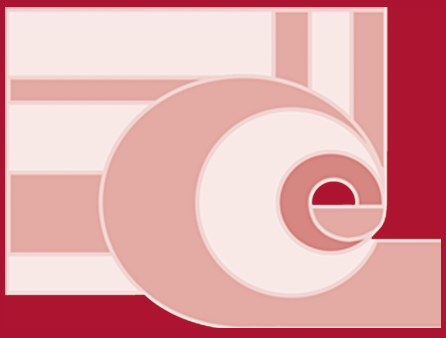
Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

LNCS 9041

16th International Conference, CICLing 2015
Cairo, Egypt, April 14–20, 2015
Proceedings, Part I

Part I



Springer

Lecture Notes in Computer Science **9041**

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

More information about this series at http://www.springer.com/series/7407

Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

16th International Conference, CICLing 2015
Cairo, Egypt, April 14–20, 2015
Proceedings, Part I

Springer

Editor
Alexander Gelbukh
Centro de Investigación en Computación
Instituto Politécnico Nacional
Mexico DF
Mexico

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-18110-3 ISBN 978-3-319-18111-0 (eBook)
DOI 10.1007/978-3-319-18111-0

Library of Congress Control Number: 2015936674

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

CICLing 2015 was the 16^{th} Annual Conference on Intelligent Text Processing and Computational Linguistics. The CICLing conferences provide a wide-scope forum for discussion of the art and craft of natural language processing research, as well as the best practices in its applications.

This set of two books contains four invited papers and a selection of regular papers accepted for presentation at the conference. Since 2001, the proceedings of the CICLing conferences have been published in Springer's *Lecture Notes in Computer Science* series as volumes 2004, 2276, 2588, 2945, 3406, 3878, 4394, 4919, 5449, 6008, 6608, 6609, 7181, 7182, 7816, 7817, 8403, and 8404.

The set has been structured into 13 sections representative of the current trends in research and applications of Natural Language Processing:

- Lexical Resources
- Morphology and Chunking
- Syntax and Parsing
- Anaphora Resolution and Word Sense Disambiguation
- Semantics and Dialogue
- Machine Translation and Multilingualism
- Sentiment Analysis and Emotions
- Opinion Mining and Social Network Analysis
- Natural Language Generation and Summarization
- Information Retrieval, Question Answering, and Information Extraction
- Text Classification
- Speech Processing
- Applications

The 2015 event received submissions from 62 countries, a record high number in the 16-year history of the CICLing series. A total of 329 papers (second highest number in the history of CICLing) by 705 authors were submitted for evaluation by the International Program Committee; see Figure 1 and Tables 1 and 2. This two-volume set contains revised versions of 95 regular papers selected for presentation; thus, the acceptance rate for this set was 28.9%.

In addition to regular papers, the books feature invited papers by:

- Erik Cambria, Nanyang Technical University, Singapore
- Mona Diab, George Washington University, USA
- Lauri Karttunen, Stanford University, USA
- Joakim Nivre, Uppsala University, Sweden

who presented excellent keynote lectures at the conference. Publication of full-text invited papers in the proceedings is a distinctive feature of the CICLing conferences.

Table 1. Number of submissions and accepted papers by topic[1]

Accepted	Submitted	% accepted	Topic
19	51	37	Emotions, sentiment analysis, opinion mining
19	56	34	Text mining
17	65	26	Arabic
17	58	29	Information extraction
17	49	35	Lexical resources
15	53	28	Information retrieval
14	35	40	Under-resourced languages
12	45	27	Semantics, pragmatics, discourse
11	40	28	Clustering and categorization
11	33	33	Machine translation and multilingualism
10	29	34	Practical applications
8	37	22	Social networks and microblogging
8	21	38	Syntax and chunking
7	17	41	Formalisms and knowledge representation
7	23	30	Noisy text processing and cleaning
5	21	24	Morphology
4	12	33	Question answering
4	10	40	Textual entailment
3	9	33	Natural language generation
3	8	38	Plagiarism detection and authorship attribution
3	13	23	Speech processing
3	21	14	Summarization
3	12	25	Word sense disambiguation
2	10	20	Computational terminology
2	8	25	Co-reference resolution
2	16	12	Named entity recognition
2	9	22	Natural language interfaces
1	1	100	Computational humor
1	15	7	Other
1	11	9	POS tagging
0	7	0	Spelling and grammar checking

[1] As indicated by the authors. A paper may belong to more than one topic.

Furthermore, in addition to presentation of their invited papers, the keynote speakers organized separate vivid informal events; this is also a distinctive feature of this conference series.

With this event, we continued with our policy of giving preference to papers with verifiable and reproducible results: in addition to the verbal description of their findings given in the paper, we encouraged the authors to provide a proof of their claims in electronic form. If the paper claimed experimental results, we asked the authors to make available to the community all the input data necessary to verify and reproduce these results; if it claimed to introduce an algorithm, we encourage the authors to make the algorithm itself, in a programming language, available to the public. This additional

Table 2. Number of submitted and accepted papers by country or region

Country or region	Authors	Papers[2]		Country or region	Authors	Papers[2]	
	Subm.	Subm.	Accp.		Subm.	Subm.	Accp.
Algeria	6	2.5	0.5	Lebanon	2	1	–
Argentina	2	0.5	0.5	Malaysia	3	3	–
Australia	4	1.33	1.33	Mexico	13	5.95	1.08
Belgium	7	2	1	Morocco	16	6	–
Brazil	19	8.5	2.5	Myanmar	3	1.5	–
Canada	20	9.5	6	The Netherlands	2	1	–
Chile	5	1	–	New Zealand	4	2	1
China	37	15.05	4.3	Nigeria	5	2	1
Colombia	6	2	1	Oman	1	1	–
Czech Rep.	11	3.33	2	Peru	6	3.5	0.5
Egypt	89	42.8	13.33	Philippines	2	1	–
Estonia	1	1	–	Poland	3	2	–
Ethiopia	2	1	–	Portugal	8	3	2
Finland	6	1.75	1.75	Qatar	1	1	–
France	30	13.05	2.58	Romania	7	5	1
Georgia	1	1	–	Russia	11	7.33	1.67
Germany	15	6.83	1.33	Saudi Arabia	10	4.42	0.5
Greece	3	1	–	Serbia	3	1	–
Hong Kong	10	3.7	2.7	Singapore	1	1	1
Hungary	2	2	2	South Africa	1	0.83	0.5
India	98	57	15	Spain	16	6.92	3.92
Iran	1	1	–	Sri Lanka	7	3	0.67
Iraq	2	1.33	0.5	Sweden	2	0.67	–
Ireland	4	1.33	1.33	Switzerland	4	2.25	1.25
Israel	6	2	–	Tunisia	57	23.58	4
Italy	9	3.33	0.33	Turkey	29	14.42	2.83
Japan	17	7.25	1.25	Ukraine	2	0.33	–
Jordan	1	0.5	–	UAE	4	2.83	0.5
Kazakhstan	4	1	1	UK	23	10.75	3.58
Korea, South	1	0.5	–	USA	36	13.78	6.58
Kuwait	1	0.17	0.17	Vietnam	3	1.67	–
				Total:	705	329	95

[2] By the number of authors: e.g., a paper by two authors from the USA and one from UK is counted as 0.67 for the USA and 0.33 for UK.

electronic material will be permanently stored on the CICLing's server, www.CICLing.org, and will be available to the readers of the corresponding paper for download under a license that permits its free use for research purposes.

In the long run, we expect that computational linguistics will have verifiability and clarity standards similar to those of mathematics: in mathematics, each claim is accompanied by a complete and verifiable proof, usually much longer than the claim itself; each theorem's complete and precise proof—and not just a description of its general idea—is made available to the reader. Electronic media allow computational linguists to provide material analogous to the proofs and formulas in mathematic in full length—

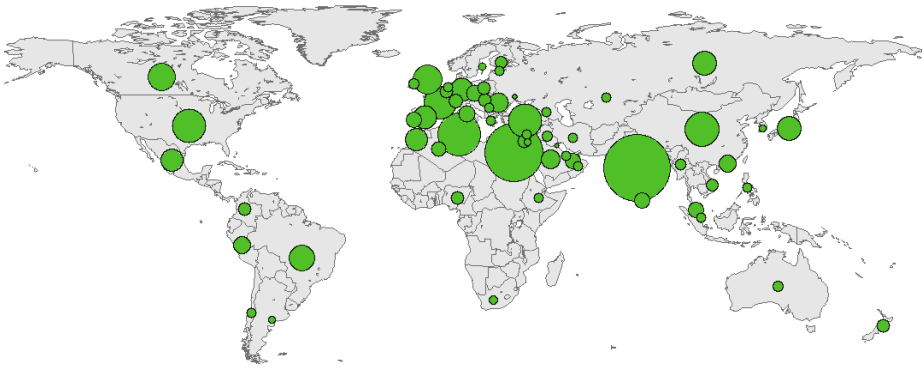


Fig. 1. Submissions by country or region. The area of a circle represents the number of submitted papers.

which can amount to megabytes or gigabytes of data—separately from a 12-page description published in the book. More information can be found on http://www.CICLing.org/why_verify.htm.

To encourage providing algorithms and data along with the published papers, we selected three winners of our Verifiability, Reproducibility, and Working Description Award. The main factors in choosing the awarded submission were technical correctness and completeness, readability of the code and documentation, simplicity of installation and use, and exact correspondence to the claims of the paper. Unnecessary sophistication of the user interface was discouraged; novelty and usefulness of the results were not evaluated—instead, they were evaluated for the paper itself and not for the data.

In this year, we introduced a policy of allowing—and encouraging—papers longer than the 12-page limit included in the fee. The reason was an observation that longer papers tend to be more complete and useful for the reader. In contrast, when a restrictive page limit is enforced, very often the authors have to omit important details; to the great frustration of the readers, this usually renders the whole paper largely useless because the presented results cannot be reproduced. This our observation was strongly confirmed by the fact that all four papers selected for the Best Paper Awards, especially the winners of the first two places, were much over the usual 12-page limit imposed by other conferences.

The following papers received the Best Paper Awards, the Best Student Paper Award, as well as the Verifiability, Reproducibility, and Working Description Awards, correspondingly:

Best Paper 1st Place: *Automated Linguistic Personalization of Targeted Marketing Messages Mining User-generated Text on Social Media*, by Rishiraj Saha Roy, Aishwarya Padmakumar, Guna Prasaad Jeganathan, and Ponnurangam Kumaraguru, India;

Best Paper 2nd Place: *Term Network Approach for Transductive Classification*, by Rafael Geraldeli Rossi, Solange Oliveira Rezende, and Alneu de Andrade Lopes, Brazil;

Best Paper 3rd Place: *Building Large Arabic Multi-domain Resources for Sentiment Analysis*, by Hady ElSahar and Samhaa R. El-Beltagy, Egypt;

Best Student Paper:[1] *Translation Induction on Indian Language Corpora using Translingual Themes from Other Languages*, by Goutham Tholpadi, Chiranjib Bhattacharyya, and Shirish Shevade, India;

Verifiability 1st Place: *Domain-specific Semantic Relatedness from Wikipedia Structure: A Case Study in Biomedical Text*, by Armin Sajadi, Evangelos E. Milios, and Vlado Keselj, Canada;

Verifiability 2nd Place: *Translation Induction on Indian Language Corpora using Translingual Themes from Other Languages*, by Goutham Tholpadi, Chiranjib Bhattacharyya, and Shirish Shevade, India;

Verifiability 3rd Place: *Opinion Summarization using Submodular Functions: Subjectivity vs Relevance Trade-off*, by Jayanth Jayanth, Jayaprakash S., and Pushpak Bhattacharyya, India.[2]

The authors of the awarded papers (except for the Verifiability award) were given extended time for their presentations. In addition, the Best Presentation Award and the Best Poster Award winners were selected by a ballot among the attendees of the conference.

Besides its high scientific level, one of the success factors of CICLing conferences is their excellent cultural program in which all attendees participate. The cultural program is a very important part of the conference, serving its main purpose: personal interaction and making friends and contacts. The attendees of the conference had a chance to visit the Giza Plateau with the Great Pyramid of Cheops and the Sphinx—probably the most important touristic place on Earth; The Egyptian Museum, the home to the largest collection of Pharaonic or ancient Egyptian relics and pieces; and the Old Cairo, to mention only a few most important attractions.

In this year we founded, and held in conjunction with CICLing, the First Arabic Computational Linguistics conference, which we expect to become the primary yearly event for dissemination of research results on Arabic language processing. This is in accordance with CICLing's mission to promote consolidation of emerging NLP communities in countries and regions underrepresented in the mainstream of NLP research and, in particular, in the mainstream publication venues. With founding this new conference, and with the very fact of holding CICLing in Egypt in a difficult moment of its history, we expect to contribute to mutual understanding, tolerance, and confidence between the Arabic world and the Western world: the better we know each other the more lasting peace between peoples.

[1] The best student paper was selected from among papers of which the first author was a full-time student, excluding the papers that received Best Paper Awards.

[2] This paper is published in a special issue of a journal and not in this book set.

I would like to thank all those involved in the organization of this conference. In the first place, these are the authors of the papers that constitute this book: it is the excellence of their research work that gives value to the book and sense to the work of all other people. I thank all those who served on the Program Committee of CICLing 2015 and of the First Arabic Computational Linguistics conference, the Software Reviewing Committee, Award Selection Committee, as well as additional reviewers, for their hard and very professional work. Special thanks go to Ted Pedersen, Savas Yıldırım, Roberto Navigli, Manuel Vilares Ferro, Kenneth Church, Dafydd Gibbon, Kais Haddar, and Adam Kilgarriff for their invaluable support in the reviewing process.

I would like to thank Prof. Tarek Khalil, president of Nile University (NU), for welcoming CICLing at NU. I also want to cordially thank the conference staff, volunteers, and members of the Local Organization Committee headed by Prof. Samhaa R. El-Beltagy and advised by Prof. Hussein Anis. In particular, I am very grateful to Ms. Aleya Serag El-Din for her great effort in coordinating all the aspects of the conference. I wish to thank the Center for Informatics Science for all the support they have provided. I am deeply grateful to the administration of the Nile University for their helpful support, warm hospitality, and in general for providing this wonderful opportunity of holding CICLing in Egypt. I am also grateful to members of the Human Foundation for their great effort in planning the cultural program. I acknowledge support from Microsoft Research, the project CONACYT Mexico–DST India 122030 "Answer Validation through Textual Entailment," and SIP-IPN grant 20150028.

The entire submission and reviewing process was supported for free by the EasyChair system (www.EasyChair.org). Last but not least, I deeply appreciate the Springer staff's patience and help in editing these volumes and getting them printed in very short time—it is always a great pleasure to work with Springer.

March 2015 Alexander Gelbukh

Organization

CICLing 2015 was hosted by the Nile University, Egypt and was organized by the CICLing 2015 Organizing Committee in conjunction with the Nile University, the Natural Language and Text Processing Laboratory of the Centro de Investigación en Computación (CIC) of the Instituto Politécnico Nacional (IPN), Mexico, and the Mexican Society of Artificial Intelligence (SMIA).

Organizing Chair

Samhaa R. El-Beltagy	Nile University, Egypt

Organizing Committee

Samhaa R. El-Beltagy	Chair	Nile University, Egypt
Hussein Anis	Senior Adviser	Nile University, Egypt
Aleya Serag El-Din	Principal co-coordinator	Nile University, Egypt
Yasser Nasr	Finance	Nile University, Egypt
Sameh Habib	Facilities and Logistics	Nile University, Egypt
Hoda El-Beleidy	Accommodation	Nile University, Egypt
Mariam Yasin Abdel Ghafar	Cultural activities	The Human Foundation, Egypt
Yasser Bayoumi	Engineering	Nile University, Egypt
Mahmoud Gabr	IT	Nile University, Egypt
Layla Al Roz		Nile University, Egypt
Hady Alsahar		Nile University, Egypt
Mohamed Fawzy		Nile University, Egypt
Amal Halby		Nile University, Egypt
Muhammad Hammad		Nile University, Egypt
Talaat Khalil		Nile University, Egypt
Yomna El-Nahas		Nile University, Egypt

Program Chair

Alexander Gelbukh	Instituto Politécnico Nacional, Mexico

Program Committee

Ajith Abraham	Machine Intelligence Research Labs (MIR Labs), USA
Rania Al-Sabbagh	University of Illinois at Urbana-Champaign, USA
Marianna Apidianaki	LIMSI-CNRS, France
Alexandra Balahur	European Commission Joint Research Centre, Italy
Sivaji Bandyopadhyay	Jadavpur University, India

Srinivas Bangalore	AT&T Labs-Research, USA
Leslie Barrett	Bloomberg, LP, USA
Roberto Basili	University of Rome Tor Vergata, Italy
Núria Bel	Universitat Pompeu Fabra, Spain
Anja Belz	University of Brighton, UK
Pushpak Bhattacharyya	IIT Bombay, India
António Branco	University of Lisbon, Portugal
Nicoletta Calzolari	Istituto di Linguistica Computazionale – CNR, Italy
Nick Campbell	TCD, Ireland
Michael Carl	Copenhagen Business School, Denmark
Violetta Cavalli-Sforza	Al Akhawayn University, Morocco
Niladri Chatterjee	IIT Delhi, India
Kenneth Church	IBM, USA
Dan Cristea	A.I.Cuza University of Iași, Romania
Walter Daelemans	University of Antwerp, Belgium
Samhaa R. El-Beltagy	Cairo University, Egypt
Michael Elhadad	Ben-Gurion University of the Negev, Israel
Anna Feldman	Montclair State University, USA
Alexander Gelbukh (Chair)	Instituto Politécnico Nacional, Mexico
Dafydd Gibbon	Universität Bielefeld, Germany
Gregory Grefenstette	Inria, France
Eva Hajičová	Charles University, Czech Republic
Sanda Harabagiu	University of Texas at Dallas, USA
Yasunari Harada	Waseda University, Japan
Karin Harbusch	University of Koblenz and Landau, Germany
Ales Horak	Masaryk University, Czech Republic
Veronique Hoste	Ghent University, Belgium
Nancy Ide	Vassar College, USA
Diana Inkpen	University of Ottawa, Canada
Aminul Islam	Dalhousie University, Canada
Guillaume Jacquet	JRC, Italy
Doug Jones	MIT, USA
Sylvain Kahane	Université Paris Ouest Nanterre La Défense and CNRS/Alpage, Inria, France
Alma Kharrat	Microsoft Research, USA
Adam Kilgarriff	Lexical Computing Ltd, UK
Philipp Koehn	University of Edinburgh, UK
Valia Kordoni	Humboldt University Berlin, Germany
Leila Kosseim	Concordia University, Canada
Mathieu Lafourcade	LIRMM, France
Krister Lindén	University of Helsinki, Finland
Bing Liu	University of Illinois at Chicago, USA
Elena Lloret	University of Alicante, Spain
Bente Maegaard	University of Copenhagen, Denmark
Cerstin Mahlow	University of Stuttgart, IMS, Germany

Suresh Manandhar	University of York, UK
Sun Maosong	Tsinghua University, China
Diana McCarthy	University of Cambridge, UK
Alexander Mehler	Goethe-Universität Frankfurt am Main, Germany
Rada Mihalcea	University of Michigan, USA
Evangelos Milios	Dalhousie University, Canada
Jean-Luc Minel	Université Paris Ouest Nanterre La Défense, France
Dunja Mladenic	Jožef Stefan Institute, Slovenia
Marie-Francine Moens	Katholieke Universiteit Leuven, Belgium
Masaki Murata	Tottori University, Japan
Preslav Nakov	Qatar Computing Research Institute, Qatar Foundation, Qatar
Roberto Navigli	Sapienza Università di Roma, Italy
Joakim Nivre	Uppsala University, Sweden
Kjetil Nørvåg	Norwegian University of Science and Technology, Norway
Attila Novák	Pázmány Péter Catholic University, Hungary
Kemal Oflazer	Carnegie Mellon University in Qatar, Qatar
Constantin Orasan	University of Wolverhampton, UK
Ekaterina Ovchinnikova	KIT, Karlsruhe and ICT, University of Heidelberg, Germany
Ivandre Paraboni	University of São Paulo – USP/EACH, Brazil
Maria Teresa Pazienza	University of Rome, Tor Vergata, Italy
Ted Pedersen	University of Minnesota Duluth, USA
Viktor Pekar	University of Birmingham, UK
Anselmo Peñas	Universidad Nacional de Educación a Distancia, Spain
Stelios Piperidis	Institute for Language and Speech Processing, Greece
Octavian Popescu	FBK-IRST, Italy
Marta R. Costa-Jussà	Institute For Infocomm Research, Singapore
German Rigau	IXA Group, Universidad del País Vasco / Euskal Herriko Unibertsitatea, Spain
Fabio Rinaldi	IFI, University of Zürich, Switzerland
Horacio Rodriguez	Universitat Politècnica de Catalunya, Spain
Paolo Rosso	Technical University of Valencia, Spain
Vasile Rus	The University of Memphis, USA
Horacio Saggion	Universitat Pompeu Fabra, Spain
Patrick Saint-Dizier	IRIT-CNRS, France
Franco Salvetti	University of Colorado at Boulder and Microsoft Inc., USA
Rajeev Sangal	Language Technologies Research Centre, India
Kepa Sarasola	Euskal Herriko Unibertsitatea, Spain
Roser Sauri	Pompeu Fabra University, Spain

Hassan Sawaf	eBay Inc., USA
Satoshi Sekine	New York University, USA
Bernadette Sharp	Staffordshire University, UK
Grigori Sidorov	Instituto Politécnico Nacional, Mexico
Vivek Kumar Singh	Banaras Hindu University, India
Vaclav Snasel	VSB-Technical University of Ostrava, Czech Republic
Efstathios Stamatatos	University of the Aegean, Greece
Josef Steinberger	University of West Bohemia, Czech Republic
Jun Suzuki	NTT, Japan
Stan Szpakowicz	SITE, University of Ottawa, Canada
Juan-Manuel Torres-Moreno	Laboratoire Informatique d'Avignon / Université d'Avignon et des Pays de Vaucluse, France
George Tsatsaronis	Technical University of Dresden, Germany
Dan Tufiș	Institutul de Cercetări pentru Inteligență Artificială, Academia Română, Romania
Olga Uryupina	University of Trento, Italy
Renata Vieira	Pontifícia Universidade Católica do Rio Grande do Sul, Brazil
Manuel Vilares Ferro	University of Vigo, Spain
Aline Villavicencio	Universidade Federal do Rio Grande do Sul, Brazil
Piotr W. Fuglewicz	TiP Sp. z o. o., Poland
Bonnie Webber	University of Edinburgh, UK
Savaş Yıldırım	Istanbul Bilgi University, Turkey

Software Reviewing Committee

Ted Pedersen	University of Minnesota Duluth, USA
Florian Holz	Universität Leipzig, Germany
Miloš Jakubíček	Lexical Computing Ltd, UK, and Masaryk University, Czech Republic
Sergio Jiménez Vargas	Universidad Nacional, Colombia
Miikka Silfverberg	University of Helsinki, Finland
Ronald Winnemöller	Universität Hamburg, Germany

Award Committee

Alexander Gelbukh	Instituto Politécnico Nacional, Mexico
Eduard Hovy	Carnegie Mellon University, USA
Rada Mihalcea	University of Michigan, USA
Ted Pedersen	University of Minnesota Duluth, USA
Yorick Wilks	University of Sheffield, UK

Additional Referees

Naveed Afzal
Rodrigo Agerri
Saad Alanazi
Itziar Aldabe
Iñaki Alegria
Haifa Alharthi
Henry Anaya-Sánchez
Vít Baisa
Francesco Barbieri
Janez Brank
Jorge Carrillo de Albornoz
Dave Carter
Jean-Valère Cossu
Victor Manuel Darriba Bilbao
Elnaz Davoodi
Claudio Delli Bovi
Bart Desmet
Steffen Eger
Luis Espinosa Anke
Tiziano Flati
Ivo Furman
Dimitrios Galanis
Dario Garigliotti
Mehmet Gençer
Rohit Gupta
Afli Haithem
Ignacio J. Iacobacci
Milos Jakubicek
Magdalena Jankowska
Assad Jarrahian
Kirill Kireyev
Vojtech Kovar
Majid Laali
John Lowe
Andy Lücking
Raheleh Makki Niri
Shachar Mirkin
Alexandra Moraru
Andrea Moro
Mohamed Mouine
Mohamed Outahajala
Michael Piotrowski
Alessandro Raganato
Arya Rahgozar
Christoph Reichenbach
Francisco José Ribadas-Pena
Alvaro Rodrigo
Francesco Ronzano
Pavel Rychly
Armin Sajadi
Ehsan Sherkat
Janez Starc
Yasushi Tsubota
Tim vor der Brück
Rodrigo Wilkens
Tuğba Yıldız

First Arabic Computational Linguistics Conference

Program Chairs

Alexander Gelbukh	Instituto Politécnico Nacional, Mexico
Khaled Shaalan	The British University in Dubai, UAE

Program Committee

Bayan Abu-shawar	Arab Open University, Jordan
Hanady Ahmed	Qatar University, Qatar
Hend Al-Khalifa	King Saud University, Saudi Arabia
Mohammed Attia	Al-Azhar University, Egypt

Aladdin Ayesh	De Montfort University, UK
Khalid Choukri	ELDA, France
Samhaa R. El-Beltagy	Cairo University, Egypt
Ossama Emam	ALTEC, Egypt
Aly Fahmy	Cairo University, Egypt
Alexander Gelbukh (Chair)	Instituto Politécnico Nacional, Mexico
Ahmed Guessoum	University of Science and Technology Houari Boumediene, Algeria
Nizar Habash	New York University Abu Dhabi, UAE
Kais Haddar	MIRACL Laboratory, Faculté des sciences de Sfax, Tunisia
Lamia Hadrich Belguith	MIRACL Laboratory, Tunisia
Sattar Izwaini	American University of Sharjah, UAE
Mark Lee	University of Birmingham, UK
Sherif Mahdy Abdou	RDI, Egypt
Farid Meziane	University of Salford, UK
Herman Moisl	Newcastle University, UK
Ahmed Rafea	American University in Cairo, Egypt
Allan Ramsay	University of Manchester, UK
Mohsen Rashwan	Cairo University, Egypt
Paolo Rosso	Technical University of Valencia, Spain
Nasredine Semmar	CEA, France
Khaled Shaalan (Chair)	The British University in Dubai, UAE
William Teahan	Bangor University, UK
Imed Zitouni	IBM Research, USA

Additional Referees

Sherif Abdou	Feryal Haj Hassan
Nora Al-Twairesh	Harish Tayyar Madabushi

Website and Contact

The webpage of the CICLing conference series is http://www.CICLing.org. It contains information about past CICLing conferences and their satellite events, including links to published papers (many of them in open access) or their abstracts, photos, and video recordings of keynote talks. In addition, it contains data, algorithms, and open-source software accompanying accepted papers, in accordance with the CICLing verifiability, reproducibility, and working description policy. It also contains information about the forthcoming CICLing events, as well as contact options.

