh Esmaili, Kyumars, 300

Shieber, Stuart M., 597  
Si, Jianfeng, 24  
Sims, Valerie K., 890  
Smith, Noah A., 617  
Šnajder, Jan, 731, 784, 797  
Snyder, Justin, 63  
Soon, Lay-Ki, 294  
Specia, Lucia, 543  
Srivastava, Shashank, 467  
Stanoi, Ioana R., 804  
Steedman, Mark, 604, 610  
Sudoh, Katsuhito, 678  
Sui, Zhifang, 810  
Sun, Lin, 736  
Sun, Meng, 364  
Sun, Ni, 177  
Sun, Xiaoping, 822  
Suzuki, Jun, 18  
Suzuki, Yoshimi, 474  
Szpektor, Idan, 283  
Szumlanski, Sean, 890

Täckström, Oscar, 92  
Tan, Jiwei, 87  
Tang, Enya Kong, 294  
Teng, Zhiyang, 382  
Tian, Hao, 312  
Tian, Le, 434  
Tofighi Zahabi, Samira, 318  
Toivanen, Jukka M., 243  
Toivonen, Hannu, 243  
Tomeh, Nadi, 549  
Tonelli, Sara, 742  
Toutanova, Kristina, 406  
Trancoso, Isabel, 171  
Tsarfaty, Reut, 578  
Tse, Daniel, 98  
Tsukada, Hajime, 678  
Tu, Mei, 370  
Tu, Zhaopeng, 358  
Turchi, Marco, 771

Vadapalli, Anandaswarup, 196  
Valitutti, Alessandro, 243  
Van Durme, Benjamin, 63, 159, 702  
Vigliocco, Gabriella, 878  
Vincze, Veronika, 255  
Vlachos, Andreas, 47  
Vogel, Adam, 74, 499  
Vogel, Stephan, 12  
Volkova, Svitlana, 505  
Vu, Ngoc Thang, 206

Wan, Xiaojun, 87, 526  
Wang, Baoxun, 843  
Wang, Chenguang, 41  
Wang, Houfeng, 30, 177  
Wang, Tao, 521  
Wang, Xiaolong, 843  
Wang, Zhiguo, 623  
Wang, Zhiyang, 364  
Weese, Jonathan, 63  
Wei, Zhongyu, 58  
Wen, Miaomiao, 836  
Wiebe, Janyce, 120  
Williams, Philip, 352  
Wilson, Theresa, 505  
Wisniewski, Guillaume, 137  
Wolfe, Travis, 63  
Wong, Derek F., 171  
Wong, Kam-Fai, 58  
Wu, Dekai, 375

Xia, Fei, 306, 585  
Xia, Rui, 521  
Xiang, Guang, 836  
Xiao, Jianguo, 87  
Xiao, Tong, 110  
Xie, Lei, 190  
Xiong, Deyi, 382  
Xu, Tan, 63  
Xu, Wei, 665  
Xu, Wenduan, 352  
Xue, Nianwen, 623

Yamangil, Elif, 597  
Yan, Jun, 810  
Yang, Nan, 110  
Yang, Xiaofang, 484  
Yang, Zhenxin, 278  
Yao, Xuchen, 63, 159, 702  
Yarowsky, David, 505  
You, Gae-won, 201  
Yu, Dianhai, 312  
Yu, Hongliang, 855  
Yu, Mo, 312

Zeller, Britta, 731  
Zeng, Junyu, 810  
Zeng, Xiaodong, 171  
Zesch, Torsten, 451  
Zhang, Ce, 658  
Zhang, Chunyue, 393  
Zhang, Dongdong, 340  
Zhang, Hao, 92  
Zhang, Jianwen, 810  
Zhang, Longkai, 30, 177  
Zhang, Ming, 41  
Zhang, Renxian, 567  
Zhang, Xingxing, 810  
Zhang, Yue, 352  
Zhang, Ziqi, 289  
Zhao, Jun, 104  
Zhao, Kai, 628  
Zhao, Le, 665  
Zhao, Tiejun, 312, 393  
Zheng, Zeyu, 836  
Zhou, Guangyou, 104  
Zhou, Guodong, 511  
Zhou, Lanjun, 58  
Zhou, Ming, 30, 41, 340  
Zhou, Yu, 370  
Zhu, Jingbo, 110  
Zhu, Zede, 278  
Zhuge, Hai, 822  
Zong, Chengqing, 370, 521, 623  
Zuraw, Kie, 873  
Zweigenbaum, Pierre, 759