Contents – Part I

Grammar Formalisms and Lexical Resources

Morphology and Chunking

Syntax and Parsing

Anaphora Resolution and Word Sense Disambiguation

Semantics and Dialogue

Invited Paper:

Verifiability Award, First Place:

Machine Translation and Multilingualism

Contents – Part II

Sentiment Analysis and Emotion Detection

Opinion Mining and Social Network Analysis

Natural Language Generation and Text Summarization

Information Retrieval, Question Answering, and Information Extraction

Text Classification

Speech Processing

Applications

Grammar Formalisms and Lexical Resources

Towards a Universal Grammar for Natural Language Processing

Joakim Nivre

Uppsala University,
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Abstract. Universal Dependencies is a recent initiative to develop cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective. In this paper, I outline the motivation behind the initiative and explain how the basic design principles follow from these requirements. I then discuss the different components of the annotation standard, including principles for word segmentation, morphological annotation, and syntactic annotation. I conclude with some thoughts on the challenges that lie ahead.

1 Introduction

The notion of a universal grammar in linguistics and philosophy goes back at least to Roger Bacon's observation that "[i]n its substance, grammar is one and the same in all languages, even if it accidentally varies" [1, p. 27]. It can be traced forward through the speculative grammars of the Middle Ages and the Port-Royal grammar of Arnauld and Lancelot [2], all the way to the theories of Noam Chomsky [3,4]. What these theories have in common is the assumption that all human languages are species of a common genus because they have all been shaped by a factor that is common to all human beings. For the speculative grammarians working in the Aristotelian tradition of scholastic philosophy, this factor was simply thought to be the world itself. Arnauld and Lancelot replaced the external world by the human mind, which in Chomskyan linguistics has been further specified to an innate language faculty. Regardless of these differences, the main idea is that we can bring order into the chaos of linguistic variation by referring to a common underlying structure.

How is this relevant for natural language processing? Traditionally, research in our community has not paid much attention to language typology or linguistic universals. At one end of the scale, we find systems based on language-specific resources that cannot easily be ported or generalized even to closely related languages. At the other end, we find general statistical models that can be applied to any language and therefore are not attuned to the special characteristics of any individual language. The first approach eschews linguistic variation altogether; the second embraces it with ignorance; but neither manages to bring any order into the chaos.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 3–16, 2015.
DOI: 10.1007/978-3-319-18111-0_1

There are definitely signs that this is about to change. Research on statistical parsing of morphologically rich languages has highlighted the interplay between language typology and parsing technology [5,6]. Studies on cross-lingual learning have shown that typological information can be exploited to improve learning and adaptation across languages [7,8]. However, a major obstacle to progress in these areas has been the fact that annotation standards vary across languages almost as much as the languages themselves. Hence, it is often very difficult to say exactly which differences in performance are due to real structural differences between languages, as opposed to more or less arbitrary differences in annotation conventions.

In this paper, I will present a recent initiative to overcome these difficulties by creating guidelines for cross-linguistically consistent grammatical annotation, called Universal Dependencies (UD). The UD project builds on and subsumes several earlier initiatives, including Interset [9], Google Universal Part-of-Speech Tags [10], HamleDT [11], Universal Dependency Treebanks [12], and Universal Stanford Dependencies [13]. We may think of UD as a universal grammar for natural language processing, but as such it is fundamentally different from the notion of universal grammar found in linguistics and philosophy. Our goal is not to give an explanatory account of the structural variation found in the world's languages, but to represent it in a way that is revealing and useful for the purpose of natural language processing. Hence, as long as the representation is found to be practically useful, it is immaterial whether it captures the true universal grammar, or indeed whether such a grammar even exists. Needless to say, it would be very rewarding if our efforts turned out to have significance also for the more theoretical discussion of a universal grammar, but our current ambitions are more modest than that.

2 Motivation

Syntactic parsing is rarely an end in itself, so its main role in natural language processing is to extract information about grammatical structure from sentences for the benefit of applications like machine translation, information extraction, and question answering. It is an open question to what extent these applications benefit from grammatical information, and in what form it should be provided, but we currently see a trend towards an increased use of parsing, also in applications like information retrieval that traditionally have not considered grammatical information. The recent parsing trend has also clearly favored dependency-based representations, which provide a simple and transparent encoding of predicate-argument structure and which are also amenable to very efficient processing.

To develop efficient and accurate parsers, we currently need access to grammatically annotated corpora, or treebanks. Although unsupervised parsing is an interesting alternative in theory, the accuracy is still much too low for practical purposes. This is a bottleneck because grammatical annotation is expensive. Thus, treebanks are only available for a small fraction of the world's languages,

and the amount of data for each language is often quite limited. Moreover, the annotation schemes vary considerably across languages, which makes it hard to use data from rich-resource languages to bootstrap parsers for low-resource languages.

The large variation in annotation schemes across languages can to some extent be explained by different theoretical preferences among treebank developers. More important, however, is the fact that broad-coverage linguistic annotation almost inevitably has to rely on descriptive grammatical traditions established over long time for specific languages. These traditions are often roughly similar across similar languages but nevertheless with more or less arbitrary (and often quite subtle) differences in terminology and notation. When these differences are inherited into treebank annotation schemes, they give rise to a number of problems for researchers and developers in natural language processing, including but not limited to the following [12]:

- In multilingual applications involving parsers, it is a major drawback if downstream components cannot assume uniform output representations from parsing, because we then require specialized interfaces for each language.
- In cross-lingual learning and parser evaluation, inconsistent annotation standards make it impossible to properly evaluate system performance, because we cannot separate parse errors from discrepancies in the standards.
- In statistical parsing generally, it is hard to make effective use of linguistic knowledge, because we cannot assume a consistent representation of linguistic categories and structures across languages.

The major goal of the UD project is to alleviate these problems by creating a standard for cross-linguistically consistent grammatical annotation, a standard that brings out cross-lingual similarities in a perspicuous way without forcing all languages into the same mold. In a way, UD tries to do for statistical parsing what initiatives like ParGram [14] and DELPH-IN [15] have done for the grammar-based parsing community, and we focus on the needs of multilingual natural language processing from a mainly practical point of view. This does not preclude that UD can be useful for other purposes as well, but a few disclaimers may be appropriate in order not to create false expectations:

- UD is not proposed as a linguistic theory. While we like to think that most of our design decisions are informed by linguistic theory, we are also well aware that we sometimes have to make compromises in the interest of practical utility. The representations are in general oriented towards surface syntax, but in the interest of a transparent encoding of predicate-argument structure we also encode aspects of deep syntax, and the representations probably do not correspond to any well-defined level in theoretical grammar frameworks. We nevertheless think that UD could be a useful resource also for more linguistically oriented studies of grammatical structure across languages.
- UD may not be the ideal treebank annotation scheme for all projects. The main goal is to provide a kind of *lingua franca* for grammatical annotation,

which can be used for data interchange and development of multilingual systems, but we do not have the ambition to capture all the information that is encoded in specific treebank annotation schemes. Hence, we do not expect all treebank developers to abandon their language-specific schemes in favor of UD, but we do hope that treebank developers will find UD useful enough to make the extra effort to ensure that their own scheme can be converted to UD in a noiseless fashion (though possibly with some loss of information). In addition, we think that UD could be a convenient choice for quick-starting new annotation projects given the availability of consistent guidelines for many languages.

– UD is not necessarily an optimal parsing representation. It is clear that the need for cross-linguistic consistency and perspicuity often runs counter to the requirements of optimal parsability for specific languages. We therefore envisage that parsers expected to output UD representations often will have to use different representations internally. In fact, we believe that research on finding optimal representations for parsers, which has been a dominant theme in constituency-based parsing for the last twenty years, is an under-exploited area in dependency parsing. With a touch of irony, we could even say that the obvious suboptimality of UD representations for parsing is our way of encouraging more research into these problems.

3 Design Principles

Given our ambition to support both parsing research and system development in a multilingual setting, we have opted for an annotation standard that is close to common usage and based on existing de facto standards. The basic structure of the annotation is that *sentences* are segmented into *words* and that words are described by *morphological properties* and linked by *syntactic relations*. A typical representation of this kind is shown in Figure 1.[1]

The decision to treat words as the basic units of analysis constitutes a commitment to the *lexicalist hypothesis* in syntax [13], but it is also consistent with common practice in practical natural language processing. This means that we do not attempt any morphological segmentation of words but instead use a word-based morphology [17], where morphological categories are represented as properties of whole words. By words, however, we mean *syntactic* words (not orthographic or phonological words), so clitics are treated as separate words regardless of how they are represented orthographically, and contractions are split if they consist of syntactically independent words. The principles of word segmentation are described in more detail in Section 4.

The morphological description of a word consists of three parts: a lemma (or base form), a universal part-of-speech tag, and a set of morphological features

[1] The format used to encode these annotation is a revised version of the CoNLL-X format for dependency treebanks [16] called CoNLL-U. For more information about this format, see `http://universaldependencies.github.io/docs/format.html`.

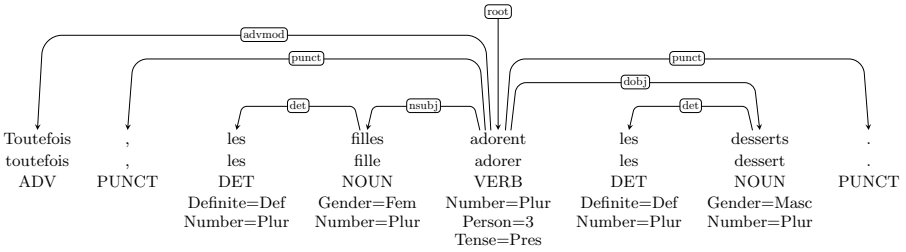


Fig. 1. UD representation of a French sentence

(attribute-value pairs). The UD tagset is a revised and extended version of the Google Universal Part-of-Speech Tagset [10], and the inventory of morphological attributes and values is based on Interset [9]. The morphological annotation is further described in Section 5.

The adoption of lexicalism and word-based morphology fits very well with a dependency-based view of syntax [18], which is also the most widely used form of syntactic annotation in available treebanks. In addition, as noted earlier, it is currently favored over other representations by system developers. UD adopts a version of the Universal Stanford Dependencies [13], containing 40 universal dependency relations, further described in Section 6.

The overriding principle of the guidelines for morphological and syntactic annotation is to bring out similarities and differences between languages by maximizing parallelism in annotations. This can be summed up in two slogans:

- Don't annotate the same thing in different ways!
- Don't make different things look the same!

However, it is important to apply these principles with reason, to avoid turning the annotation scheme into a Procrustean bed. Hence, we also apply the slogan:

- Don't annotate things that are not there!

For instance, we do not introduce empty subjects in pro-drop languages just because other languages have obligatory overt subjects. We also allow language-specific extensions of the annotation scheme in two places. In the morphological features, each language selects a subset of the universal features and can in addition add language-specific features (cf. Section 5). In the syntactic relations, each language may define language-specific subtypes of the universal relations (cf. Section 6). The subtyping is important because it gives us a mechanism for backing off to completely homogeneous representations in contexts where this is important. The language-specific documentation for each treebank should specify what language-specific extensions are used (if any), so that users of the treebanks can make informed choices about how to use the resource.

Finally, it must be emphasized that the UD scheme is still evolving. The first version of the guidelines, released in October 2014, will remain stable for at least

a year (and probably longer) to give the community a chance to explore it and apply it in diferent projects. But it is very unlikely that the first version is also the final version. Therefore, we are very eager to get feedback on the first version from treebank developers, parsing researchers and system developers alike. Universal Dependencies is an open project and anyone is invited to contribute by developing guidelines for a new language, contributing treebank data, or just providing feedback on existing guidelines and treebanks.

4 Word Segmentation

It is hard to give exact criteria for distinguishing syntactic words in all languages, but the basic idea is that a syntactic word can be assigned a single consistent morphological description with a unique lemma, part-of-speech tag and morphological feature set, as well as a single syntactic function in relation to other words of the sentence. A consequence of this characterization is that clitics normally need to be separated from their hosts. For example, in the Spanish orthographic word *dámelo* (give it to me), there are three different parts of speech (verb, pronoun, pronoun) and three different syntactic functions (predicate, indirect object, direct object). Hence, it should be split into three separate words: *da*, *me*, *lo*. Similarly, for a contraction like the French *au*, we need to postulate two words *à* and *le* with different parts of speech (adposition, determiner) and syntactic functions (case marker, determiner). In principle, the word-based view could also be taken to imply that certain fixed multiword annotations should be treated as single words in the annotation. So far, however, multiword expressions are annotated using special dependency relations, instead of collapsing multiple tokens into one, which has the additional advantage that we can accommodate discontiguous multiword expressions.

Since word segmentation in general is a non-trivial task in many languages, and since the usefulness of tools trained on treebank data ultimately depends on how well the word segmentation can be reproduced for new data, it is important to document the principles of word segmentation for each language. The nature of this documentation will vary from one language to the next, depending on properties of the language and the writing system. For languages where word segmentation can be performed by a simple script given white-space and punctuation, only the words need to be represented in the treebank. For languages not using white-space at all, such as Chinese and Japanese, a complex word segmentation algorithm has to be employed, but there is no need to represent the basic character sequence in the treebank since it is completely recoverable from the word representation. By contrast, in languages where the mapping between white-space delimited tokens and syntactic words is highly ambiguous, such as Arabic and Hebrew, we provide the option of including both tokens and words in the treebank using a two-level indexing scheme. The morphological and syntactic annotation is only defined at the word level, but a heuristic mapping to the token level can usually be provided. The language-specific documentation for each treebank must describe how word segmentation has been performed,

Table 1. Morphological annotation: universal part-of-speech tags and features

Part-of-Speech Tags	
ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary verb
CONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Features	
Animacy	animacy
Aspect	aspect
Case	case
Definite	definiteness or state
Degree	degree of comparison
Gender	gender
Mood	mood
Negative	if a word is negated
NumType	numeral type
Number	number
Person	person
Poss	possessive
PronType	pronominal type
Reflex	reflexive
Tense	tense
VerbForm	form of verb
Voice	voice

whether the treebank includes (multiword) tokens as well as words, and what types of white-space separated tokens are split into multiple words (if any).

5 Morphological Annotation

The morphological description of a word consists of three components:

- A lemma representing the semantic content of the word.
- A part-of-speech tag representing the abstract lexical category of the word.
- A set of features representing lexical and grammatical properties associated with the lemma or the particular word form.

Lemmas are typically determined by language-specific dictionaries. By contrast, the part-of-speech tags and features are taken from two universal inventories. The list of universal part-of-speech tags is a fixed list containing 17 tags shown in Table 1 (left). It is based on the Google Universal Part-of-Speech Tagset [10], which in turn is based on a generalization over tagsets in the CoNLL-X shared task on multilingual dependency parsing [19]. In the new version, the category VERB has been split into AUX and VERB, NOUN into NOUN and PROPN, and CONJ into CONJ and SCONJ; the two new categories INTJ and SYM have been added; and the category PRT has been renamed PART to dissociate it from the label commonly used for verb particles because the universal tag covers a larger class of grammatical particles. The universal tags must be used in all UD treebanks. Some tags may not be used in all treebanks, but the list

cannot be extended with language-specific categories. Instead, more fine-grained classifications can be achieved via the use of features.[2]

Features give additional information about the word, its part of speech and morphosyntactic properties. Every feature has the form Name=Value and a word can have any number of features.[3] Table 1 lists our current set of universal features, which are all attested in multiple corpora and need to be standardized. The list is certainly not exhaustive and later versions of the standard may include new features or values found in new languages, corpora or tagsets. Users can extend the set of universal features with language-specific features as needed. Such features must be described in the language-specific documentation and follow the general format principles. In addition to simple features, we also provide a mechanism for *layered* features in cases where the same feature is marked more than once on the same word. This happens, for instance, in the case of possessives that agree with both the possessor and the possessed.

6 Syntactic Annotation

The syntactic annotation consists of typed dependency relations between words, with a special relation *root* for words that do not depend on any other word. Every sentence is associated with a set of *basic* dependencies that form a rooted tree representing the backbone of the syntactic structure. Many grammatical constructions introduce additional dependencies, which can be represented in an *enhanced* dependency structure, which is a general directed graph. Examples of such constructions are secondary predication, control structures and dependencies that need to be propagated over coordination structures. The guidelines for the enhanced structure are still under development and will not be discussed further in this paper.

The universal dependency relations are meant to capture a set of broadly observed grammatical functions that work across languages. More precisely, we want to maximize parallelism by allowing the same grammatical relation to be annotated in the same way across languages, while making enough crucial distinctions to differentiate constructions that are not the same. As a fundamental principle we assume that dependency relations hold primarily between content words, rather than being indirect relations mediated by function words. This principle is illustrated in Figure 2, where the solid arcs represent direct dependencies between content words.

Given the dependency relations between content words, function words attach as direct dependents of the most closely related content word (dashed arcs), while punctuation marks attach to the head of the clause or phrase to which they belong (dotted arc). Preferring content words as heads maximizes parallelism between languages because content words vary less than function words

[2] In addition, the CoNLL-U format allows the inclusion of language-specific tags on top of the universal ones.

[3] For readability, Figure 1 displays multiple features on top of each other, whereas the CoNLL-U format uses a vertical bar to separate them: Gender=Fem|Number=Plur.

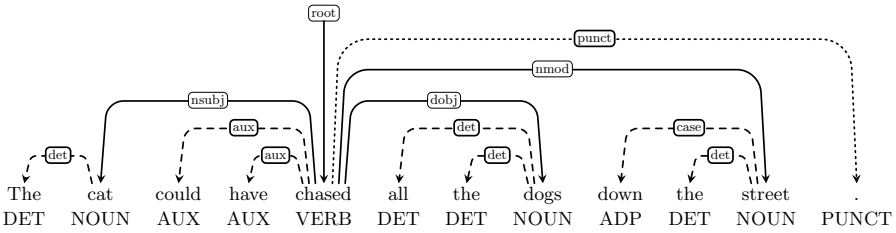


Fig. 2. Dependency tree for an English sentence (lemmas and features omitted)

between languages. In particular, one commonly finds the same grammatical relation being expressed by morphology in some languages or constructions and by function words in other languages or constructions, while some languages may not mark the information at all (such as not marking tense or definiteness). Therefore, we treat adpositions as dependents of nouns, rather than the other way round, because they often correspond to case markers or nothing at all in other languages. We also treat auxiliary verbs as dependents of main predicates, with the copula as a dependent of a nominal or adjectival predicate as a special case.

We assume the taxonomy of the Universal Stanford Dependencies [13], which posits the 40 syntactic relations listed in Table 2.[4] The main organizing principle of this taxonomy is the distinction between three types of syntactic structures:

- Nominal phrases, primarily denoting entities but also used for other things.
- Clauses headed by a predicate, usually a verb but sometimes an adjective, an adverb, or a predicate nominal.
- Miscellaneous other kinds of modifier words, which may allow modifiers but which do not expand into rich structures like nominal phrases and clauses.

This distinction is reflected in two dimensions in the upper part of Table 2, where the three columns represent *dependents* of all three types, while rows represents different types of constructions where the *head* is either a clausal predicate or a nominal. The taxonomy also distinguishes between core arguments (subjects, objects, clausal complements) and other dependents, but it makes no attempt to distinguish adjuncts from oblique arguments. The latter distinction has proven notoriously difficult to draw in practice and is often omitted in treebank annotation schemes, notably that of the Penn Treebank.

The first row in Table 2 shows relations for core arguments of predicates, with one series for nominal arguments (*nsubj*, *nsubjpass*, *dobj*, *iobj*) and one for clausal arguments (*csubj*, *csubjpass*, *ccomp*, *xcomp*). For subjects, we differentiate canonical voice (*nsubj*, *csubj*), where the proto-agent argument is the subject, from non-canonical voice (*nsubjpass*, *csubjpass*), where another argument is the

[4] The current UD inventory of syntactic relations differs from that described in [13] by omitting the relations *nfincl*, *relcl* and *ncmod* and adding the relation *acl*.

Table 2. Universal dependency relations. Dependents of predicates are divided into core (top), non-core (middle), and special (bottom) dependents. Dependents marked * are auxiliary verbs rather than real predicates. Note that some relations occur in more than one place.

	Nominal Dep	**Predicate Dep**	**Other Dep**
Predicate Head	nsubj	csubj	
	nsubjpass	csubjpass	
	dobj	ccomp	
	iobj	xcomp	
	nmod	advcl	advmod
			neg
	vocative	aux*	mark
	discourse	auxpass*	punct
	expl	cop*	
	dislocated		
Nominal Head	nummod	acl	amod
	appos		det
	nmod		neg
			case
No Head		root	

Compounding	**Coordination**	**Other**
compound	conj	list
name	cc	parataxis
mwe	punct	remnant
goeswith		reparandum
		foreign
		dep

subject. For clausal complements, we differentiate clauses with obligatory control (*xcomp*) from clauses with other types of subject licensing (*ccomp*), but we do not differentiate finite from nonfinite clauses.

The second row in Table 2 shows the relations for non-core dependents of predicates, which differentiates nominal, clausal and other dependents:

John talked [in the movie theatre] (*nmod*)
John talked [while we were watching the movie] (*advcl*)
John talked [very quickly] (*advmod*)

The third row contains special dependents of the predicate, including function words like auxiliary verbs (*aux*, *auxpass*, *cop*), complementizers (*mark*), and punctuation (*punct*), as well as dependents that are more loosely connected to the predicate, such as vocatives and discourse particles. The fourth row contains dependents of nominal heads, again divided into three structural classes. Finally, we have the *root* relation, which is used for independent words, usually the predicate of a main clause.

The lower part of Table 2 displays relations that can occur with (almost) any type of head and that do not necessarily correspond to traditional grammatical relations. The first class covers lexical relations like compounding (*compound*), which we take to be fundamentally different from phrasal modification, fixed multiword expressions (*mwe*), and complex names that lack compositional structure (*name*). The second class is concerned with coordination, which is analyzed as an asymmetrical relation, where the first conjunct is the head on which all other conjuncts depend via the *conj* relation. Coordinating conjunctions and punctuation delimiting the conjuncts are attached using the *cc* and *punct* relations, respectively.

The primacy of content words implies that function words normally do not have dependents of their own. In particular, it means that multiple function words related to the same content word always appear as siblings, never in a nested structure, regardless of their interpretation. Typical cases in Figure 2 are auxiliary verbs (*could have*) and multiple determiners (*all the*). One possible interpretation of these flat structures is that they constitute dissociate nuclei in the sense of Tesnière [20], rather than regular dependency structures, and that the function words modify the syntactic category of their head, rather than performing a grammatical function in relation to a nominal or predicate. Nevertheless, there are a few exceptions to the rule that function words do not take dependents:

- Multiword function words: The word forms that make up a fixed multiword expressions are connected into a head-initial structure using the special dependency relation *mwe*. When the multiword expression is a functional element, the initial word form will then superficially look like a function word with dependents. Typical examples are *in spite of*, *because of*, *by and large*.
- Coordinated function words: Head coordination is a syntactic process that can apply to almost any word category, including function words like conjunctions and prepositions. In line with the general analysis of coordination, the first conjunct will then be the head of both the conjunction and the second conjunct, regardless of whether it is a function word or a content word. Examples: *to and from*, *if and when*.
- Promotion by head elision: When the natural head of a function word is elided, we "promote" the function word to the function normally assumed by the content word head (instead of introducing a null node representing the head). Typical examples are:

  Bill could not answer, but Ann could. [*conj*(answer, could)]
  She forgot which address she wrote to. [*nmod*(wrote, to)]
  I know how. [*ccomp*(know, how)]

In addition, certain types of function words can take a restricted class of modifiers, mainly negation and light adverbials. Typical cases are modified determiners like *not every* and *exactly two* as well as modifiers of subordinating conjunctions *right when*.

In addition to the basic universal dependencies, it is always possible to add language-specific subtypes for constructions that are of special significance in a given language. These relations have the form *uni*:*spec*, where *uni* is one of the 40 universal relations, and *spec* is a descriptive label. Commonly used subtypes in the first release are *acl:relcl*, for relative clauses as a subtype of clauses modifying nouns, and *compound:prt* for verb particles. Language-specific subtypes of the universal relations must be described in the language-specific documentation.

7 Conclusion

I have presented Universal Dependencies, a recent initiative to create guidelines for cross-linguistically consistent grammatical annotation. So far, a first version of the guidelines has been released, as well as a first batch of treebanks for ten languages: Czech, English, Finnish, French, German, Hungarian, Irish, Italian, Spanish and Swedish.

In order to increase the usefulness of these resources, there are two important challenges for the future. The first is to improve the quality of the annotated treebanks with respect to real cross-linguistic consistency, as opposed to merely notational consistency. The second is to expand the coverage of languages and increase the typological diversity. This will require contributions from the entire treebank and parsing community, so we invite anyone who is interested to take part in this development.

It still remains to be seen whether we will ever manage to construct something that deserves to be called a universal grammar for natural language processing. After all, the quest for a real universal grammar is still on almost 800 years after Bacon's initial observation. So even if our goals ar more modest, we may need another decade or two to figure it out.

Acknowledgments. Sections 4–6 of this paper are based on the UD guidelines, which is collaborative work with Jinho Choi, Marie-Catherine de Marneffe, Tim Dozat, Filip Ginter, Yoav Goldberg, Jan Hajič, Chris Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. I also want to thank Masayuki Asahara, Cristina Bosco, Binyam Ephrem, Richárd Farkas, Jennifer Foster, Koldo Gojenola, Hiroshi Kanayama, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shinsuke Mori, Petya Osenova, Prokopis Prokopidis, Mojgan Seraji, Maria Simi, Kiril Simov, Aaron Smith, Takaaki Tanaka, Sumire Uematsu, and Veronika Vincze, for contributions to language-specific guidelines and/or treebanks. The views expressed in this paper on the status and motivation of UD are primarily my own and are not necessarily shared by all contributors to the project.

References

1. Nolan, E., Hirsch, S. (eds.): The Greek Grammar of Roger Bacon and a Fragment of his Hebrew Grammar. Cambridge University Press (1902)
2. Brekle, H.E., Lancelot, C., Arnauld, A.: Grammaire générale et raisonnée, ou La Grammaire de Port-Royal. Friedrich Frommann Verlag (1966)
3. Chomsky, N.: Aspects of the Theory of Syntax. MIT Press (1965)
4. Chomsky, N.: Cartesian Linguistics. Harper and Row (1965)
5. Tsarfaty, R., Seddah, D., Goldberg, Y., Kuebler, S., Versley, Y., Candito, M., Foster, J., Rehbein, I., Tounsi, L.: Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pp. 1–12 (2010)
6. Tsarfaty, R.: A unified morpho-syntactic scheme of Stanford dependencies. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 578–584 (2013)
7. Naseem, T., Barzilay, R., Globerson, A.: Selective sharing for multilingual dependency parsing. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 629–637 (2012)
8. Täckström, O., McDonald, R., Nivre, J.: Target language adaptation of discriminative transfer parsers. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pp. 1061–1071 (2013)
9. Zeman, D.: Reusable tagset conversion using tagset drivers. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), pp. 213–218 (2008)
10. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
11. Zeman, D., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., Hajič, J.: HamleDT: To parse or not to parse? In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), pp. 2735–2741 (2012)
12. McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., Lee, J.: Universal dependency annotation for multilingual parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 92–97 (2013)
13. de Marneffe, M.C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal Stanford Dependencies: A cross-linguistic typology. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC), pp. 4585–4592 (2014)
14. Butt, M., Dyvik, H., Holloway King, T., Masuichi, H., Rohrer, C.: The parallel grammar project. In: Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics, pp. 1–7 (2002)
15. Bender, E.M., Flickinger, D., Oepen, S.: The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In: Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics, pp. 8–14 (2002)

16. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL), pp. 149–164 (2006)
17. Blevins, J.P.: Word-based morphology. Journal of Linguistics 42, 531–573 (2006)
18. Mel'čuk, I.: Dependency Syntax: Theory and Practice. State University of New York Press (1988)
19. McDonald, R., Nivre, J.: Characterizing the errors of data-driven dependency parsing models. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 122–131 (2007)
20. Tesnière, L.: Éléments de syntaxe structurale. Editions Klincksieck (1959)

Deletions and Node Reconstructions in a Dependency-Based Multilevel Annotation Scheme

Jan Hajič, Eva Hajičová, Marie Mikulová, Jiří Mírovský,
Jarmila Panevová, and Daniel Zeman

Charles University in Prague, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics, Prague, Czech Republic
{hajic,hajicova,mikulova,mirovsky,panevova,
zeman}@ufal.mff.cuni.cz

Abstract. The aim of the present contribution is to put under scrutiny the ways in which the so-called deletions of elements in the surface shape of the sentence are treated in syntactically annotated corpora and to attempt at a categorization of deletions within a multilevel annotation scheme. We explain first (Sect. 1) the motivations of our research into this matter and in Sect. 2 we briefly overview how deletions are treated in some of the advanced annotation schemes for different languages. The core of the paper is Sect. 3, which is devoted to the treatment of deletions and node reconstructions on the two syntactic levels of annotation of the annotation scheme of the Prague Dependency Treebank (PDT). After a short account of PDT relevant for the issue under discussion (Sect. 3.1) and of the treatment of deletions at the level of surface structure of sentences (Sect. 3.2), we concentrate on selected types of reconstructions of the deleted items on the underlying (tectogrammatical) level of PDT (Sect. 3.3). In Section 3.4 we present some statistical data that offer a stimulating and encouraging ground for further investigations, both for linguistic theory and annotation practice. The results and the advantages of the approach applied and further perspectives are summarized in Sect. 4.

1 Motivation and Specification of Deletions (Ellipsis)

Deletion (ellipsis) in language is a long-standing hard problem for all types of theories of formal description of language, and, consequently, also for those who design annotation schemes for language corpora. As such, this phenomenon present in all languages deserves a special attention both from the theoretical viewpoint as well as with regard to empirical studies based on large annotated corpora. Our contribution is based on a dependency-based grammatical theory, on a multilevel treatment of language system and is supported by language data present in the Prague Dependency Treebank for Czech (PDT); when relevant, we also comment upon the English data of the deep-structure annotation of the Wall Street Journal.[1]

[1] A theoretically-oriented analysis of ellipsis from the point of view of dependency grammar is presented in Panevová, Mikulová and Hajičová, to be submitted for DepLing 2015.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 17–31, 2015.
DOI: 10.1007/978-3-319-18111-0_2

Ellipsis is generally defined as an omission of a unit at the surface shape of the sentence that is, however, necessary for the semantic interpretation of the sentence. In other words, ellipsis may be regarded as an empty place in a sentence that has not been occupied by a lexical unit. A similar specification is given by Fillmore (2000) who discusses elements that are represented as "understood but missing" and distinguishes Constructionally Licensed Null Instantiation, Indefinite Null Instantiation, and Definite Null Instantiation, as separate ways of cataloguing the "missing" elements. In a similar vein, Kayne (2005, p.v) speaks about silent elements, that is "elements that despite their lack of phonetic realization seem to have an important role in the syntax of all languages".[2]

With language descriptions working with two syntactic levels, one reflecting the surface shape of the sentence, and one representing the level of deep syntactic structure (linguistic meaning), it is possible to consider an establishment of a new element (node in a tree-like representation of the sentence) in the deep structure tree. From this point of view, two situations may obtain: (i) the newly established ("reconstructed") node on the deep level corresponds to an element that as a matter of fact might have been an element (even if perhaps stylistically awkward) of the surface structure but which has been actually "deleted" (we may call this situation a "textual" deletion/ellipsis), as is the case of *John gave a flower to Mary and [he gave] a book to his son*, or (ii) the grammatical structure of the surface shape of the given sentence does not allow for such an insertion but the semantic interpretation of the sentence requires a node to be present in the deep structure (e.g. the controllee in the constructions with verbs of control, such as *John promised to come* has to be interpreted as *John promised that he=John comes*). This type of ellipsis may be called grammaticalized ellipsis.

2 Treatment of Ellipsis in Some of the Advanced Annotation Schemes for Different Languages

There are not very many studies on ellipsis within the formalism of dependency grammar, and even less frequent are general treatments of this phenomenon in annotation scenarios for corpora. However, as the developers of annotation schemes often have to provide instructions how to deal with such a phenomenon, one can observe some commonalities and differences in schemes for individual languages.

One of the most frequent and complicated types of deletion occurs in coordinated structures in which one element of the structure is missing and for its dependents ("orphan") there is no suitable parent node. Several solutions have been adopted:[3] the "orphans" are "lifted" to a position where their head would have been placed and marked by a special label (similar to the label ExD in the analytical level of PDT, see below Sect.

[2] Such a broad specification of ellipsis allows to include under such a heading also cases of movement or shifting or similar kinds of restructuring, as Chaves (2014) duly notes. However, this is not our concern in the present contribution.

[3] For a detailed analysis of the treatment of coordination in most different dependency treebanks and for the taxonomy of these approaches, see Popel et al. (2013).

3.2). Similar to the PDT treatment is that of the Danish Treebank: the "orphan" is placed where the missing parent node would be and is attached to an existing node and marked by a special label. Thus in the tree for *Skær de rensede løg igennem en gang og derefter i mindre stykker på tværs.* [Cut the cleaned onions through once and then into smaller pieces across.] the node for *derefter* [then] is attached to the conjunction *og* and assigned the label <mod> (i.e. there is no copy of the verb *skær*). In a similar vein, the phrases *i mindre stykker* and *på tværs* are attached to the conjunction and labelled as <avobj> and as <mod>, respectively. Had their head verb been present, they would be labeled avobj and mod (without the angle brackets).

A different solution is proposed in the Universal Stanford Dependency scheme,[4] in which the "orphan" is attached by means of a special dependency function called remnant to the corresponding dependent of the non-deleted governor. Thus, e.g. in a sentence corresponding to English *John visited Mary and George Eva*, the node for *George* would "depend" on *John*, and *Eva* on *Mary* (and both *John* and *Eva* on the verb *visited*, with their corresponding dependency relations, e.g. Subj, and Obj respectively); such a treatment can be understood as an attempt to "copy" the node of the expressed verb, but would lead to serious non-projectivities; its advantage is that the reconstruction including the identification of the type of dependency would be straightforward.

Another possibility is to establish an independent NULL node that represents the deleted second verb; the "orphans" are then attached to this newly established node. As far as we can say, there is no reference to the first verb and also there are no copies of the lemma etc. of this first node. One example of an insertion of empty heads is the insertion of the Phantom node in the SYNTAGRUS corpus for Russian; another example is the Turku Dependency Treebank of Finnish (Haverinen et al. 2010).[5] The same is true about the Hindi Treebank (Husain et al. 2010).

In the dependency treebank of Russian, SYNTAGRUS (Boguslavsky et al. 2009)[6], one sentence token basically corresponds to one node in the dependency tree. There is, however, a noticeable number of exceptions, one of which concerns so-called Phantom nodes for the representation of those cases of deletions of heads that do not correspond to any particular token in the sentence; e.g. *ja kupil rubashku, a on galstuk*

[4] The "remnant" analysis adopted in the Universal Stanford Dependencies is discussed briefly in de Marneffe et al. (2014).

[5] E.g. in *Liikettä ei ole, ei *null* toimintaa.* [There is no movement, no action.] the copula *ole* (the negative verb *ei* is attached similarly to negative particles in other languages) is the root which in turn is the head of the node *null* the type of relation being `conj` (a "Stanford" style of coordination). Attached to this null node is the second negative particle *ei* (as `neg`) and *toimintaa* (as `nsubj`).

[6] SYNTAGRUS currently contains about 40,000 sentences (roughly 520,000 words) belonging to texts from a variety of genres and is steadily growing. It is the only corpus of Russian supplied with comprehensive morphological and syntactic annotation. The latter is presented in the form of a full dependency tree provided for every sentence; nodes represent words annotated with parts of speech and morphological features, while arcs are labeled with syntactic dependency types. There are over 65 distinct dependency labels in the treebank, half of which are taken from Meaning-Text Theory (see e.g. Mel'čuk, 1988).

[I bought a shirt and he a tie], which is expanded into *ja kupil rubashku, a on kupil.*PHANTOM *galstuk*. A Phantom node gets a morphological tag by which it is characterized. In the version of SYNTAGRUS discussed in Nivre et al. (2008), out of the 32000 sentences 478 sentences (1.5%) contained Phantom nodes, and there were 631 Phantom nodes in total. Phantom nodes may be introduced also for cases other than coordination: e.g. the missing copula in *Kak #Phantom vasha familija?* [What PHANTOM your name], *bojatsja otvetsvennosti kak cherty #Phantom ladana* [They fear responsibility as devils PHANTOM incense].

A different situation occurs when a sentence element present in the surface is understood as a modification of more than a single element (a shared modification), as in *John bought and ate an apple*. Here *John* modifies the two conjuncts as their subject. Several strategies are applied in different treebanks: in the "Prague" style treebanks the shared modification is attached to the head of the coordination, mostly a node representing the conjunction, and it is marked in some way to be distinguished from other nodes of the coordination; in the "Stanford" and "Mel'čuk" styles the first conjunct of the coordination is considered to be the head of the coordination.[7]

In the German TIGER Treebank[8], the elided (i.e. borrowed, copied) constituents in coordinate clauses are represented by so-called secondary edges, also labelled with a grammatical function. This feature facilitates well-targeted extraction of syntactic trees that embody various types of coordinate ellipsis. (Secondary edges are represented by curved arrows in TIGER tree diagrams.) According to Brants et al. (2004: p. 599), "secondary edges are only employed for the annotation of coordinated sentences and verb phrases". Nevertheless, secondary edges occasionally turn up as parts of non-clausal coordination types; however, ellipsis in non-clausal coordinate structures was not annotated systematically.

Deletions occurring in the so-called pro-drop languages and conditioned by the fact that the occurrence of subjects in sentences can be omitted are treated either by reflecting the surface structure, with no additional node inserted in the representation of the sentence (this treatment is present in the treebanks of Italian, Portuguese and Hindi, and also in the analytical level of PDT and in other "Prague" style treebanks), or a new node is established (depending on the verb that lacks a subject in the surface shape of the sentence) as the subject of that verb and marked by a morphological tag for pronouns. See the Spanish *La mujer toma riendas que _ nunca usó* [The woman

[7] We refer to these two "styles" without describing them in detail but we assume that it is clear from the context which treebanks are referred to.

[8] The TIGER Treebank (Release 2) contains 50,474 German syntactically annotated sentences from a German newspaper corpus (Brants et al., 2004). The annotation scheme uses many clause-level grammatical functions (subject, direct and indirect object, complement, modifier, etc.) represented as edge labels in the sentence diagrams). As reported in Harbush and Kempen (2007), the total of 7,194 corpus sentences (about 14 percent) include at least one clausal coordination, and in more than half of these (4,046) one or more constituents have been elided and need to be borrowed from the other conjunct.

takes reins that [missing:she] never used] (Taulé et al. 2008).[9] A similar approach is reflected in the tectogrammatical level of PDT (see Sect. 3.1 below).

A special category that may be also placed under the notion of ellipsis is represented by independent sentences without a predicate, headings etc. In most treebanks, there is just one non-dependent node, usually labeled ROOT. The label does not distinguish whether this node is a deleted verb, a noun or some other POS. Some treebanks, e.g. the Floresta sintá(c)tica treebank of Portuguese (Afonso et al. 2002), introduces the label UTT for the root of non-verbal sentences. If the root may have more than a single child that would be attached to the missing verb (see the Czech *Majitelé rodinných domků* [omitted:*zaplatí*] *ještě více, pokud topí např. koksem* [The owners of family houses [omitted: will pay] still more if [omitted:they] heat e.g. with coke], no unified treatment can be found.

As can be seen from the above very cursory overview, most annotation schemes that work with a single level of syntactic annotation are inclined to adopt the strategy not to reconstruct nodes in the trees unless such a strategy prevents to capture rather complex sentence structures, or, taken from the opposite angle, they allow for reconstructions of nodes when this reconstruction is evident and well definable (as with omitted subjects and so on). It is no wonder then that in those types of ellipsis in which there is no evident position in the surface structure where a reconstructed node would be placed the treebanks capturing the surface shape of sentences ignore the fact that reconstructions would lead to a more transparent (semantic) interpretation of the sentence. This is the case e.g. with structures with control verbs (e.g. *John decided to leave = John decided that [John=he] leaves*), structures with some type of general modification (e.g. *This book is easy to read*), etc. The usability of a multilevel annotation scheme, with annotations of the surface shape of the sentence and with those of its deep syntactic structure, can be well demonstrated on the parallel annotation of the Prague Czech-English Dependency Treebank (PCEDT),[10] with a two-level annotation of Czech and English; the original English texts are taken from the Penn Treebank, translated to Czech and analyzed, both for Czech and for English, by using the Prague PDT-style of annotation. The same philosophy of annotation has been successfully applied to both sides, namely to reconstruct all missing nodes in the deep syntactic (tectogrammatical) level of annotation that are necessary for a correct interpretation of the meaning of the sentence (see Sect. 3.3 below), except for some very specific types of English constructions that are not present in Czech.

[9] In constructions with modal verbs plus infinitive only a single subject is reconstructed hanging on the infinitive which is also supposed to be the head of the finite verb. Ex. *Puedo afirmar mucho de su trayectoria intelectual* [I can confirm much of his intellectual trajectory].

[10] See Hajič et al. (2012).

3 Ellipsis and Node Reconstruction in the Prague Dependency Treebank

3.1 The Prague Dependency Treebank

The Prague Dependency Treebank (referred to as PDT in the sequel) is an effort inspired by the Penn Treebank; the work started as soon as in the mid-nineties and the overall scheme was published already in 1998 (see e.g. Hajič 1998). The basic idea was to build a corpus annotated not only with respect to the part-of-speech tags and some kind of (surface) sentence structure but capturing also the syntactico-semantic, underlying structure of sentences. Emphasis was put on several specific features:

(i) the annotation scheme is based on a solid, well-developed theory of an integrated language description, formulated in the 1960s and known as Functional Generative Description,

(ii) the annotation scheme is "natively" dependency-based, and the annotation is manual,

(iii) the "deep" syntactic dependency structure (with several semantically-oriented features, called "tectogrammatical" level of annotation) has been conceptually and physically separated from the surface dependency structure and its annotation, with full alignment between the elements (tree nodes) of both annotation levels being kept,

(iv) the basic features of the information structure of the sentence (its topic-focus articulation, TFA) have been included, as a component part of the tectogrammatical annotation level,

(v) from the very beginning, both the annotation process and its results have been envisaged, among other possible applications, as a good test of the underlying linguistic theory.

The Prague Dependency Treebank consists of continuous Czech texts mostly of the journalistic style (taken from the Czech National Corpus) analyzed on three levels of annotation (morphological, surface syntactic shape and deep syntactic structure). At present, the total number of documents annotated on all the three levels is 3,168, amounting to 49,442 sentences and 833,357 (occurrences of) nodes. The PDT version 1.0 (with the annotation of only morphology and the surface dependencies) is available from the Linguistic Data Consortium, as is the PDT version 2.0 (with the annotation of the tectogrammatical level added). Other additions (such as discourse annotation) appeared in PDT 2.5 and in PDT 3.0, which are both available from the LINDAT/CLARIN[11] repository (Bejček et al. 2013).

The annotation scheme has a multilevel architecture: (a) morphological level: all elements (tokens) of the sentence get a lemma and a (disambiguated) morphological tag, (b) analytical level: a dependency tree capturing surface syntactic relations such as subject, object, adverbial: all edges of the dependency tree are labeled with a (structural) tag, and (c) tectogrammatical level capturing the deep syntactic relations: the dependency structure of a sentence is a tree consisting of nodes only for

[11] http://lindat.cz

autonomous meaningful units, called "autosemantic" units or elements; function words such as prepositions, conjunctions, auxiliary verbs etc. are not included as separate nodes in the structure, their contribution to the meaning of the sentence is captured by complex symbols of the autonomous units. The edges of the tree are interpreted as deep syntactic relations such as Actor, Patient, Addressee, different kinds of circumstantial relations etc.; each node carries also one of the values of contextual boundness on the basis of which the topic and the focus of the sentence can be determined. Pronominal coreference is also annotated.[12]

In addition to the above-mentioned three annotation levels in the PDT there is also one non-annotation level, representing the "raw-text". On this level, called word level, the text is segmented into documents and paragraphs and individual tokens are recognized and associated with unique identifiers (for easy and unique reference from the higher annotation levels).

Crucial for the discussion of the issue of ellipsis is the difference between the two syntactic levels, the analytical (with analytical tree structures, ATSs in the sequel) and the tectogrammatical (with tectogrammatical tree structures as representations of sentences, TGTSs) one. In the ATSs all and only those nodes occur that have a lexical realization in the surface shape of the sentence (be they auxiliaries or autonomous lexical units) and also nodes that represent the punctuation marks of all kinds. No insertions of other nodes are possible (with the exception of the root node identifying the tree in the set). In contrast, the TGTS contains nodes for autosemantic lexical units only, but they might be complemented by newly established (reconstructed) nodes for elements that correspond to deletions in the surface structure. A comparison of the ATS and the TGTS of a particular sentence and of TGTS's of most different sentence structures with different types of newly established nodes makes it possible to categorize the reconstructions and analyze them as for their characteristics and statistics, which is the core of our contribution.

3.2 Deletions in the Representation of the Surface Shape of the Sentence

With the approach to ellipsis described above, one issue has to be raised with respect to the ATS. The problem arises if a node representing some element occurring in the surface shape of the sentence has not an appropriate governor in that structure, i.e. there is no node on which the given node depends. A specific label ExD (Extra-Dependency) is introduced to mark such a "pseudo-depending" node. The position of the node with the label ExD in the ATS is governed by specific instructions; basically, it is placed in a position in which the missing governor would be placed (see Sect. 2 for similar approaches).

[12] In the process of the further development of the PDT, additional information is being added to the original one in the follow-up versions of PDT, such as the annotation of basic relations of textual coreference and of discourse relations, multiword expressions etc.

3.3 Reconstructions of Nodes on the Tectogrammatical Level

3.3.1 Treatment of ellipsis on the analytical and tectogrammatical levels of PDT is illustrated in Fig. 1. The ATS structure is displayed in Fig. 1a (left side), where the deletions are not reconstructed (for the pseudo-dependency within a shortened comparative construction the node ExD is used), whereas in the corresponding TGTS in Fig. 1b (right side) the generalized ACT (#Gen.ACT) is established as dependent on the main verb.[13] Another node for #Gen.ACT is newly established in the reduced comparative construction; the full (expanded) shape of the embedded sentence includes both the comparison (CPR) for the whole comparison construction as well as its local modification. Figures 1a, 1b may also help to compare the number of nodes in the ATS structure (with the function words represented by specific nodes) and the number of nodes in TGTS (with the omission of the function words and with the addition of the nodes for the elements deleted on the surface).

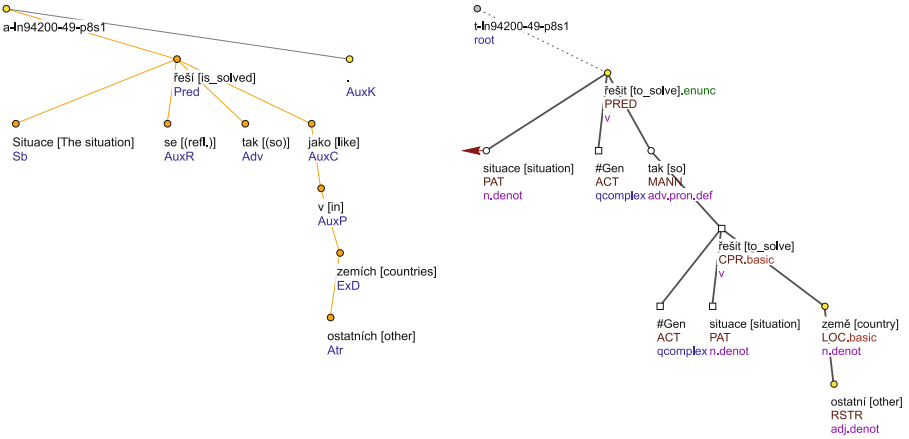


Fig. 1. Situace se řeší tak jako v ostatních zemích.
[The situation is solved like in other countries.]

3.3.2 All syntactically annotated corpora share the problem of reflecting the gaps in coordination constructions. This problem is multiplied by the fact that there exist several types of deletions. In Fig. 2, the omitted noun for one of the conjuncts within coordination in the nominal group is restored by copying the node *podnikání* [enterprise]. (For some properties of the copied nodes, see Sect. 3.4 below.)

[13] The reconstructed nodes in the trees are represented as squares rather than as circles.

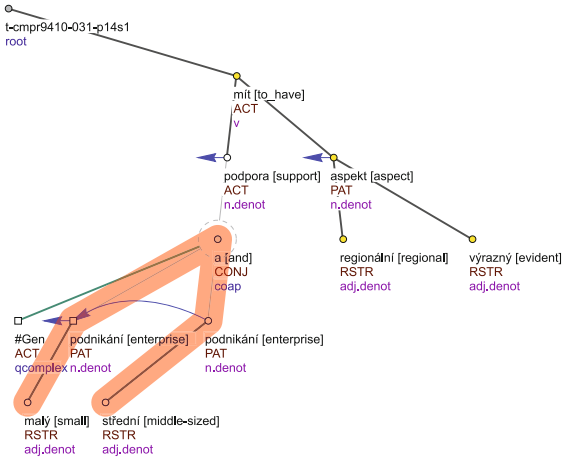


Fig. 2. Podpora malého a středního podnikání má výrazný regionální aspekt.
[Support of small and middle-sized enterprises has an evident regional aspect.]

3.3.3 Fig. 3 exemplifies the PDT treatment of the deletion of the identical predicate in the disjunctive coordination by means of the copied node *dít se* [to_happen]. The shared Actor (Subject) for both clauses is demonstrated here, too. (The treatment of sentence negation present in Fig. 3 is explained below in Sect. 3.3.6.)

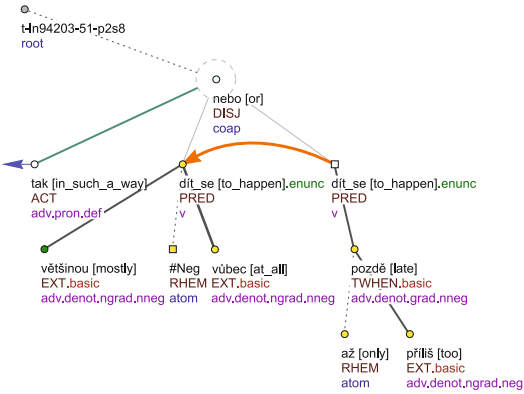


Fig. 3. Většinou se tak neděje vůbec nebo až příliš pozdě.
[Mostly it does not happen in such a way at all or only too late.]

3.3.4 In Fig. 4 the structure of the sentence with missing predicate as the root of the sentence is illustrated. Since the lemma cannot be identified from the context, the node #EmpVerb is established rather than a node with a concrete lemma.

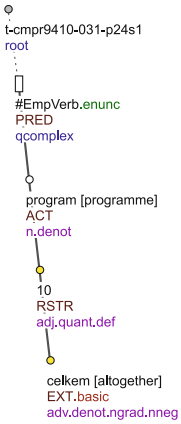


Fig. 4. Celkem 10 programů
[Altogether 10 programmes]

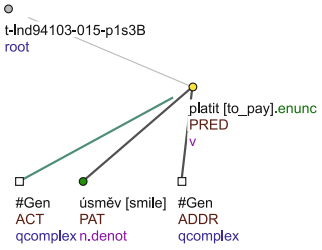


Fig. 5. Za úsměv se platí.
[For smile one pays.]

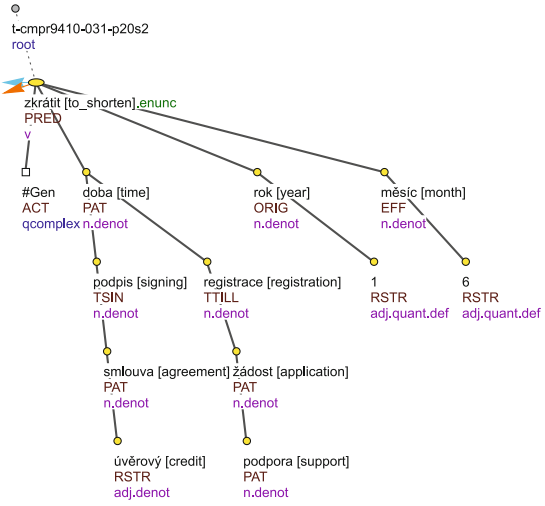


Fig. 6. Byla zkrácena doba od podpisu úvěrové smlouvy k registraci žádosti o podporu z 1 roku na 6 měsíců.
[The time from signing the credit agreement to the registration of the application for support was shortened from 1 year to 6 months.]

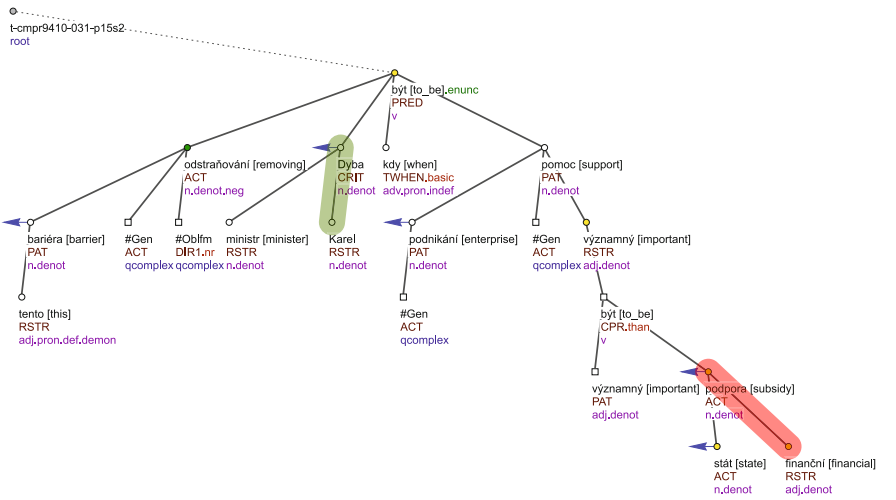


Fig. 7. Odstraňování těchto bariér může být podle ministra Karla Dyby někdy významnější pomocí podnikání než finanční podpora státu.
[Removing of these barriers may be according to minister Karel Dyba sometimes more important support of enterprises than a financial subsidy from the state.]

3.3.5 The generalization of the Actor and of other valency members (participants and some adjuncts) belongs to frequent phenomena in the PDT. For the generalization of ACT (#Gen.ACT), there is a special form in Czech (called deagentization, or, in older tradition, reflexive passive), see Fig. 5 and Fig. 1. General ACT often occurs in passive sentences (see Fig. 6). The generalization of other participants and modifiers missing in the surface shape of the sentence is handled in the TGTS's by added nodes with the lemmas #Gen and their corresponding functions (PAT, ADDR etc.); #Oblfm is used as the lemma for a generalized adjunct. General Actor depending on a deverbal noun is present in Fig. 7, the local modification (LOC) specified as "from where (DIR1)" is annotated here as an obligatory modifier of the noun *odstraňování* [removal].

3.3.6 In Fig. 8, three types of an insertion of a new node are present: (a) The arrow from the newly established node #PersPron.ACT standing for the deleted Actor indicates that the deleted Actor is present in the preceding context. (b) The missing head of the first conjunct *záležitost* [matter] within the noun group is inserted as a copy. (c) In case of sentential negation formed in Czech by a prefix *ne-* attached to the positive form of the verb (*vidí* = he sees, *nevidí* = he does not see) a new node labelled #Neg and attached a functor RHEM is established depending on the verb (the lemma of which is the positive form of the verb). The position of the #Neg node with regard to the verb and other nodes depending on the given verb indicates the (semantic) scope of negation, which in a general case does not necessarily include the verb.

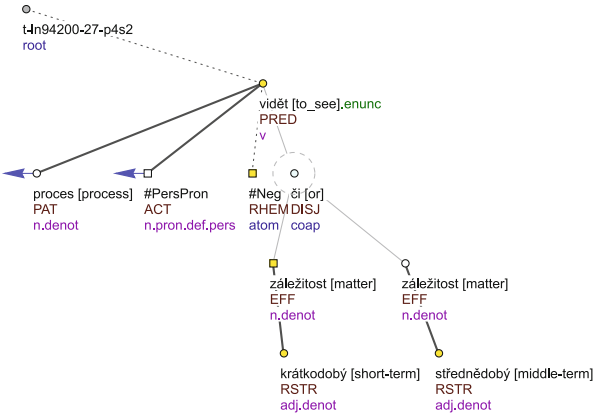


Fig. 8. Proces nevidí jako krátkodobou či střednědobou záležitost.
[He does not see the process as a short-term or a middle-term matter.]

3.3.7 The predicate *lze* [it is possible] is connected with the relation of control. In the given sentence (Fig. 9) the Benefactor functions as the controller (generalized #Gen.BEN). Its Actor fills the role of the controllee and is represented by the node #Cor indicating the grammatical coreference required by the underlying structure of infinitive constructions.

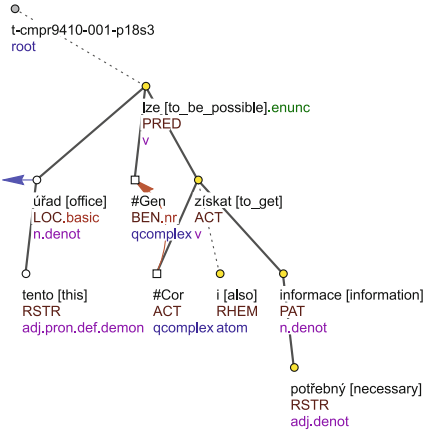


Fig. 9. Na tomto úřadě lze získat i potřebné informace.
[At this office it is possible to get also the necessary information.]

3.4 Some Simple Statistics

The existence of syntactic annotations on two levels of sentence structure allows for some interesting statistical comparisons. Out of the total of 43,955 sentences of the PDT 3.0 training + dtest data (9/10 of the whole PDT) there are 29,243 sentences with

a newly generated node with a t-lemma label of reconstructed nodes and 4,154 sentences with a reconstructed copied node (mostly in coordination structures).

There is a total of 65,593 occurrences of newly generated nodes of the former category (their t-lemma starts with #). The reconstruction of nodes for General Participants prevails rather significantly (see Figs. 5 through 7), followed by cases of reconstructions of nodes mostly for textual deletions in which case the new node labeled as #PersPron has a counterpart in the preceding context (see Fig. 8); these two groups account for 41,136 cases. The next most frequent group (7,476) covers a reconstruction of the controllee in so-called "control" structures (see Fig. 9). The third group relates to negation (7,647 cases), which is more or less a formal reconstruction, though important from the semantic point of view as mentioned above (see Figs. 3 and 8). The categories at the bottom of the frequency list are of a more or less technical character: the label #Forn (1,495) for foreign words or the label #Idph (754) for idiomatic phrases, or they belong to rather specific cases. In between there are three categories that are theoretically biased and given – similarly as #Gen – by the respective verbal valency frames: #Oblfm for semantically obligatory modifications of verbs (1,927 occurrences, see Fig 7), #Unsp for Actor with an unspecified reference without a counterpart on the surface (201 occurrences), and #Rcp for reciprocal constructions (994 occurrences). There is a total of 3,539 nodes for reconstructed root nodes without a lexical label (#EmpVerb, see Fig. 4, and #EmpNoun).

The category of an insertion of so-called copied nodes applies especially in the cases of coordination (see Figs. 2, 3 and 8). In the given set of data, there is a total of 6,799 newly established copied nodes, out of which there are 5,988 cases copied from the same sentence and 811 cases copied from a different sentence. The newly established node is inserted into a position in which it should be placed in the tectogrammatical structure. Both the original node and the copied one refer to the same lexical counterpart on the analytical level (ATS), which is to say that a copied node shares with the "original" node the t-lemma. As for the values of other attributes relevant for the given node, there is a list of those that are copied unchanged together with the t-lemma (e.g. the values for gender, aspect, iterativeness with verbs etc.). However, values other that those given by the list may be changed by the annotator to correspond to their actual value corresponding to the context of the newly established node. This concerns e.g. the values of functors: out of the total of 6,799 newly established nodes 5,027 of them share the value of the functor with the original node, and in 1,772 cases the functors are different. There are 197 pairs of different functors (original – copy), and it is interesting to note that among the first 20 of most frequent pairs (with 1,584 occurrences), in 905 cases (more than 57%) the copied node gets the functor CPR for the relation of comparison (e.g. PRED – CPR, see Fig. 1b).

It was a general principle that any newly established node (i.e. a node not expressed in the surface shape of the sentence) should get the TFA value 't' for a contextually bound element. This default assignment is based on the intuitive assumption that such a node deleted on the surface should refer to a piece of information which has already been in the previous context. However, the annotators were offered the possibility to change the TFA value according to the actual TFA

structure of the sentence. To confirm the validity of the default assignment, we have checked the data in the set of sentences with copied nodes and have found out that in 855 cases the annotators considered necessary to change the default 't' value into the 'f' value (for contextually non-bound). Having checked these sentences carefully, the largest group consisted of coordination of the type *Proces nevidí jako krátkodobou či střednědobou záležitost* [He does not see the process as a short-term or mid-term matter]: here (see Fig. 8 above) the newly established node copies the lemma *záležitost* [matter], shares the functor EFF with the original node (somebody sees something as a matter) and is also a part of the contextually non-bound information (the sentence communicates about the process and says that it is not seen as a short-term and middle-term matter). It follows that the inserted new node *záležitost* [matter] should get the TFA value 'f'. In few cases, the newly inserted node has been considered as a contrastive contextually bound node and marked as such by 'c' see e.g. *I u průmyslové a stavební výroby nejlepší výsledky dociluje polská ekonomika* [Also with the industrial and building production the best results are achieved by Polish economics]. If the reduced coordination constructions are compared with full constructions even on the surface, the element in question would get these values.

4 Summary and Outlook

The problem of ellipsis, the reconstruction of which is triggered by the context or by the type of syntactic structure, is shared by all languages though the rules for the treatment of deletions and their reconstruction may be language specific; this phenomenon represents a difficult issue for syntactic annotation of sentences as well. In our contribution we have focused on the treatment of ellipsis on two levels of syntactic representation based on dependencies, namely on the analytic (surface) one and on the deep (tectogrammatical) one as present in the Prague Dependency Treebank (PDT). We have attempted at a classification of types of ellipsis as reflected in the PDT scenario documenting that each type requires a different treatment in order to achieve an appropriate semantic interpretation of the surface structures in which ellipsis is present. In this way and also by comparing such a scenario with mono-level ones, we wanted to demonstrate the advantages of a corpus scenario reflecting two levels of syntactic structure (surface and deep) separately but with pointers (references/links) which make it possible to search in both levels simultaneously.

The preliminary classification of the types of ellipsis and the data about their frequency drawn from the PDT as presented in this contribution opens new stimuli for more subtle theoretical studies of the relations between surface and deep structure of sentences, of their relations in discourse, and it serves as a great challenge for an explanation of their conditions and sources.

Acknowledgments. We gratefully acknowledge support from the Grant Agency of the Czech Republic (projects n. P406/12/0658, 15-10472S and P406/12/0557) and the Ministry of Education of the Czech Republic (project LM2010013 – LINDAT/CLARIN). This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project.

References

1. Afonso, S., Bick, E., Haber, R., Santos, D.: Floresta sintá(c)tica: a treebank for Portuguese. In: Proc. of LREC 2002(2002)
2. Bejček, E., Hajičová, E., Hajič, J., et al.: Prague Dependency Treebank 3.0. Data/software, Univerzita Karlova v Praze, MFF, ÚFAL, Prague, Czech Republic (2013), http://ufal. mff.cuni.cz/pdt3.0/
3. Boguslavsky, I., et al.: Development of a Russian Tagged Corpus with Lexical and Functional Annotation. In: Proc. of Metalanguage and Encoding Scheme Design for Digital Lexicography. MONDILEX Third Open Workshop, Bratislava, Slovakia, pp. 83–90 (2009)
4. Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: Linguistic Interpretation of a German Corpus. Research on Language and Computation 2, 597–620 (2004)
5. Chaves Rui, P.: On the Disunity of Right-node Raising Phenomena: Extraposition, Ellipsis and Deletion. Language 90, 834–886 (2014)
6. de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C.D.: Universal Stanford Dependencies: A cross-linguistic typology. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavík, Iceland, pp. 4585–4592 (2014)
7. Fillmore, C.J.: Silent Anaphora, Corpus, FrameNet and Missing Complements. Paper presented at the TELRI Workshop, Bratislava (November 1999)
8. Hajič, J.: Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In: Issues of Valency and Meaning, Karolinum, Prague, pp. 106–132 (1998)
9. Hajič, J., Hajičová, E., Panevová, J., et al.: Announcing Prague Czech-English Dependency Treebank 2.0. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), İstanbul, Turkey, pp. 3153–3160 (2012)
10. Harbusch, K., Kempen, G.: Clausal coordinate ellipsis in German: The TIGER treebank as a source of evidence. In: Proceedings of NODALIDA 2007 (2007)
11. Haverinen, K., Viljanen, T., Laippala, V., Kohonen, S., Ginter, F., Salakoski, T.: Treebanking Finnish. In: Proceedings of TLT9, pp. 79–90 (2010)
12. Husain, S., Mannem, P., Ambati, B., Gadde, P.: The ICON-2010 Tools Contest on Indian Language Dependency Parsing. In: Proc. of ICON 2010, Kharagpur, India (2010)
13. Kayne, R.S.: Movement and Silence, Oxford University Press (2005)
14. Mel'čuk, I.: Dependency Syntax: Theory and Practice. State University of New York Press (1988)
15. Mikulová, M.: Semantic Representation of Ellipsis in the Prague Dependency Treebanks. In: Proceedings of the Twenty-Sixth Conference on Computational Linguistics and Speech Processing ROCLING XXVI, Taipei, Taiwan, pp. 125–138 (2014)
16. Nivre, J., Boguslavsky, I.M., Iomdin, L.L.: Parsing the SynTagRus treebank of Russian. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 641–648. Association for Computational Linguistics (2008)
17. Panevová, J., Mikulová, M.: Assimetrii mezhdu glubinným i poverxnostnym predstavleniem predlozhenija (na primere dvux tipov obstojatel'stv v cheshskom jazyke). In: Apresjan, J.D., et al. (eds.): Smysly, teksty i drugie zachvatyvajushchie sjuzhety. Sbornik statej v chest'80-letija I. A. Mel'čuk, pp. 486 – 499. Jazyki slavjanskoj kul'tury, Moscow (2012)
18. Popel, M., Mareček, D., Štěpánek, J., Zeman, D., Žabokrtský, Z.: Coordination Structures in Dependency Treebanks. In: Proceedings of ACL, Sofia, Bulgaria (2013)
19. Taulé, M., Martí, M.A., Recasens, M.: AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In: Proc. of LREC 2008 (2008)

Enriching, Editing, and Representing Interlinear Glossed Text

Fei Xia[1], Michael Wayne Goodman[1], Ryan Georgi[1], Glenn Slayden[1], and William D. Lewis[2]

[1] Linguistics Department, University of Washington, Seattle, WA 98195, USA
{fxia,goodmami,rgeorgi,gslayden}@uw.edu
[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
wilewis@microsoft.com

Abstract. The majority of the world's languages have little to no NLP resources or tools. This is due to a lack of training data ("resources") over which tools, such as taggers or parsers, can be trained. In recent years, there have been increasing efforts to apply NLP methods to a much broader swathe of the worlds languages. In many cases this involves bootstrapping the learning process with enriched or partially enriched resources. One promising line of research involves the use of Interlinear Glossed Text (IGT), a very common form of annotated data used in the field of linguistics. Although IGT is generally very richly annotated, and can be enriched even further (e.g., through structural projection), much of the content is not easily consumable by machines since it remains "trapped" in linguistic scholarly documents and in human readable form. In this paper, we introduce several tools that make IGT more accessible and consumable by NLP researchers.

1 Introduction

Of the world's 7,000+ spoken languages, only a very small fraction have text resources substantial enough to allow for the training of NLP tools, such as part-of-speech (POS) taggers and parsers. Developing enriched resources, e.g., treebanks and POS-tagged corpora, which allow supervised training of such tools, is expensive and time-consuming. In recent years, work has been done to bootstrap the development of such resources for resource-poor languages by tapping the enriched content of a better resourced language and, through some form of alignment, "projecting" annotations onto data for the resource-poor language. Some studies have focused on the typological similarity of languages, using cognates and similar word forms in typologically similar languages, to bridge between languages and to build tools and resources [1,2]. Other work has relied on parallel corpora, where one language of a corpus is highly resourced, and annotations are projected onto the lesser resourced language(s) [3,4]. A third line of work is to use linguistically annotated data, specifically Interlinear Glossed Text (IGT), a data format very commonly used in the field of linguistics, to project annotations from a highly resourced language to one or more under-resourced languages, potentially hundreds at a time [5,6].

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 32–46, 2015.
DOI: 10.1007/978-3-319-18111-0_3

Building upon the previous studies on IGT, we see the potential for bootstrapping or training up tools for a larger number of the world's languages. Since IGT is common in the field of linguistics, and because linguists study thousands of the world's languages, the possibility exists to build resources for a sizable percentage of the world's languages. The problem is that IGT is typically locked away in scholarly linguistic papers, and not easily accessible to NLP researchers who might otherwise want access to the data. The Online Database of Interlinear text (ODIN) [7], a database of over 200,000 instances of IGT for more than 1,500 languages, tackles the issue of extracting IGT from scholarly resources, but focuses more on presenting the captured content for human consumption and query. By taking the content of ODIN, enriching it (e.g., through projected annotations), and reformatting it into a machine readable form, enriched IGT becomes a much more useful resource for bootstrapping NLP tools.

In this paper, we first describe the raw (or original) IGT used by linguists and the enriched IGT which is more relevant to the NLP field. Then we outline a data format for representing enriched called *Xigt*. Next, we introduce two packages that we have developed for processing IGT: the first one enriches raw IGT automatically, and the second one is a graphic editor which the annotators can use to edit enrich IGT in the Xigt format. By making these tools, and the resulting data, available to the NLP community, we open the door to a much wider panoply of the world's languages for NLP research.

2 Interlinear Glossed Text

IGT is a common format that linguists use to present language data relevant to a particular analysis. It is most commonly presented in a three-line canonical form, a sample of which is shown in (1). The first line, the *language line*, gives data for the language in question, and is either phonetically encoded or transcribed in the language's native orthography. The second line, the *gloss line*, contains a morpheme-by-morpheme or word-by-word gloss for the data on the language line. The third line, the *translation line*, contains a translation of the first line, often into a resource-rich language such as English. There could be additional lines showing other information such as a citation and a language name and/or code. In Ex (1), *(Bailyn, 2001)* is the source of the IGT instance, referring to [8]; *cym* is the language code for Welsh.

(1) Rhoddodd yr athro lyfr i'r bachgen ddoe
gave-3sg the teacher book to-the boy yesterday
"The teacher gave a book to the boy yesterday" (Bailyn, 2001) [cym]

2.1 Collecting IGT

In linguistics, the practice of presenting language data in interlinear form has a long history, going back at least to the time of the structuralists. IGT is often used to present data and analysis on a language that the reader may not know

much about, and is frequently included in scholarly linguistic documents. ODIN, the Online Database of INterlinear text, is the result of an effort to collect IGT instances in scholarly documents posted to the Web [9,7]. It currently contains approximately 200,000 IGT instances from over 1500 languages.

2.2 Enriching IGT

The unique structure of IGT makes it an extremely rich source of information for resource-poor languages: Implicit in an IGT instance is not only a short bitext between that language and a language of wider communication (almost universally English, but instances of Spanish and German have been discovered as well), but also information encoded in the so-called *gloss line* about the grammatical morphemes in the source language and word-by-word translations to lemmas of the translation language. Thus even small quantities of IGT could be used to bootstrap tools for resource-poor languages through structural projection [3,10]. However, bootstrapping tools often require the raw IGT to be enriched first. The enrichment process normally contains the following two steps.

Cleaning and Normalizing IGT Instances: The process of collecting IGT from linguistic document may introduce noise. For instance, ODIN uses an off-the-shelf converter to convert pdf documents into text format, and the converter sometimes wrongly splits a language line into two lines. One such an example is Ex (2) from [11], where the language line is incorrectly split into two lines by the converter.

```
(2) Haitian CF (Lefebvre 1998:165)
                    ak
    Jani   pale          lii/j
    John   speak  with    he
    (a) 'John speaks with him' (b) 'John
        speaks with himself'
```

Furthermore, the raw IGT is often not in the three-line canonical form. For instance, an IGT instance often contains other information such as a language name, a citation, and so on. In Ex (2), the first line contains the language name and citation,[1] the third line includes coindexes *i* and *i/j*, and the last two lines show two possible translations of the sentence.

The cleaning and normalization step aims at fixing errors that were introduced when IGT was extracted from the linguistic documents, separating out various fields in an IGT, normalizing each field, and storing the results in a uniform data structure. Ex (3) shows the resulting IGT after this step. Noticing that coindexes *i* and *j* are removed from the language line and stored in a separate field, and the wrongly split language lines are merged back.

[1] *CF* here stands for French-lexified creole.

```
(3) Language: Haitian CF
    Citation: (Lefebvre 1998:165)
    L:        Jan   pale   ak     li
    Coindx:   (Jan, i), (li, i/j)
    G:        John  speak  with   he
    T1:       John speaks with   him
    T2:       John speaks with himself
```

Adding Word Alignment and Syntactic Structure: After IGT has been cleaned and normalized, the next step is to add word alignment and syntactic structure. For word alignment, if the IGT instance is clean, the alignment between the language line and the gloss line is implicit from the layout (i.e., the i-th tokens in the two lines align to each other). The alignment between the gloss line and the translation line can be obtained by running automatic word aligner such as GIZA++ [12] or using some heuristics (e.g., aligning words with the same spelling or stem). The common way for getting syntactic structure is to parse the translation line with an English parser, and then project that parse tree to the language line via the word alignment [10]. Given the IGT in Ex (1), the algorithm will produce the word alignment in Fig 1, the syntactic structures in Fig 2.

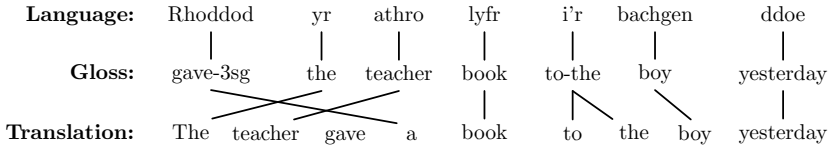


Fig. 1. Aligning the three lines in an IGT instance

2.3 Using Enriched IGT

Enriched IGT can help linguistic studies and NLP in many ways. For instance, linguists can search an IGT database for sentences with certain linguistic constructions (e.g., passives, conditionals, double object structures). Enriched IGT also allows discovery of computationally relevant typological features (e.g., word order or the presence or absence of particular grammatical markers), and does so with high accuracy [13,14]. Furthermore, enriched IGT can also be used to bootstrap NLP tools for resource-poor languages; for instance, adding features extracted from projected syntactic structures to a statistical parser provided a significant boost to parser performance [5].

3 Xigt: An XML Representation of the Enriched IGT

A human can read a textual IGT from ODIN and understand the structure of annotations, but a computer only sees strings of characters and spaces. We can—and

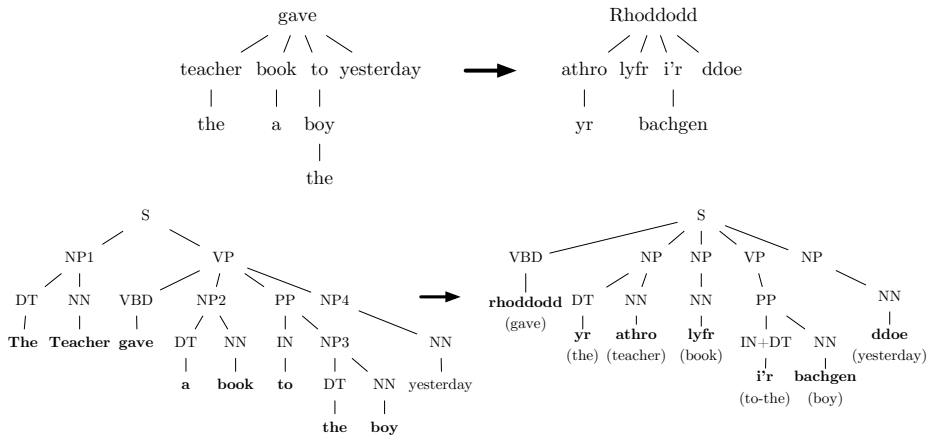


Fig. 2. Projecting dependency and phrase structure from the translation line to the language line

do, for the enrichment process—write code to interpret the spacing as delimiters for groups of related tokens, but this is unreliable in noisy data, and moreover cannot handle alignments that do not arrange vertically. Once we have initially analyzed the space-delimited structure of a textual IGT, we store the information in a structured data format so a computer can more easily understand the structure for later tasks.

Goodman et al. [15] introduced a new data model for IGT, called Xigt, that allows for complex annotation alignments, which is useful for encoding the enriched IGT.[2] Key features of Xigt include a relatively flat structure (tiers are represented side-by-side, not nested) and the use of IDs and references for annotations. There are only four structural elements: `<xigt-corpus>`, `<igt>`, `<tier>`, and `<item>`. Xigt also introduces a new referencing system called "alignment expressions" that allows annotations to have multiple targets, or to select sub-spans (e.g., character spans) from targets. These features allow for novel or complex annotation types, alternative analyses, and the ability to add new annotations without changing the structure of previous ones.

Xigt's generality and expressiveness allow for rich representations of many kinds of annotations, but at the same time the lack of hard constraints allows the same data to be represented in multiple ways. For example, *words* may be specified as segments of a *phrase* (i.e. a kind of annotation of the phrase) or as primary data (unaligned and explicitly specifying the form of the word); *glosses* may align to *morphemes*, *words*, or *phrases*, depending on the availability of these tiers. Our IGT enrichment package (INTENT) and IGT editor (XigtEdit)

[2] While Xigt itself is the data model, it has a canonical XML serialization format called XigtXML. For the sake of simplicity, in this paper we will not make such a distinction and instead use the same name for both.

need more specifications in order to efficiently process Xigt-encoded corpora, so we establish a set of conventions on top of Xigt that our data abide by.

In this section we outline the conventions for our data. The Xigt project includes an API for programmatically interacting with corpora, so we also cover the API functions that are useful for our purposes.

3.1 Representing Enriched IGT in Xigt

To represent the enriched IGT, as described in Section 2.2, in Xigt, we need to extend Xigt in several ways. Figure 3 shows an enriched IGT in this extended format. First, we define some new tier types and constrain how the default ones are used. Collectively, the tiers can be divided into three groups according to the source of information: (1) Original text, (2) Inferred structure, and (3) Additional enrichment.

Group (1): Stores the original text from ODIN IGT so that structural information is encoded as stand-off annotation of it. This is useful, in part, in case the process of enriching IGT changes and we need to regenerate the IGT in Xigt. This group has only one tier type, called *odin*, and each `<item>` on such a tier contains a line from the original IGT. In Figure 3, lines 7–11 encode the raw text, while lines 12–16 encode the text after normalization. The *odin* tier is also used for storing cleaned text (not shown here as it is identical to the raw IGT for this particularly clean example). The `state` attribute specifies the level of processing undergone by each *odin* tier.

Group (2): Encodes the structural annotations that are implicit in the textual IGT. This group only uses the default tier types in Xigt, although we specify some constraints on their usage to aid INTENT and XigtEdit in their processing, and includes:

- *phrases*: representing the language line (lines 17–19)
- *words*: showing the segmentation of the language or translation line (lines 20–25 and 41–45)
- *morphemes*: marking the morpheme boundary within a word (lines 26–32)
- *glosses*: providing word-to-word or morpheme-to-morpheme glosses (lines 33–37)
- *translations*: providing the translation of the language line (lines 38–40)

The structural annotations in Group (2) can be inferred by examining tokens in the tiers from word segmentation (i.e., by spaces) or morpheme segmentation (i.e., first by spaces, then by hyphens or other morpheme delimiters). By definition, in a clean IGT the i^{th} token from a morpheme-segmented gloss line will align to the i^{th} token of a morpheme-segmented language line. In a less clean IGT, these lines may not have the same number of tokens, in which case we back off to aligning word-segmented gloss and language lines. Again, in the case that this doesn't work, we align the unsegmented gloss line to the language line.

```
<?xml version="1.0" encoding="utf-8"?>
<xigt-corpus alignment-method="auto" xml:lang="en">
 <metadata xmlns:olac="http://www.language-archives.org/OLAC/1.1/" ...>
   ...
 </metadata>
 <igt id="i1" doc-id="397" line-range="959 961" tag-types="L G T">
  <tier type="odin" state="raw" id="r">
   <item id="r1" line="959" tag="L"
        >(1) Nay-ka ai-eykey pap-ul mek-i-ess-ta</item>
   <item id="r2" line="960" tag="G"
        >    I-Nom child-Dat rice-Acc eat-Caus-Pst-Dec</item>
   <item id="r3" line="961" tag="T"
        >    'I made the child eat rice.'</item>
  </tier>
  <tier type="odin" state="normalized" id="n" alignment="r">
   <item id="n1" alignment="r1" line="959" tag="L"
        >Nay-ka ai-eykey pap-ul mek-i-ess-ta</item>
   <item id="n2" alignment="r2" line="960" tag="G"
        >I-Nom child-Dat rice-Acc eat-Caus-Pst-Dec</item>
   <item id="n3" alignment="r3" line="961" tag="T"
        >I made the child eat rice.</item>
  </tier>
  <tier type="phrases" id="p" content="n" xml:lang="ko">
   <item id="p1" content="n1"/>
  </tier>
  <tier type="words" id="w" segmentation="p" xml:lang="ko">
   <item id="w1" segmentation="p1[0:6]"/>
   <item id="w2" segmentation="p1[7:15]"/>
   <item id="w3" segmentation="p1[16:22]"/>
   <item id="w4" segmentation="p1[23:35]"/>
  </tier>
  <tier type="morphemes" id="m" segmentation="w" xml:lang="ko">
   <item id="m1.1" segmentation="w1[0:3]"/>
   <item id="m1.2" segmentation="w1[4:6]"/>
   <item id="m2.1" segmentation="w2[0:2]"/>
   <item id="m2.2" segmentation="w2[3:8]"/>
   ...
  </tier>
  <tier type="glosses" id="g" alignment="m" content="n">
   <item id="g1.1" alignment="m1.1" content="n2[0:1]"/>
   <item id="g1.2" alignment="m1.2" content="n2[2:5]"/>
   ...
  </tier>
  <tier type="translations" id="t" alignment="p" content="n">
   <item id="t1" alignment="p1" content="n3"/>
  </tier>
  <tier type="words" id="tw" segmentation="t">
   <item id="tw1" segmentation="t1[0:1]"/>
   <item id="tw2" segmentation="t1[2:6]"/>
   ...
  </tier>
  <tier type="bilingual-alignments" id="a" source="tw" target="g">
   <item id="a1" source="tw1" target="g1.1"/>
   <item id="a2" source="tw2" target="g4.2,g4.3"/>
   ...
  </tier>
  <tier type="dependencies" id="dt" dep="tw" head="tw">
   <item id="dt1" dep="tw1" head="tw2">nsubj</item>
   <item id="dt2" dep="tw2">root</item>
   <item id="dt3" dep="tw3" head="tw4">det</item>
   ...
  </tier>
  ...
 </igt>
 ...
</xigt-corpus>
```

Fig. 3. The Xigt representation of an enriched IGT example

Group (3): Encodes new information (i.e., information not present in the original IGT) obtained through manual annotation or by running the IGT through NLP systems. If needed, a tier can appear multiple times in an IGT, representing alternative analyses.[3] This group includes:

- *bilingual-alignments*: showing word alignment between the gloss line and the translation line (lines 46–50)
- *dependencies*: showing syntactic dependencies (lines 51–56)
- *phrase-structure*: showing the syntactic phrase structure
- *pos*: providing POS tags for the words in a *words* tier

Group (3) can be extended further if new tier types are needed to present new type of information (e.g., co-reference for the words in the language line as in Ex (2)).

Other Extensions: Besides defining three groups of tiers, we extend Xigt in other ways. Most importantly, we provide a detailed specification about what information should be represented in which tier and how. For instance, the word alignment between the language line and the gloss line is represented in the *alignment* field in the *glosses* tier, whereas the alignment between the gloss line and the translation line is shown in the *bilingual-alignments* tier. We distinguish these two types of alignment because it is more likely that users would want to store alternative analyses for the second type than the first type. In that case, they can simply include multiple *bilingual-alignments* tiers without repeating the segmentation of the gloss or translation line. We also define the conventions for naming tier IDs and item IDs so that those IDs can be generated automatically and systematically. Furthermore, we conventionalize a partial order between tiers so that a tier can only refer to itself or tiers that precedes it. This partial order is crucial when XigtEdit determines how editing in one tier affects other tiers (see Section 5.2).

3.2 Processing Documents with the Xigt API

To make it easier for the researchers to access IGT data in the Xigt format, Xigt provides an application-program interface (API), with a reference implementation in Python, for interacting with Xigt-encoded corpora computationally. The API provides the following functionalities:

- Serialize/deserialize Xigt documents to in-memory data structures
- Iterate over data collections (corpora, IGT, tiers)
- Retrieve object attributes, metadata, and content
- Retrieve the parent (i.e. container) of some object, such as a tier from an item
- Resolve the content, or the targeted items/tiers, of alignment expressions
- Construct new in-memory data structures

[3] It is worth noting that multiple tiers for alternative analyses can be used on the tiers in Groups (1) and (2) as well.

These functions allow users to easily build more complicated functions for their data, such as for counting statistics (e.g., finding the most frequent word), forming complex queries of data (e.g., "what are all the morphemes appearing on words marked as verbs?"), or augmenting a corpus with new analyses (e.g., creating a word-sense tier by looking up each word and its context in an external ontology and aligning the result to the word it came from). The API also enables users to construct new corpora in-memory (e.g., by converting or analyzing some other data) which can then be serialized to disk.

We make use of this API for serializing the ODIN textual data into Xigt and for the subsequent enrichment of the data, as described in Section 4.

4 INTENT: A Package for Creating Enriched IGT

In the previous sections, we described what type of information is in enriched IGT and how it is represented in Xigt. Because manually creating enriched IGT is time consuming and error-prone, we have developed a package, the INterlinear Text ENrichment Toolkit (INTENT), which takes an original IGT file as the input, and produces the enriched IGT in the Xigt format as the output. This output can then be corrected by a human annotator using XigtEdit, or be used to train an NLP system such as a POS tagger or a parser.

4.1 Toolkit Components

Figure 4 shows a typical enrichment workflow in INTENT. The input to INTENT is a file with the original IGT in either plain text format or in Xigt. INTENT first cleans and normalizes the IGT by some simple heuristic rules. It then generates the second group of tiers including *words*, *morphemes* (if the morpheme boundary is present in the IGT), *glosses*, and the like. After that, the third group of tiers are created by running the following modules.

Word Alignment: In our previous study [10], we proposed two methods for aligning the gloss line and the translation line. The first method ran a morphological analyzer on the translation line, and then aligned the words in the two lines if they had the same stems. The second method used GIZA++ [12], a statistical word aligner. Experimental results showed that the performances of the two methods were similar and combining them yielded a small boost. INTENT currently re-implemented those methods, showing F_1 scores of around 0.84 for the heuristic approach and 0.86 for the statistical approach [16]. We are enhancing the heuristic method by taking advantage of the POS tags in the enriched IGT.

Part-of-Speech Tagging: INTENT tags the translation line by running Stanford's English POS tagger [17], trained on the English Penn Treebank [18].[4] As for the language line, while one can simply project the POS tags from the translation line, the quality of the resulting tags is often low due to word alignment

[4] `http://nlp.stanford.edu/software/tagger.shtml`

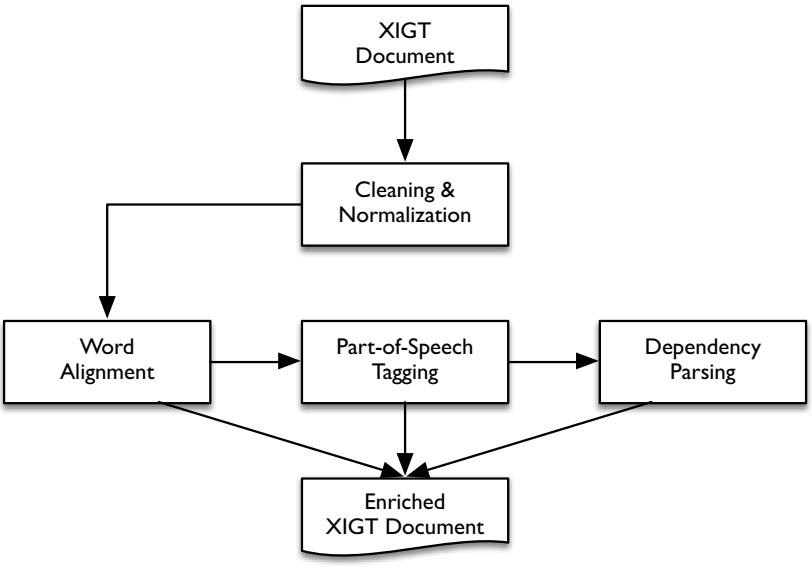


Fig. 4. A typical enrichment workflow in INTENT

errors and translation divergence [19]. Instead, INTENT takes advantage of the annotation on the gloss line; for instance, grammatical markers such as *-Nom* (nominative case marker) and *-Dec* (declarative marker) are good cues for predicting the POS tags of the corresponding words in the language line. We can also find the POS tags of most morphemes in the gloss line using an English dictionary even if those morphemes are not aligned to the words in the translation line. We built a classifier using those features and trained it with a small amount of labeled gloss line data from multiple languages.[5] Evaluation of the classifier yielded a 90% POS tagging accuracy for the gloss line tokens in IGT, compared with 69.5% for the heuristic projection-based approach [16].

Dependency Parsing: The dependency structure for the translation line is produced in three steps. First, the dependency structure for the language line is produced by running the Stanford Parser [20,21].[6] Second, INTENT projects the dependency structure to the language line following the heuristic algorithm in [10]. Third, from a small amount of dependency tree pairs, INTENT learns common divergence patterns automatically and applies them to the dependency structure produced in the second step. Our experiment shows that adding the third step results in an average of 25% error reduction over using the heuristic projection algorithm alone, when tested on eight different languages [5].

[5] We use a classifier, not a sequence labeller, because the word order in the gloss line will be language-dependent, and the training and test data of our POS tagger can come from different languages.

[6] `http://nlp.stanford.edu/software/lex-parser.shtml`

While the workflow in Figure 4 shows a pipeline approach, we are expanding the package to allow feedback loops among the modules. For instance, INTENT runs the word aligner first to get the initial alignment, which will be used by the classifier-based POS tagger. The output of that POS tagger can then be fed back to the word aligner to improve word alignment; for instance, two words unaligned during the first pass of word alignment is more likely to be linked together in the second pass if they have the same POS tags after POS tagging. The improved word alignment can in turn improve the next round of POS tagging.

4.2 Implementation of INTENT

INTENT is written in Python 3 and uses the Xigt API to interface with the serialized documents. INTENT also supplements the Xigt API's internal representations with a number of convenience subclasses for performing tasks such as tokenization and word alignment. Each type of enrichment can be run individually or in sequence.

5 XigtEdit: A GUI Editor for Enriched IGT

As Xigt is an XML-based format, it is nominally human-readable and thus editable with any text editor which is compatible with the desired or appropriate text encodings. However, using existing text editors to edit enriched IGT in the Xigt format is not convenient due to the special properties of Xigt:

- Xigt standoff annotation requires each tier and each tier item to have a unique ID, which is used for cross-reference within an IGT instance. Assigning unique IDs manually is tedious and error-prone.
- Some alignment expressions (e.g., *segmentation* and *alignment* fields in many tiers) require precise computation of string offsets, which are tedious to manually derive.
- Phrase-structure and dependency-structure views are inherently graphical views that do not lend themselves to convenient text-based editing.
- Because tiers can refer to one another, editing one tier could affect the validity of annotation in its cross-referenced tiers. Manually keeping track of the ripple effect of such editing is challenging.

In order to address these issues, we developed a graphical Xigt editor, *XigtEdit*, which facilitates the creation, editing and manipulation of Xigt files.

5.1 Main Functionality of XigtEdit

The fundamental user interface of XigtEdit is a hierarchical structure which closely follows the Xigt abstract data model. Figure 5 shows a screen capture of the XigtEdit application.[7] There are three resizable panels:

[7] To make Figure 5 more readable, certain attributes (e.g., tier IDs, item IDs, alignment between tiers) are not displayed in the screen capture.

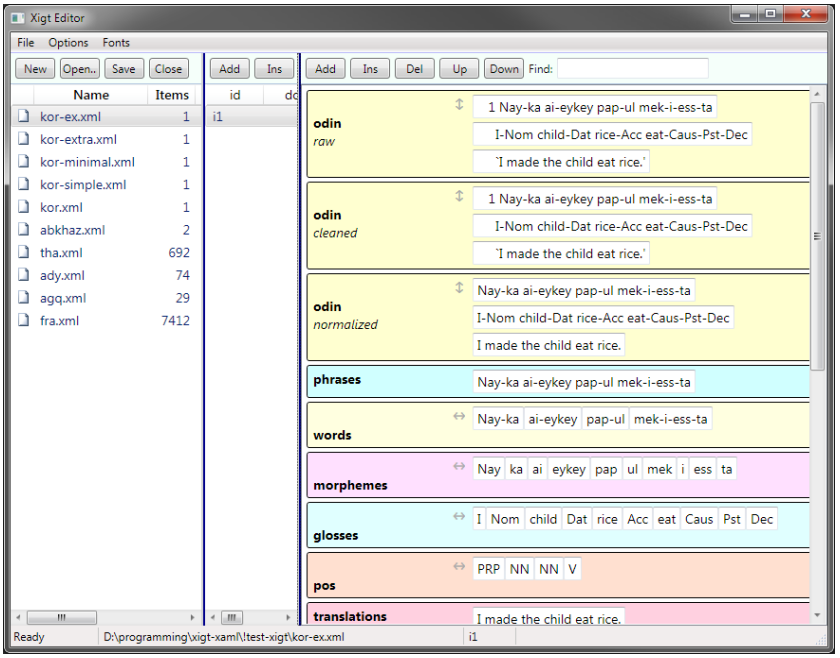


Fig. 5. Main editing interface screen from the XigtEdit application

- The leftmost panel is a list of the Xigt files that have been loaded. If there are more than one file, exactly one file is *currently selected*.
- The next panel, in the second column, lists the IGT instances which are in the selected file. Again, if the file contains multiple IGT instances, exactly one instance is *currently selected*.
- The rightmost panel is the editing area for the selected IGT instance. Tiers are arranged vertically in this area. For some tiers, items in the tiers are arranged vertically (line-oriented data such as in the *odin* tier) while others have items displayed horizontally (word-oriented data such as in the *words*, *morphemes*, and *glosses* tiers).

At any time during editing, individual Xigt files can be opened, edited, saved, closed (added or removed from the files list), or reverted. Files are read and saved directly in the Xigt format. To enhance annotator productivity, XigtEdit assigns unique IDs to new tiers and tier items based on predefined naming conventions (see Section 3.1). To address the inconvenience of computing and maintaining Xigt alignment expressions (e.g., the *segmentation* field in the *words* tier), XigtEdit allows the text spans for dependent items to be defined automatically. This can be achieved either through automatic tools for segmenting text based on whitespace or other criteria, or manually via intuitive user interfaces for manipulating text ranges. Furthermore, XigtEdit displays dependency or phrase structure tiers as graphical trees which the user can edit with mouse clicks.

To support efficient annotation, XigtEdit provides keyboard alternatives to the use of the mouse for most application navigation and editing operations.

5.2 Editing Parent Tiers

As mentioned in Section 3, tiers can refer to other tiers and we define a partial order among tiers such that any tier can only refer to preceding tiers or itself. If a tier C refers to another tier P, we call P a parent of C. A tier can have multiple parents (e.g., a *bilingual-alignments* tier refers to two *words* tiers). Thus, this parent relation among tiers can be represented as a directed acyclic graph, where each node represents a tier, and each link goes from the parent tier to its child tier.

XigtEdit supports the propagation of editing changes from parent to child tiers in an IGT. Consider how editing tier P would affect another tier C that refers to it. The behavior depends on the relation between P and C, and whether other tiers refer to the text region in P that was edited. XigtEdit keeps track of these relationships and analyzes whether the change would invalidate other tiers. In cases where XigtEdit determines that a change has a deterministic effect on its dependent tiers (and where the *edit-propagation* feature has been enabled in the software), the change can be propagated from the parent tier to its child tiers automatically. Alternatively, if the change has ambiguous effects on other tiers, XigtEdit will prompt the user to choose what action should be taken for dependent tiers.

5.3 Implementation of XigtEdit

XigtEdit is a Windows Presentation Foundation (WPF) application which runs on the .NET Application Framework version 4.5.[8] It will run on any modern version of Microsoft Windows (Vista or later) without requiring any additional software or libraries. We chose to develop XigtEdit in WPF primarily because of the developer productivity benefits that WPF offers, such as the ability to implement the software using a declarative markup notation known as Extensible Application Markup Language (XAML). Also compelling in WPF is the "retained mode" graphics subsystem, which, for example, allows persistent data-bindings to be established between entities in the (abstract) Xigt data model and their on-screen representations. These two-way bindings automatically keep the data model and user-interface in-sync without the need for any procedural code. XigtEdit is open source and licensed under the MIT license.

6 Conclusion

The majority of the world's languages lack large-scale annotated resources over which NLP tools such as POS taggers or parsers can be trained. In recent years,

[8] `http://msdn.microsoft.com/en-us/library/aa970268(v=vs.110).aspx`

there have been increasing efforts in bootstrapping NLP systems for resource-poor languages. One line of research uses linguistically annotated data, IGT in particular, to project annotations from resource-rich languages to resource-poor ones.

In this paper, we first provide an overview of enriched IGT and outline several ways that enriched IGT can help linguistic studies and NLP research. Second, we extend Xigt, an XML representation for enriched IGT, and provide an API for it. Third, we introduce INTENT, a package that enriches raw IGT automatically by adding word alignment, POS tags, and syntactic structures to IGT. Finally, we describe XigtEdit, a graphic editor for annotating IGT, which overcomes limitations of existing, general-purpose text editors. By making those tools freely available to the public,[9] we hope that NLP researchers will have easier access to the enriched IGT data and can then focus on exploring new methods for bootstrapping NLP tools for thousands of resource-poor languages by taking advantage of rich annotation in IGT.

As for future work, we are working on improving the performance of INTENT by allowing feedback loops in the workflow. In addition, we plan to create enriched IGT data sets for a dozen resource-poor languages, by first running INTENT on the raw IGTs coming from ODIN and then using XigtEdit for manual correction. The data sets will be released to the public and can be used as training and test data for evaluating NLP systems.

Acknowledgments. The work is supported by the National Science Foundation under Grant No. BCS-1160274 and BCS-0748919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to thank two anonymous reviewers for helpful comments.

References

1. Hana, J., Feldman, A., Amaral, L., Brew, C.: Tagging portuguese with a spanish tagger using cognates. In: Proc. of the Workshop on Cross-language Knowledge Induction, in conjunction with the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006), Trento, Italy (2006)
2. Feldman, A., Hana, J., Brew, C.: A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In: Proc. of the 5th international conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy (2006)
3. Yarowsky, D., Ngai, G.: Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In: Proc. of the 2001 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-2001), pp. 200–207 (2001)
4. Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., Kolak, O.: Bootstrapping Parsers via Syntactic Projection across Parallel Texts. Special Issue of the Journal of Natural Language Engineering on Parallel Texts, 311–325 (2005)

[9] `http://depts.washington.edu/uwcl/packages/`

5. Georgi, R., Xia, F., Lewis, W.D.: Enhanced and portable dependency projection algorithms using interlinear glossed text. In: Proceedings of ACL 2013 (Volume 2: Short Papers), Sofia, Bulgaria, pp. 306–311 (2013)
6. Georgi, R., Xia, F., Lewis, W.D.: Capturing divergence in dependency trees to improve syntactic projection. Language Resources and Evaluation 48, 709–739 (2014)
7. Lewis, W., Xia, F.: Developing odin: A multilingual repository of annotated language data for hundreds of the world's languages. Journal of Literary and Linguistic Computing (LLC) 25, 303–319 (2010)
8. Bailyn, J.F.: Inversion, Dislocation and Optionality in Russian. In: Zybatow, G. (ed.) Current Issues in Formal Slavic Linguistics (2001)
9. Lewis, W.D.: Mining and migrating interlinear glossed text. Technical report, Workshop on Digitizing and Annotating Texts and Field Recordings, LSA Institute (2003), http://emeld.org/workshop/2003/papers03.html
10. Xia, F., Lewis, W.D.: Multilingual structural projection across interlinear text. In: Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007), Rochester, New York, pp. 452–459 (2007)
11. Lefebvre, C.: Creole Genesis and the Acquisition of Grammar: The case of Haitian Creole. Cambridge University Press, Cambridge (1998)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29, 19–51 (2003)
13. Lewis, W.D., Xia, F.: Automatically Identifying Computationally Relevant Typological Features. In: Proc. of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008), Hyderabad, India (2008)
14. Bender, E.M., Goodman, M.W., Crowgey, J., Xia, F.: Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In: Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, Sofia, Bulgaria, pp. 74–83 (2013)
15. Goodman, M.W., Crowgey, J., Xia, F., Bender, E.M.: Xigt: extensible interlinear glossed text for natural language processing. In: Language Resources and Evaluation, pp. 1–31 (2014)
16. Georgi, R., Xia, F., Lewis, W.D.: Training part-of-speech taggers using interlinear text (2015) (manuscript)
17. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL 2003, pp. 252–259 (2003)
18. Marcus, M., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics 19, 313–330 (1993)
19. Dorr, B.J.: Machine translation divergences: a formal description and proposed solution. Computational Linguistics 20, 597–635 (1994)
20. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, ACL 2003 (2003)
21. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proc. of LREC 2006 (2006)

Comparing Neural Lexical Models of a Classic National Corpus and a Web Corpus: The Case for Russian

Andrey Kutuzov[1,2] and Elizaveta Kuzmenko[1]

[1] National Research University Higher School of Economics, Moscow, Russia
[2] Mail.ru Group, Moscow, Russia
akutuzov@hse.ru, eakuzmenko_2@edu.hse.ru

Abstract. In this paper we compare the Russian National Corpus to a larger Russian web corpus composed in 2014; the assumption behind our work is that the National corpus, being limited by the texts it contains and their proportions, presents lexical contexts (and thus meanings) which are different from those found 'in the wild' or in a language in use.

To do such a comparison, we used both corpora as training sets to learn vector word representations and found the nearest neighbors or associates for all top-frequency nominal lexical units. Then the difference between these two neighbor sets for each word was calculated using the Jaccard similarity coefficient. The resulting value is the measure of how much the meaning of a given word is different in the language of web pages from the Russian language in the National corpus. About 15% of words were found to acquire completely new neighbors in the web corpus.

In this paper, the methodology of research is described and implications for Russian National Corpus are proposed. All experimental data are available online.

Keywords: corpora comparison, deep learning, semantic similarity, vector representations of lexical units, lexical co-occurrence networks, Russian National Corpus, Web as corpus, word2vec.

1 Introduction

Contemporary linguistics is in many aspects based on large national corpora carefully crafted by linguists. There are many examples of such 'academic' corpora: British National Corpus, Corpus of Contemporary American English, Turkish National Corpus, Russian National corpus, etc. However, the recent years saw great rise in using text corpora crawled from the Web for linguistic purposes. To some extent they compete with traditional national corpora ([1], [2]).

In this research we compare Russian National Corpus[1] (RNC) to a larger Russian web corpus. Both corpora in some sense represent the Russian language, with the first one being a product of many years of linguistic work on gathering

[1] http://ruscorpora.ru/en

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 47–58, 2015.
DOI: 10.1007/978-3-319-18111-0_4

texts and annotating them and the second one being a random sample of millions of web documents in Russian.

Scholars have already indicated the problem that academic corpora sometimes present researchers with counter-intuitive features, for example, improbable frequency distribution, which puts on top some peripheral scientific lexis [3], see also [4] on the comparison of genre distribution in English and Russian national and Internet corpora. The implications of incorrect representation of the Russian language in RNC were previously discussed in corpus linguistics community, with web corpora proposed as possible solution [5].

The assumption behind our work is that RNC is obviously influenced by the manual choice of constituent texts and genre proportions. Another limitation is its size (only 230 million tokens in the main corpus). That is why the corpus remains biased in various directions and for many words typical contexts (or lexical meanings, which is in fact the same thing) in RNC are different from the ones used in natural living written language. Certainly, the concept of representativeness is complicated; for the purpose of this research we define it as the ability to reflect the associations which the majority of population would have upon meeting a given lexical unit.

We hypothesize that such ability can be found in a vast and full-featured web corpus which serves as an impartial sample of a living language. It means that we should identify words with the meaning in the web corpus essentially (or totally) different from those in RNC. In this research we show that such lexical units can be discovered with the help of neural language models exploiting vectors as distributed representations of words: a paradigm, which has become quite a buzzword in computational linguistics in the last couple of years [6]. It is possible to use these representations to find topical or functional associates of lexical units. We employ them to solve the aforementioned problem. We also describe categories into which these underrepresented words fall. This knowledge is useful when considering the national corpus' architecture and further development.

In Section 2 we explain the architecture and features of the corpora we compare. Section 3 describes the methodology used in our research and provides background for the model of distributional semantics which we employ to identify differing lexical units. In section 4 we perform the comparison of the two corpora with regard to the differing nouns and observe possible causes for the discrepancies found. Section 5 offers some implications for RNC revealed during our research. Finally, in Section 6 we describe limitations of our experiment and future work.

2 Library of Babel vs. Selected Works: The Corpora Used

Opposition of the two corpora under analysis is to some extent similar to that of Selected Works for a writer and the Library of Babel from the famous Borges short story. Russian National Corpus consists of texts which supposedly represent the Russian language as a whole. It has been developed for more than 10

years by a large group of top-ranking linguists, who 'pick' texts and segments for inclusion into the corpus. It was extensively described in the literature[2]. Current composition of RNC is presented on its website[3]. The size of the main part of RNC (without additional sub-corpora) is 230 million word tokens. We worked with the dump containing 174 million tokens. Moreover, to exclude the influence of purely diachronic factors, we restricted ourselves only to texts created after 1950, which amounted to 115 million tokens in total.

Its 'competitor' is a large corpus of texts found on Russian web pages. It originates from a sample of the Russian Internet segment crawled in 2014. This repository contains billions of web documents and actually serves as a source of search index for one of the major search engines in the Russian market, thus is supposed to be quite representative. Consequently, the crawler was sophisticated enough to process even complex dynamic content. Spam and junk pages were also filtered out without our intervention.

To compose the corpus for analysis, we randomly selected about 9 million documents from this repository (no attention was paid to their source or any other properties). Thus, by design the corpus can contain any type of texts found in the Internet (supposedly all major types) in a fully representative proportion. In that, it functions like the Library of Babel, embracing all possible genres and styles in a uniform way.

Boilerplate and templates were filtered out to leave only main textual content of these pages. This was done with the help of *boilerpipe* library [7].

The resulting text archive contained approximately 1.8 billion word tokens. It was split into sentences, and those lacking Cyrillic letters were removed. Thus, we came up with a mainly Russian-language corpus containing about 940 million tokens and 87 million sentences.

Both corpora were lemmatized with the state-of-the-art *MyStem* tool [8]. We used version 3.0 of the software, with disambiguation turned on. It should be noted that some lemmatizer errors became visible later through the output of our language models. For example, for the word *роба* 'boilersuit' the RNC model outputs various types of clothing as topical associates, as expected. However, web corpus model outputs male proper names, obviously, because lemmatizer associated this word and many occurrences of male proper name *Rob* in Genitive, which is homonymous to *роба*. Still, such gross mistakes are rare and do not seriously influence the models in general.

At the stage of lemmatizing, stop-words were removed, as well as single-word sentences (they are useless for constructing context vectors). Then we discovered bigrams which function as integral semantic units using simple data-driven approach proposed in [9] with threshold 1000. These bigrams were joined together with the underscore sign and were treated as single tokens. For example, *углекислый газ* 'carbon dioxide' was transformed into *углекислый_газ*, etc. Such transformation allowed to detect bi-word contextual neighbors which

[2] `http://ruscorpora.ru/corpora-biblio.html`
[3] `http://ruscorpora.ru/en/corpora-stat.html`

otherwise would be split across several word elements (for example, *прямая_наводка* 'direct fire' as an associate for the word *танк* 'tank').

After this preprocessing stage, our RNC corpus was about 70 million tokens in size, and the web corpus shrank to about 620 million tokens. Thus, the web corpus is almost an order of magnitude larger than RNC. It is supported by the size of the corpora lexicons: 751 894 word types for RNC and 5 543 556 word types for the web corpus. So, it seems justified that the latter should at least in some cases provide better contexts for lexical units (and a lot more of lexical units themselves).

3 Learning Word Embeddings and Choosing Test Sets to Compare

Our research is performed within the framework of distributional semantics ([10], [11], [12]) and vector space modeling [13]. In particular, we used *word2vec* neural network language model algorithm [14] to learn vector word representations or neural embeddings.

First we recall the basics of neural embeddings. Lexical meaning is generally the sum of word usages, which is quite traditional for distributional semantics. Thus, the most obvious way to capture meaning is to take into account all contexts the word participates in. In other words, this means to represent each word as a vector of its 'neighborhood' to all other words in the lexicon, with various distances and weighting coefficients (Dice, etc). The matrix of n rows and n columns (where n is the size of lexicon) with 'neighborhood degrees' in the cells is then a distributional model of language. One can compare vectors for different words (for example, calculating their cosine similarity) and find how 'far' they are from each other from the point of view of their contexts.

However, this demands operations on sparse but very large matrices. As we saw in the previous section, our RNC corpus features 750 thousand word types. It means that we would have to compute dot products of 750K-length vectors each time we need to know how similar two words are, which is computationally expensive. Vectors' dimensionality can be reduced to reasonable values using methods like singular value decomposition or principal components, but this often degrades performance or quality. This is where neural embeddings come forward. Neural models are directly trained on large corpora to produce vectors or embeddings of a comparatively small size (usually hundreds of components) which maximize similarity between contextual neighbors found in the data, while minimizing similarity for unseen contexts. The dimensions of the resulting vectors cannot be directly mapped to other words as in traditional distributional models. However, if we use them to calculate which lexical units are similar and which are not, they perform surprisingly well and clearly reveal semantic relations between words (see [15] for further details).

Once the embeddings are learned, we can find the nearest neighbors or quazy-synonyms (most topically related words) for any lexical unit. It is as trivial as to iterate through all the embeddings in the model and rank them according to

their cosine similarity with the embedding for the word analyzed. Words with top-ranking embeddings are the quazy-synonyms we looked for (throughout this paper they are also called associates). It is also possible to compare associates' lists for one and the same word produced by different models trained with different parameters or on different corpora, which is what we proposed above.

Thus, we used RNC and the web corpus to train language models with Python *word2vec* implementation[4]. The models were trained with Continuous Bag-of-Words (CBOW) architecture, vector dimensionality of 500 and context window of 5 words to the left and to the right. Also, for the web corpus we ignored the lexical units occurring only once (3 190 000 word types are known to the model) and for the RNC we ignored the lexical units occurring less than 5 times (205 610 word types are known to the model).

This set of parameters was found to be effective during our participation in Russian Semantic Similarity Evaluation track (RUSSE) [16]. The models trained with such settings performed better than the others trained on the same corpora. Thus, we hypothesize that these models are the best we can derive from corpora under analysis given our tools. Notably, the models trained on RNC outperformed web corpus models in semantic relatedness tasks but not in association tasks (see Table 1).

Table 1. Results of semantic similarity tasks evaluation for different models

	Average precision for relatedness tasks	Average precision for association tasks
RNC model	0.795	0.89
Web corpus model	0.785	0.91

Thus, the RNC-based models are better in singling out exact semantic relations (synonymy, hyponymy and hyperonymy), while the web-based ones excel at discovering associations or topical relatedness. It is also impressive that the RNC is generally on par with the web corpus, despite being an order of magnitude smaller. This is another proof that linguistic balance and well-considered composition of the corpus are of much importance and can sometimes outweigh the pompous 'big data'. See more on this in the forthcoming paper[5].

However, for the purpose of the current research it is sufficient to know that the models we have trained are indeed able to distinguish semantically similar words with state-of-the-art quality. This gives us ground to use them for the comparison of lexical units behavior in the two corpora under analysis.

For the following step, we had to select words to compare. Comparing (and interpreting the results of the comparison) for all lexical units in the models is

[4] `http://radimrehurek.com/gensim/models/word2vec.html`

[5] Kutuzov and Andreev 2015, 'Texts in, meaning out: neural language models in semantic similarity tasks for Russian'

unrealistic. What is more, the data for rare words is sparse, and thus the models are not so reliable for the associates computed. That is why we decided to restrict our experiment to 10 thousand top-frequency nominal lexical units: a quantity which one can realistically look through. Nominal units were chosen because it is usually easier to interpret their semantic relations, and neural models perform better for them, as discovered in the course of the above-mentioned experiments in RUSSE track.

Thus, we selected the top 10 thousand nominal units in RNC, which amounted to approximately $\frac{1}{45}$ of all such units in the lexicon. Then we intersected this set with the similar top $\frac{1}{45}$ set from the web corpus and left only units present in both sets. As a result, we got 9113 nouns (of which 197 are bigrams) frequent in both corpora. The nouns at the end of the lists had absolute frequency of 283 and 232 occurrences in RNC and the web corpus accordingly, which corresponds to the relative frequency of nearly 2 ipm for RNC and 0.15 ipm for the web corpus. The reason for this selection was that we did not want to compare frequent units with lots of data about their usage to rare units with very limited contexts. Also, top-ranking words are more likely to be generally important.

We computed 10 nearest neighbors, or associates, for these nouns, using both models (RNC and the web corpus). Here is an example of the output for the word *динозавр* 'dinosaur' (associates are ranked by their cosine similarity to the query word vector).

RNC:

1. *мамонт* 'mammoth' 0.397899210453
2. *рептилия* 'reptile' 0.360172241926
3. *млекопитающее* 'mammal' 0.328677803278
4. *ящерица* 'lizard' 0.326320767403
5. *птеродактиль* 'pterodactyl' 0.320571988821
6. *черепаха* 'turtle' 0.308944404125
7. *крыса* 'rat' 0.30866342783
8. *птица* 'bird' 0.308208823204
9. *людоед* 'cannibal' 0.303090155125
10. *вымирать* 'to become extinct' 0.295859247446

Web Corpus:

1. *рептилия* 'reptile' 0.496797531843
2. *мамонт* 'mammoth' 0.443771362305
3. *млекопитающее* 'mammal' 0.424831837416
4. *хищный* 'carnivore' 0.412433445454
5. *ящер* 'pangolin' 0.401978999376
6. *крокодил* 'crocodile' 0.396325200796
7. *ящерица* 'lizard' 0.393893510103
8. *черепаха* 'turtle' 0.393123477697
9. *доисторический* 'prehistoric' 0.391041249037
10. *гигантский* 'giant' 0.386854737997

It is obvious that we have two partially intersecting sets of quazy-synonyms or topical associations. The degree of difference between them can be trivially calculated using the Jaccard similarity coefficient (the size of the intersection divided by the size of the union of two sets) [17]. It takes values in the interval [0,1] and serves here as a measure of how much the meaning of a given word is different in the National corpus from the 'unsupervised' web corpus. If the Jaccard coefficient equals to 0, that means that the two sets of neighbors do not intersect and thus the meaning is supposed to be totally different. If, on the contrary, the coefficient takes the value 1, the two sets are identical: both models provided precisely the same set of 10 nearest semantic neighbors. With our data this happened only once with the word *август* 'August', for which both models output names of other months. As for the example above ('dinosaur'), its Jaccard similarity is $\frac{1}{3}$ (5 identical associates of 15 total).

Note also that this way we cannot discover lexical units which RNC lacks altogether or for which it does not provide enough frequency (such a problem can be solved without neural embeddings, by simple comparison of lexicons). Instead, we find discrepancies in the words which are present in the corpus, and even reach top positions in the frequency list.

Our trained models are available online, together with the Jaccard coefficients and example scripts[6].

4 What's Different: Analysis of Lexical Units' Neighbors in RNC and the Internet Corpus

We have compared associates for 9113 nouns and noun phrases, considering the Jaccard coefficient between corresponding sets for RNC and the web corpus. 1467 lexical units (about 15%) show the coefficient equal to 0, which means they do not have any neighbors in common. Almost the same number of lexical units (1463) show the Jaccard coefficient higher than 0.3, which means that at least half of their 10 semantic neighbors are the same in RNC and in the web corpus.

About 20 words are as close to maximum Jaccard value as 0.81 (only one differing neighbor). These are mostly months and female patronymics ('*степановна*', '*николаевна*', etc). Strangely, the remaining words with such a high value of Jaccard coefficient all belong to a rather threatening cluster: *бандит* 'bandit', *кинжал* 'dagger', *граната* 'grenade', *конфликт* 'conflict', *пытка* 'torture', *опасение* 'fear'. Supposedly, criminal and military topical segments in RNC are quite consistent with those found 'in the wild'.

In general, the two corpora do agree with each other in most cases. Indeed, considering unsupervised nature of our models, 3 coinciding associates out of 10 is already enough to suppose that both corpora share similar meaning. If so, more than 50% of all units under analysis should be considered as 'agreeing'.

We studied 1467 lexical units with Jaccard coefficient equal to 0, because they are the most critical cases of discrepancy: no coinciding associates at all.

[6] `http://www.cicling.org/2015/data/107`

Table 2. Thematic classes of most differing words

	commerce and finance	politics and law	terminology	high register	recent concepts	Soviet era concepts
Absolute value	97	65	59	122	53	24
Percentage	6.6%	4.4%	4.0%	8.3%	3.6%	1.6%

The aim of our analysis was to reveal possible patterns according to which nouns can fall in this category. The results are demonstrated in the table 2.

The first category which we found within the problematic cases was nouns describing *economics concepts* such as trade, finances, natural resources. There is little wonder that nouns from these categories differ significantly in the two corpora because economics changes rapidly, and, therefore, its concepts in a language also develop quite fast. One of the examples of 'incorrect' neighbor sets in RNC can be the word *брокер* 'broker'. Its neighbors in the web corpus include *биржа* 'exchange market', *дилинг* 'dealing', and *трейдер* 'trader', while in RNC we can find only general terms: *фирма* 'firm', *компания* 'company', *менеджер* 'manager'. Another example is *вакансия* 'vacancy, opening': its neighbors in RNC are *безработный* 'unemployed', *приработок* 'side job', *должность* 'job position', etc, while in the web corpus the word is associated with *резюме* 'resume', *соискатель* 'applicant', *трудоустройство* 'recruitment', and titles of various Russian recruiting web sites, like *job.ru*. It can be seen that in general the web corpus describes these concepts more precisely, although it can contain some unappealing language data, like titles of web sites.

The same cause of discrepancy between the corpora can also be found in the second category, where one finds nouns that refer to *political and social phenomena*. For example, the word *бюллетень* 'bulletin' in RNC produces the following associates: *газета* 'newspaper', *брошюра* 'brochure', *сводка* 'report', *некролог* 'obituary'. In the web corpus, on the contrary, we find a different list: *избирательный бюллетень* 'voting paper', *избиратель* 'voter', *открепительное удостоверение* 'absentee ballot', etc. The senses of this word extracted from both corpora demonstrate radically different shades of meaning: whereas in the web corpus the meaning of the word *бюллетень* is more politically biased, in the RNC it is more official and academic.

In the *terminology* category we find words which are associated with specific professional domains, such as chemistry, physics, or mathematics. These lexical units also differ significantly in the two corpora under analysis. In the web corpus associates are more 'terminological' and more precise than those in RNC. For example, the word *анализатор* 'analyzer' in RNC is associated with *передатчик* 'transmitter', *механизм* 'mechanism', *контроллер* 'controller', while in the web corpus the associates were *спектрометр* 'spectrometer', *глюкоза* 'glucose', *lactate* 'лактат'. Probably, this reflects more specific lexical functions. Similarly, *бензол* 'benzol' in RNC is associated with a few chemical substances like *метанол* 'methanol', but mostly with lexical units like *вата* 'cotton pellet' or

растворитель 'organic solvent'. In the web corpus this unit is associated with chemical substances only (e.g., *серная кислота* 'sulfuric acid' and *аммиак* 'ammonia'). It seems that in the web corpus words are employed in more particular contexts, whereas in the RNC their usage is general.

'Literary' words belonging to the *high register*, which are not normally used in speech, were found to constitute 8.3% of all differing lexical units (122 instances. We encountered some interesting examples with regard to these units. For instance, the word *кроха* 'little one, little piece' in the web corpus was associated with *малышка* 'little girl', *младенец* 'baby', *карапуз* 'little child', etc. and clearly denoted 'a kid, a baby', whereas in RNC there were such neighbors as *кусочек* 'little piece', *лихва* 'more than needed', *грош* 'farthing', *посылочка* 'delivery' and the word clearly denoted 'a crumb, a little piece of some object' and not a person. The first set of neighbors and implied meaning indeed seems to be more intuitively correct. The word *приют* 'asylum/orphanage' can serve as another discrepancy observed for words in this category. In RNC, the words like *прибежище* 'refuge', *пристанище* 'haven', *утешение* 'consolation' can be found among its associates, which implies that its meaning is closer to 'asylum'. In the web corpus, however, the associates were *бездомный* 'homeless', *сирота* 'orphan', *детдом* 'orphanage', which points at the second meaning, 'orphanage'. Other words which share similar pattern include *палата* 'chamber/ward', *химера* 'dream/chimera', etc.

However, there are other words in this category which lack the dichotomy of meaning, but are archaic or simply bookish. For these words sometimes neighbors identified in the RNC are more reasonable than those in the web corpus. One example is the word *потеха* 'merry-making', associated in RNC with quite relevant units like *гулянка* 'party' or *перекур* 'smoke-break, while the web corpus outputs less relevant neighbors (except *забава* 'jolly' and to some extent *петросятина*, derogatory derivative of the name of a famous Russian stand-up gag-man Petrosyan).

There was also a minor number of nouns that refer to Soviet era concepts, such as *комсомол* 'Komsomol', *комиссариат* 'comissariat', *партком* 'party committee', etc. With these concepts, more incorrect neighbor sets can be found in the web corpus, while in RNC the associates are better, as it features relatively higher number of texts originating from that time.

Another kind of situation can be observed with respect to nouns describing *contemporary concepts*, like IT and the Internet. RNC, supposedly, does not contain a sufficient amount of texts created in the last ten or fifteen years, when we saw a tremendous growth of Internet usage, and this is the reason why the word embeddings in the web corpus are more accurate and therefore more representative of lexical meaning. For instance, for the word *гиперссылка* 'hyperlink' there are no meaningful associates in RNC (only unclear or unrelated words), and in the web corpus we can find associates like *ссылка* 'link' and *индексировать* 'to index' (and also, interestingly, the same word misspelled: *гипперссылка*).

In spite of all the cases of discrepancy which serve in favor of the web corpus, there are also some negative examples. For instance, the word *нота* 'note/pitch' has the following associates in the web corpus: *мускус* 'musk', *амбра* 'ambergris', *пачули* 'patchouli', etc. Of course, the meaning related to music seems to be more prototypical than the one originating from perfume industry. This is probably a sign of the web corpus bias towards commercials.

Another negative example is the word *бич* 'whip', whose associates in the web corpus are *Таиланд* 'Thailand', *пляж* 'beach', *курорт* 'resort'. The possible reason is the Russian transliteration of English *beach*, frequently used in the web and homonymous to *бич*.

We note that the cases of difference which were analyzed constitute only 28.5% of all the nouns demonstrating the Jaccard coefficient equal to 0. Our analysis is limited to pointing out the most distinctive cases of discrepancy, and it seems impossible to assign an exact category to every noun, either because it falls outside all classifications or because of the fact that, due to the statistical nature of the models, some words possess a set of neighbors which are totally unrelated or are outright junk. This happens because of *MyStem* or *boilerpipe* errors, occasional spam or duplicate texts and other noise factors. However, even the categories revealed above can help to decide with which texts it would be better to augment RNC with.

5 Discussion

It should be emphasized that Russian National Corpus still remains the most sustainable and authoritative Russian language corpus. Its composition strategy was a success, and it indeed provides a balanced sample of the national language. This is again proved by the excellent performance of neural semantic models trained on its texts.

Having said that, we have shown that some lexical units in the corpus are surrounded by contexts that make it difficult to grasp the meaning of the word as it is used in the living written language. It means that some bias exists in the corpus data, making it less representative.

Manual discovery of such cases is extremely difficult (if possible at all). At the same time, comparing neural language models trained on the RNC to those trained on other corpora allows to perform this task in an unsupervised way, extracting necessary data automatically from lexical co-occurrences. Web corpora are good candidates for such comparisons, as in constructing them we at least partially avoid human selection bias: in our case documents are drawn randomly from the 'Babel library' of the Internet (from its representative model crawled by a search engine, to be exact). Such corpora are a useful resource to enhance 'academic' RNC corpus and make it more up-to-date through applying linguistic data from the web to augment the corpus with the texts of particular categories.

Even based on our initial research one can conclude that the RNC maintainers should pay more attention to texts related to economics, politics, law and the Internet (and possibly some other rapidly changing spheres of our life). At the

same time, detailed analysis in order to determine precise areas of expanding is certainly needed. Also, sometimes it is difficult to decide which set of associates (which meaning) is 'correct': this is the case with many 'high register' words. Should *приют* be more of a refuge for a tired soul or should it be an orphanage for children who lost their parents and pets who lost their hosts? If one of these meanings is a bit archaic, does it mean that the National corpus should reduce its presence?

Such questions are not easy to answer, but corpus linguists should at least possess the tools to measure the degree of disagreement between National corpora and other linguistic resources. One of such tools is presented here.

6 Limitations and Future Work

Main limitations of our experiment concern the composition of the web corpus. First of all, no duplicates were removed, thus, multiple copies of texts are undoubtedly present in the corpus. According to some estimates, about 40% of all web pages on the Internet are duplicates [18], which means that this can become a harsh problem and critically dis-balance the corpus. Thus, we are going to experiment with de-duplicating our web corpus using shingles or other established approaches, and see whether this would change the results.

Another issue is the language of web pages in the corpus. We did not apply proper language detection and considered all sentences with Cyrillic characters to be 'Russian', which led to some noise. For example, the word *свая* 'pile, pole' is characterized in the web corpus by a set of 'neighbors', all of which seem to be Belorussian ('*цяпер будза яго ён калі пра толькі гэт якать ў*'). It means that there is a sufficient amount of Belorussian texts to pop a Belorussian reflexive pronoun up above the homonymous Russian noun.

Therefore, we plan to apply a simple n-gram based language detector to the corpus and to select only Russian texts in order to avoid multilingual noise.

Another kind of noise is created by a widespread use of English lexicon in the Web. This problem is more difficult to solve as English (and non-Cyrillic in general) words are not always inappropriate in the corpus.

Finally, it would be interesting to research into the behavior of other lexical classes (especially verbs) and multi-word entities more than 2 words length. This should induce more insights about the essence of lexical differences between the RNC and the web corpus.

Acknowledgments. The authors cordially thank Igor Andreev of Mail.ru Search applied linguistics team for his inspiring idea. Support from the Basic Research Program of the National Research University Higher School of Economics is gratefully acknowledged.

References

1. Kilgarriff, A., Grefenstette, G.: Introduction to the special issue on the web as corpus. Computational Linguistics 29(3), 333–347 (2003)
2. Baroni, M., Ueyama, M.: Building general-and special-purpose corpora by web crawling. In: Proceedings of the 13th NIJL International Symposium, Language Corpora: Their Compilation and Application, pp. 31–40 (2006)
3. Belikov, V.: What are sociolinguists and lexicographers lacking in a digitized world? (in Russian). In: Proceedings of the Dialog Conference (2011)
4. Sharoff, S.: In the garden and in the jungle: Comparing genres in the bnc and the internet. In: Genres on the Web, pp. 149–166. Springer (2011)
5. Belikov, V., Kopylov, N., Piperski, A., Selegey, V., Sharoff, S.: Corpus as language: from scalability to register variation (in Russian). In: Proceeding of the Dialog Conference (2013)
6. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1 (2014)
7. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 441–450. ACM (2010)
8. Segalovich, I.: A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In: MLMTA, Citeseer, pp. 273–280 (2003)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
10. Curran, J.R.: From distributional to semantic similarity. PhD thesis, University of Edinburgh (2004)
11. Lenci, A.: Distributional semantics in linguistic and cognitive research. Italian Journal of Linguistics 20(1), 1–31 (2008)
12. Bruni, E., Tran, G.B., Baroni, M.: Distributional semantics from text and images. In: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, pp. 22–32 (2011)
13. Turney, P.D., Pantel, P., et al.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37(1), 141–188 (2010)
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
15. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2 (2014)
16. Panchenko, A., Loukachevitch, N.V., Ustalov, D., Paperno, D., Meyer, C.M., Konstantinova, N.: Russe: The first workshop on russian semantic similarity. In: Proceeding of the Dialogue 2015 Conference (2015)
17. Jaccard, P.: Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines. Rouge (1901)
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge university press, Cambridge (2008)

Lexical Network Enrichment Using Association Rules Model

Souheyl Mallat[1], Emna Hkiri[1], Mohsen Maraoui[2], and Mounir Zrigui[1]

[1] LATICE Laboratory Research Department of Computer Science, University of Monastir, Tunisia
[2] Computational Mathematics Laboratory, University of Monastir
{Souheyl.mallat,Emna.hkiri,maraoui.mohsen}@gmail.com, mounir.zrigui@fsm.rnu.tn

Abstract. In this paper, we present our method of lexical enrichment applied on a semantic network in the context of query disambiguation. This network represents the list of relevant sentences in French (noted by $list_{RSF}$) that respond to a given Arabic query. In a first step we generate the semantic network covering the content of the $list_{RSF}$. The generation of the network is based on our approach of semantic and conceptual indexing. In a second step, we apply a contextual enrichment on this network using association rules model. The evaluation of our method shows the impact of this model on the semantic network enrichment. As a result, this enrichment increases the F-measure from 71% to 81% in terms of the ($liste_{RSF}$) coverage.

Keywords: Association rules model, semantic network, contextual enrichment.

1 Introduction

In the last decade lexical disambiguation under automatic query translation around the world has taken giant leaps. Our disambiguation method leverages on our method of listRSF representation. The list includes sentences that semantically answers the Arabic query noted list of relevant sentences in French (listRSF). Concerning the construction of the $list_{RSF}$ corresponding to the list of relevant sentences in Arabic $list_{RSA}$ [1][2], it was obtained by a aligning step at the sentence level with MkAlign tool [3].The disambiguation method will attempt to improve our system of Arabic queries translation by eliminating the translated terms, with other meanings/senses that do not belong to the semantic context of $list_{RSF}$. In this work, we are interested in representing the $list_{RSF}$ by a semantic network. A number of work under the automatic processing of natural languages NLP are based on the principles presented in [4] to exploit networks of lexical collocations (semantic, syntactic, pragmatic). [5] used the lexical networks in the context of word sense disambiguation. In addition [6] exploited it respectively in the parsing and the generation. Such networks have the advantage of being easy to build automatically. Consequently in our context, we do treat our $list_{RSF}$ without limitation to a particular theme.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 59–72, 2015.
DOI: 10.1007/978-3-319-18111-0_5

In this paper, we first propose a structure of semantic network, in order to realize the semantic cohesion through (1) the various relations (synonymy, hypernymy, hyponymy, meronymy), (2) grouping concepts relative to significant terms of the $list_{RSF}$ and (3) projecting them on the French EuroWordNet (EWNF) [7]. The problem is that the network generated in this step does not fully cover the $list_{RSF}$ content. In fact, it partially fulfills our objectives because of its limits to some queries containing ambiguous words.

To overcome this issue we pass to the following next step. In this step, we do use contextual information between concepts.

This allows emerging concepts and contextual relations defined implicitly in order to obtain a rich semantic description of the listRSF. These relations are provided by the semantic associations' rules which are generated by Apriori algorithm[8]. As a result of the previous step our semantic network is contextually enriched. Our development is performed on French data extracted from the "diplomatic Monde" corpus [9]. These data are presented in two formalisms; the first is semantic network without contextual enrichment and the second is contextually enriched. These two networks will be used in a comparative study in order to demonstrate the effect of this enrichment on the coverage of the $list_{RSF}$. The works presented above reflects the importance of this domain and shows some diversity in approaches to acquire relations between terms. In this paper, we propose first a method to represent the listRSF by a semantic network similar to the work of [8]. Our network is essentially composed of concepts associated to significant terms identified from the listRSF. So, we propose in the first step a method of representation of the listRSF by a semantic network using our indexing method. After the indexing we build the network (identification of the nodes and the relations between them). In the second step, we are interested in enriching this semantic network by adding other hidden relations.

1.1 State of Arts: Approaches of Semantic Relations Identification

In this section, we first present the different semantic relations that can exist between terms and two methods of acquisition of these relations.

Use of Contextual Distribution of Terms in Relations Extraction

It consists in grouping terms sharing context (in origin syntactic)[10]. For example the term teacher and the board are semantically close because they share the same context which is teaching. Distributional analysis method applied on a corpus of texts allows to identify several type of relations; proximity relations [11], synonymy relations [12]. This method was also used by [13] to highlight the semantic relations associated with terms. The idea was to replace the present terms in the contexts by their semantic classes, based on WordNet. For example, the terms solder is replaced by the class "ministry of defense" in WordNet. There is also a hybrid approach that combines the distributional analysis and lexico- syntactic patterns methods presented in [14].

The works presented above reflects the importance of this domain and shows some diversity in approaches to acquire relations between terms. In this paper, we propose first a method to represent the listRSF by a semantic network similar to the work of

[8]. Our network is essentially composed of concepts associated to significant terms identified from the listRSF. So, we propose in the first step a method of representation of the listRSF by a semantic network using our indexing method. After the indexing we build the network (identification of the nodes and the relations between them). In the second step, we are interested in enriching this semantic network by adding other hidden relations

2 Representation of the listRSF by Semantic Network

This section is devoted to introduce the formalism of network. Our network composed essentially by the set of concepts associated to significant terms, those are identified from $list_{RSF}$. This identification aims to extract significant information of the $list_{RSF}$ and is essentially based on the indexing process.

2.1 Description of the Indexing Method of listRSF

To create our indexing method, we are inspired by Baziz work [15], in order to represent the $list_{RSF}$ by a list of index concepts. It is based on the combination of semantic and conceptual indexing [16]. In the semantic indexing, the used semantic structure makes possible the extension of the representation of the $list_{RSF}$ by the relation of synonymy. Baziz proved that this method improves the quality of the system contrary to an indexing based only on conceptual indexing. He demonstrates that his IR system performs better with this combination, since it had produced less than 30% of disambiguation errors. Our indexing method is based on the use of the semantic network French EuroWordNet (EWNF). The method of indexing incorporates three main steps:

Extraction of concepts from significant terms (simple and composed) of the $list_{RSF}$ is done by projection on EWNF. If the projection generates for a given term several corresponding concepts, then this term will be disambiguated. The identification of composed terms in the list is interesting to improve the performance of the automatic indexing. The use of composed terms reduces considerably the ambiguity of terms and increases precision (reduces the number of senses of a term). For example the composed term "North America" takes one sense, with 6sense for term north, and 3 for America returned by EWNF. Our method for identifying simple and composed terms is based on a symbolic method, it requires a morphosyntactic analysis of $list_{RSF}$. We use analysis obtained by integrating TreeTagger Helmut[17]. The analysis provided by TreeTagger, can produce a list of words labeled by their grammatical categories. Most of composed terms consist of combinations of nouns, adjectives and prepositions, we generate a list of n-grams ($2 \leq n \leq 3$).

Concepts Weighting: Once the simple and composed terms are extracted from the $list_{RSF.}$ We assign to each one of them a weight in the $list_{RSF}$. The purpose of this step is to eliminate the least frequent terms and maintain only the most representative terms in the $list_{RSF}$. The weighting method, which combines statistical and semantic analysis [18], for assigning weight to the terms of $list_{RSF}$ optimally in terms of frequency of each with their semantic variation.

For the statistical analysis: in the step of concepts identification we are interested in the importance of composed terms but in some cases, the words composing these terms can refer to them even when used alone, after a number of occurrences. This represents a form of simplification or abbreviation used by the author. Let Ti be a term, its frequency depends on the number of occurrences of the term itself, and the words that compose (or sub-term (STi)). Statistical analysis is defined by the conceptual frequency of a term Ti for the $list_{RSF}$, it is calculated as follows:

$CF\ (T_i) = count(Ti) + \sum_{ST \in Ti} \left(\frac{Length(STi)}{Length(Ti)} . count(STi) \right)$(1); With Length (STi) represents the number of words in Ti and STi, represents the sub-terms (single words) derivatives of Ti.

The semantic analysis is based on the representativeness of a concept, which takes into account the frequency of occurrence of terms, denoting the concept in the $list_{RSF}$ but also its relations with other concepts in the domain. The more relations with other concepts present in the $list_{RSF}$ a concept has, the more is this concept a representative of the $list_{RSF}$ .The EWNF resource is used to generate the set of concepts related to these terms in the form of synset taking every defined sense, and its semantic relations. The basic relation between the terms of the same synset is synonymy, but different synsets are otherwise related by various semantic relations such as subsumption, or hyponymy / hypernymy. In our case, we used the weighting method of semantic frequency of the term W_frqsem (Ti), which is calculated for each term in function of: the frequency of occurrence of the concepts associated to that term, and the ranks of sentences to which those concepts do belong. The coefficients corresponding to each sentence are assigned as follows: if a term belongs to the first sentence its coefficient is 10, 9 for second, and 1 for the tenth and the rest of the sentences in the $list_{RSF}$. Assuming that term Ti containing n terms and appears p times in the $list_{RSF}$, Mi,j is the coefficient for sentences containing the conceptual occurrence j of the term Ti (different senses associated with this term, extracted from a EWNF, and for each sense of this term, a synset is associated, as well as all semantic relations). The weight of semantic frequency W_freqsem of a term Ti in the $list_{RSF}$ is calculated as follows:

$W_{freqsem(Ti)} = \frac{P(Ti)}{maxp=1..n(P(Tp))*ns}$ Where $P(Ti) = \sum_{j=1}^{K} (Mi,j)$ is the weight of term Ti, and Ns=k - number (Mi, j=0) with (ns presents the number of possible senses of Ti).W (Ti, $list_{RSF}$) represents the global weight of a term Ti in the KB ($list_{RSF}$), is defined by the expression: W (Ti, $list_{RSF}$)=WTi=$CF(Ti) * W\_freqsem(Ti)$ (3) The index of $list_{RSF}$ noted Index ($list_{RSF}$) = (Ti, WTi).

Disambiguation of index terms aims to identify the exact sense of a polysemous index term in the $list_{RSF}$. For an ambiguous term Ti belonging to the index $list_{RSF}$. Let Si, the number of senses associated with the term Ti. The principle of the disambiguation method is to select the best concept (sense) in the $list_{RSF}$ from several (C1, C2, Cn). In the semantic disambiguation, we are interested in the method used by [19]. It is based on the calculation of a symmetric similarity weight (P (c)) for each concept associated with term Ti of sense j of the list of indexes: the formula is as follows:

$P\ (C_i^j) = \sum_{l\epsilon[1..m]l\neq i} \sum_{k\epsilon[1..nl]} Dist\left(C_i^j, C_l^k\right)$ (4) with m and nl represent the number of terms in Index ($list_{RSF}$), and the number of senses of the term Ti in EWNF, Dist(C_i^j, C_l^k) is a measure of proximity between semantic concepts C_i^j and C_l^k [20] [8], it is calculated by a score based on their mutual distance in the network EWNF. The disadvantage of this method is that it considers only the semantic similarity between concepts in $list_{RSF}$, but it does not take into account the representativeness of terms in the context of $list_{RSF}$. So the best sense for a term ti in $list_{RSF}$ must be strongly correlated to the senses associated with other important terms in $list_{RSF}$. For this reason, we will integrate the weight of the term in the calculation of conceptual scores, using the following formula:

$$P(C_i^j)=\sum_{\mathbf{l\epsilon[1..m]l\neq i}} \sum_{\mathbf{k\epsilon[1..nl]}} (WC_i^j, \text{listRSF} * WC_l^k, \text{listRSF} * Dist(C_i^j, C_l^k). \quad (5)$$

The concept with the highest weight is considered the best sense of the term Ti. After extracting the concepts and calculation of their weights, the $list_{RSF}$ will be represented by m concepts (m <= n) with their respective weights called $list_{RSF}$ of indexed concepts. This list forms the semantic core of the network, designated by Nsem($list_{RSF}$).

2.2 Construction of Semantic Network (structuring and limit)

The semantic network consists essentially of the set of semantic concepts from the core $list_{RSF}$ Nsem ($list_{RSF}$) interconnected. The network is structured as (C , domain (C)) by exploiting the lexical database EWNF as C represents the concept (node), and the "domain (C)" all synset Si of Nsem($list_{RSF}$) with C subsumes Si. In EWNF, an entry is a concept that is represented by a synset, that is to say, all terms (words or set of words) synonyms that can describe this concept. The concepts are defined as a set of lexical units related to specific domains.

Let G ($list_{RSF}$) = {(C, domain (C))} represents the nodes of the semantic network $list_{RSF}$ in what follows; we describe the components of the semantic network: nodes (concepts) and semantic arcs.

Identification of Concept-Nodes

The nodes represent concepts semantically related to different concepts (C1, C2, C3 ... C_k) of Nsem ($list_{RSF}$) identified in the previous steps. The basic phases to create the network nodes associated with the $list_{RSF}$ are:

- Phase 1: designation for each variable (instance) of Nsem ($list_{RSF}$) by a corresponding concept from EWNF; Each concept C_i corresponds to values C_i^k in domain (Ci) = {C_i^1, …}

- Phase 2: each concept in Domain (Ci) is-a concept $C_i^j \epsilon$ Nsem ($list_{RSF}$) as C_i^k is -a Ci. the previous two phases, allowed us to build the set of nodes in the $list_{RSF}$ for the semantic network: Nœuds($list_{RSF}$)={(c1,domain(c1)),(c2,domain(c2)), …(Cn, domain(Cn))}. The following example presents the Ci nodes, as well as domains domain (Ci) associated with the theme "Military" by the application of the previous two phases: Consider the following example from our corpus, which illustrate an indexed $list_{RSF}$ by the following weighted concepts.

Table 1. Weighted Concepts

Concept	W	Concept	W
organisation de défense (engl. defence organisation)	0.55	encadrement (engl. supervision)	0.2
établissement de défense (engl. defense constitution)	0.5	acquérir (engl. acquire)	0.5
Action commando (engl. commando action)	0.35	obtenir (engl. get)	0.1
Effort (engl. effort)	0.3	opération aérienne (engl. air operation)	0.6
véhicule militaire (engl. Military vehicle)	0.4	force armée (engl. armed force)	0.8
véhicule de combattants (engl. vehicle of fighters)	0.25	soldats (engl. solders)	0.7
Entités (engl. Entities)	0.12	combattants (engl. fighters)	0.6
victime(engl. victim)	0.25	région montagneuse (engl. mountainous region)	0.5
blessé (engl. injured)	0.25	Forêt (engl. : forest)	0.15
Panzer (engl. Panzer)	0.1	ingérence (engl. interference)	0.7
Pistolet (engl. pistol)	0.12	négociation (engl. negotiation)	0.25
Tourelle (engl. turret)	0.08	imposer (engl. impose)	0.7
balle (engl. ball)	0.09	Demande (engl. demand)	0.1
indépendance (engl. independence)	0.6	massif de soldat (engl. solders)	0.7
triomphe (engl. triumph)	0.12	nombreux (engl. many)	0.2
Réussite (engl. success)	0.2	sécurité de pays (engl. country security)	0.4
sécurité de peuple (engl. people security)	0.4	sécurité de frontière (engl. border security)	0.5

W is the weight of each term, it gives the importance of the term (occurrence and semantic) in the $list_{RSF}$ such as *W(organisation de défense)=0.55, W(véhicule militaire)=0.4* etc. *These concepts constitute the semantic core of the* $list_{RSF}$ associated to the topic "Militaire". Using subsumption relations (is-a) between concepts and properties (relations domain) through EWNF as concept *organisation de défense, Ministère de défense, établissement de défense* are associated to the concept *Ministère de défense*. Etc. We get the following concepts representing the $list_{RSF}$ and composing the network nodes: node ($list_{RSF}$).

Table 2. Identification of concept nodes

Concept node	Domain(concept)
organisation de défense	{Ministère de défense, établissement de défense}
action militaire	{Action commando, Effort}
transporteur militaire	{véhicule militaire, véhicule de combattants, entités}
guerre	{victime, blesse}
Moyen d'attaque	{panzer, pistolet}
munitions	{tourelle, balle}
autonomie	{indépendance}
victoire	{triomphe, réussite},
occupation	{acquérir, obtenir},
opération	{opération aérienne, encadrement}
militaire	{force armée, soldats, combattants},
nature	{région montagneuse, forêt},
intervention	{ingérence, négociation},
sécurité	{sécurité de pays, sécurité de frontière, sécurité de peuple}
nombre	{massif de soldats, nombreux}
ordre	{demande, imposer }

Identification of Semantic Relations between Nodes (Concepts)

Several semantics relations are proposed by the EWNF resource, such as generic-specific relations or hypernym-hyponym (is-a), and the relations of composition holonymy-meronymy (part-whole). Finally we retain semantic relations between nodes concepts in the previous example. Figure Fig.1 below shows the following semantic network that illustrates the concepts with these relations in the "Military" theme:

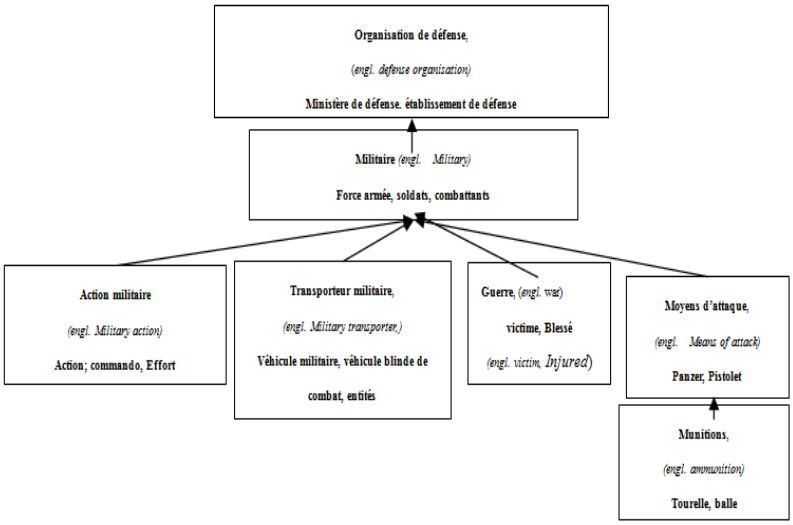


Fig. 1. A semantic network corresponding to the theme 'Military' in EWNF

This network includes only the close relations between the semantic concept nodes. However, we observed the absence of relations with other relevant concepts that are close in the same context (*victory, military operation, occupation, intervention, etc*.). Indeed, the coverage of EWNF is small compared to the list of index concepts (Nsem ($list_{RSF}$)). Using EWNF mentions the lack of useful contextual relations between relevant concepts. This lexical database contains only limited information on the use of concepts. So the network is obviously insufficient for lexical disambiguation of all existing ambiguous words in the queries. Hence, we need to increase the coverage of this network by contextual enrichment. We pass now to the second step of the representation of the semantic network that consists in the enrichment by the contextual relations.

3 Contextual Enrichment of the Semantic Network (Structuring and Utility)

Our objective is to make out existing and hidden contextual relations between nodes (concepts) representing Nsem ($list_{RSF}$). We used a method based on semantic association rules. These last are extracted by the Apriori algorithm, for more details see [21]. The principle of association rules discovery can be presented as follows: Let I = i_1, i_2, i_n a set of items and D a set of transactions where each transaction T is a set of items such that $T \subset I$.

A set of items is called an itemset. An association rule is an implication of the form $X \rightarrow Y$, where $X, Y \subset I$ and $X \cap Y = 0$. Generally, X is called the antecedent and Y the consequent.

We do apply the semantic association rules model in order to identify the contextual relations between nodes (concepts). In this step, we use the Apriori algorithm to extract relations (arcs). This algorithm has two steps; the first is to extract all frequent itemsets of the $list_{RSF}$. The second step is the generation of the association rules between frequent itemsets discovered during the first step. They are detailed as follows:

The generation of frequent itemsets includes three main phases :

construction of the group E1 1-itemsets which is the most frequent concepts in the $list_{RSF}$, which have a weight P1-itemsets greater than a given threshold;

From the E1 of 1-itemsets frequent calculated in the previous step, we generate the set 2-itemsets of candidate in order to construct E2, which have a weight greater than a given threshold $P_{2\text{-itemset}}$ With $P_{2\text{-items}} = \min_{items} = \min (P_{1\text{-itemsets}}(1\text{-itemset1}), P_{1\text{-itemsets}}(1\text{-itemset2}))$ (see Table. 3);

The stop condition of the algorithm is when there is no more generation of new itemsets candidate in order to return the set E = E1 ∪ E2 of all frequent itemsets in the $list_{RSF}$.

The generation of semantic association rules: after the construction of the set E corresponding to all significant itemsets in the $list_{RSF}$. We generate the semantic

Association rules [22]. A semantic association rule between C and S, is noted C→sem (S), and defined: C→sem (Si) ⇔exist Ci ∈ Dom (C), exist Sj ∈ Dom (S)/ Ci → Sj. The rule Ci → Si means if the $list_{RSF}$ is linked to the concept C by the semantic relation is- a (is, about), it must have the same type(is –about) with the concept S. So, the rule R: Ci → Si: represents the probability that the semantic content of $list_{RSF}$ covers Si knowing that it also covers Ci. This semantic interpretation, is based on two metrics which are : the confidence(conf) and the support (Sup).

The confidence associated to the rule R :conf(R: Ci →S_j) =P(C_i/S_j) is based on the degree of importance of Sj in the $list_{RSF}$, knowing the degree of importance of Ci in the $list_{RSF}$. It is defined as: Conf(C → sem S)=$max_{i,j}$(conf(R:C_i →S_j)) with C_i ∈ Dom(C),S_j ∈ Dom(S). But the support (Sup) is associated to a semantic association rule between entities Sup (Ci→ sem (Sj))=P(Ci→Sj)(probability of simultaneous occurrence of Ci and Sj). It is based on the number of rules of groups Ci ∈ (Domain(C_i)) and Sj ∈ Domain(S_j), having a support greater than or equal to the threshold sup_{min} (minimal support). The support of a rule is as follows:

$$Conf(R) = \frac{\min(WCI_{KB(listRSF)}, WSI_{KB(listRSF)})}{WCI_{KB(listRSF)}} \quad (6) \qquad Sup(R) = \frac{|\{Ci \rightarrow Sj / conf(Ci \rightarrow Sj) \geq conf_{min}\}|}{|\{Ci \rightarrow Sj / (Ci,Sj) \in Dom(C) \times Dom(S)\}|} \quad (7)$$

From the rules of semantic associations discussed above, we do build a semantic and contextual network of indexed concepts. This network represents the contents (subject) of the (Nsem $list_{RSF}$), and the contextual relations between them. An arc oriented from concept- node C to the concept- node S. C is the parent-node of S in the network.

Illustrative Example

Returning to the previous example, we apply the Apriori algorithm to extract the contextual relations. Other terms are the most frequent 1-itemset in the $list_{RSF}$, which are used to construct the 2-itemset (set of two terms. We calculate subsequently the weight of each 2-itemsets: P2-itemsets ({independence triumph}) = min (0.4, 0.6) = 0.4 ... etc.

We retain only the rules that have a confidence ≥ threshold of minConf = 1.These association rules are used to construct the semantic rules that form the basis for the identification of relation between concepts nodes $list_{RSF}$

The next step is the calculation of support for each semantic rule (sup (Rsemk: Ci → semSj)), with k = 1 ..n (number of semantic rules), the rules of semantic association whose support ≥ 0.5, like Rsem2 etc. Finally, the selected rules enable the selection of Semantic relations between concepts nodes of the $list_{RSF}$, in order to construct the semantic and contextual network presented by Figure Fig.2

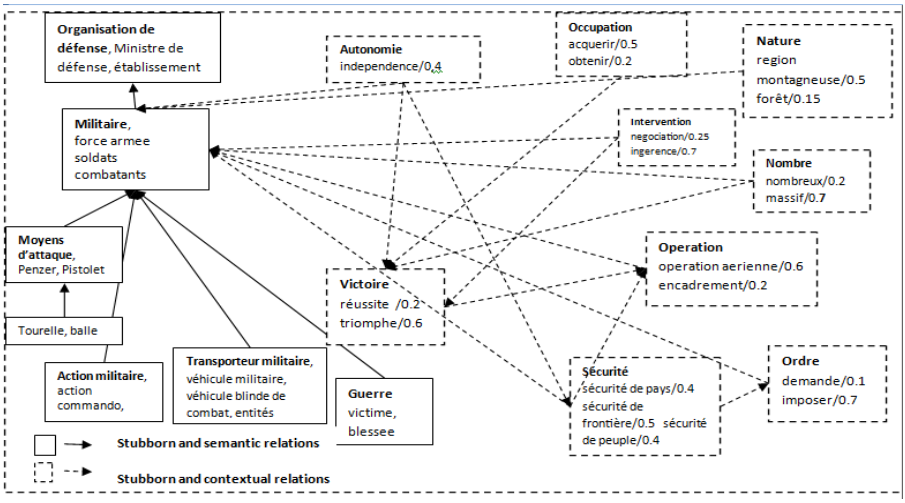


Fig. 2. Example of semantic and contextual network (nodes, arcs) from the $list_{RSF}$

4 Experimentation and Evaluation

4.1 Description of the Training Corpus

In our experiment, we used the Monde Diplomatic corpus composed of newspaper articles from the web [9]MD treats a variety of topics (geopolitics, international relations, economics, social issue, culture, etc.). This corpus is published in eight languages Arabic, French, English, Russian, Greek, Persian, Japanese, Chinese, and contains 414 articles. In our work, the used languages for the training are Arabic and French. The partition of the corpus contains 150 articles aligned in both languages. In these articles, we selected a training corpus of 200 pairs of bilingual aligned documents of size 0.6MB. These documents represent the list of sentences. This last contains objects and their properties in order to build the semantic network of the selected sentences. In addition, this list provides the application of association rules between the concepts in order to extract the contextual relations, which are used to enrich the semantic network.

4.2 Evaluation and Comparison between the Networks

The evaluation of the two networks (the semantic network and a semantic network with contextual enrichment) is established by comparing them to our reference network.

-Evaluation of the semantic network compared to the reference network: Our evaluation is inspired from the protocol based on semantic classes evaluation compared to a reference ontology. This evaluation relies on the comparison between the concepts appearing in the semantic network representing the Nsem ($list_{RSF}$) and the reference network case of the idea in [23]. Our reference network is developed by

an expert, we asked him to develop the best representation for the $list_{RSF}$. This reference network (Rref) is composed of about more than 600nodes and 790 relations.

The global evaluation of the lexical coverage of the network built over a network of reference is based on the use of a synthetic measure that combines the two measures of precision and recall. It is called F-measure (With $\beta = 1$; the two metrics precision and recall have the same importance). Table. 3 presents the different metrics of recall, precision and F-measure for a number of sentences from 10 to the complete $list_{RSF}$. These sentences are classified by their order of relevance.

Table 3. Recall, Precision, F-measure for the first evaluation

	10S	20S	30S	40S	60S	100S	$list_{RSF}$
Recall	52.7	56.8	60.6	72.3	72.8	70.3	65.2
Precision	51.7	55.9	60.4	69.4	70.8	69.8	68.3
F-measure	52.2	56.3	60.5	70.82	71.78	70.04	66.71

We can conclude that the results of recall, precision, F-measure are relatively close for the size of 40 to 60 sentences and we note a decrease of results for the 100 sentences and same for the whole $list_{RSF}$. Reducing the size of the $list_{RSF}$ to 60 sentences shows a good quality of semantic core in terms of lexical coverage. We reached a recall of 72.8%, a precision of 70.8% and an F-measure of 71.78%. The evaluation of the quality of the semantic core Nsem ($list_{RSF}$) in function of the number of sentences in the list (60) improves the response time of the machine translation system.

-Evaluation of the semantic network with contextual enrichment compared to the reference network: This evaluation is also based on the same standard measures. The table below shows the impact of the contextual enrichment of the semantic network in terms of global coverage. Our interest is to have good score of recall and precision that allow us to fix a threshold of support and confidence. So, which number of sentences in the $list_{RSF}$ allows us to get the best result? Table. 4 present the metrics of recall, precision and F-measure for 10 sentences to the complete $list_{RSF}$. These sentences are classified by their semantic relevance.

We notice that the contextual enrichment of the semantic network allows discovering hidden contextual links related to two parameters (support, confidence). These entities were absent in the network without enrichment. We reach in this case an F-measure from 67 to 82%, that corresponds to two threshold values of the support = 0.5, and confidence = 1.

This improvement is presented in the table below, we note that the use of contextual links in the semantic network with these two threshold values, increases the precision (80,3) and recall (81,9). This means that the contextual enrichment of this network covers almost the whole contents of the $list_{RSF}$. Concerning the size of the $list_{RSF}$, it may be an obstacle for its representation by a semantic network (contextually enriched). This treatment would require a lot of memory and computation time. Indeed, several sentences are classified last in the $list_{RSF}$, so we assume that their discrimination power is low. From the table below, we notice that with the first 90 sentences of the $list_{RSF}$ we obtain the best precision. This reduces the noise and increases the response time of the translation system.

Table 4. Recall, Precision, F-measure for the second evaluation

		10 S	20 S	30 S	40 S	60 S	80S	90S	100S	$list_{RSF}$
Support=0.5	**Recall**	55	61.4	66.7	79.5	80.2	80.9	81.4	81.4	81.4
Confidence=0.5	**Precision**	53	58.3	63.4	74	77.2	78.7	70.5	70.5	59
	F-measure	53.98	59.8	65	76.6	78.67	79.7	75.55	75.55	68.41
Support=0.5	**Recall**	54	60.1	64	77.8	80	81.9	81.9	81.9	81.9
Confidence=1	**Precision**	53	59.2	65	75.4	77.6	79.4	80.3	72	61
	F-measure	53.3	59.7	64.5	76.6	78.78	80.6	81.09	76.6	70
Support=0.4	**Recall**	55.7	60.8	66	78.8	81.2	82.7	82.5	82.5	82.6
Confidence=0.5	**Precision**	53	58	63	70.1	71.5	72.7	69.5	62	54.5
	F-measure	54	59.3	64.46	74.2	76.04	77.37	73.8	70.8	65.6
Support=0.4	**Recall**	55	59.6	65.2	77.5	80.6	81.8	82.2	82.2	82.2
Confidence=1	**Precision**	53.7	60	65	72	74.7	76.7	77.8	65	58.8
	F-measure	54.34	59.8	65.1	74.64	77.53	79.1	80	72.7	68.55
Support=0.3	**Recall**	54.1	59.6	63.5	74.6	77	79.6	79.6	79.6	79.6
Confidence=0.5	**Precision**	53	57.8	62.4	71.5	76.6	78.3	77	75.2	61.5
	F-measure	53.54	58.68	62.94	73	76.8	78.94	78.27	77.3	69.39
Support=0.3	**Recall**	54	59	63.2	74.2	76.5	78.8	78.8	78.8	87.8
Confidence=1	**Precision**	53	57.8	62.8	73	78.4	80.2	78	77.3	65
	F-measure	53.5	58.39	62.6	73.6	77.43	79.5	78.4	78/04	71.23

5 Conclusion and Future Works

In this paper, we build the semantic network of our $list_{RSF}$. This list contains the relevant sentences in French ($list_{RSF}$) that answer a given Arabic query. To build the network, we proposed first, a new approach of semantic and conceptual indexing. This approach is based on the combination of the statistical and semantic weighting using EWNF. Second we used a new method of disambiguation. As result we obtain a set of index- concepts forming the core of the network. Finally these index-concepts and their semantic relations form the semantic network. The evaluation results are satisfying (F-mesure=71%). Although the built network of the $list_{RSF}$ is incomplete; it does not cover all the semantic context of the list, that's why we decide to enrich it. Therefore, we used the context in order to gain the semantic richness existing in contextual relations between the indexes. These relations are extracted by the semantic associations rules based on two threshold (min-confidence and min-suport).

Our method is incorporated in our lexical disambiguation method. This last aims to improve Arabic queries translation in a bilingual information retrieval context: Arabic – French. we intend to find a compromise between the size of the generic base of rules and the time of response of our translation system.

References

1. Mallat, S., Hkiri, E., Zouaghi, A., Zrigui: Method of Lexical Enrichment in Information Retrieval System in Arabic. International Journal of Information Retrieval Research(IJIRR) 3(4) (Octobre 2013)
2. Mallat, S., Zouaghi, A., Zrigui: Proposal of a method of enriching queries by statistical analysis to search for information in Arabic. In: Conference Association for Machine Translation in the Americas (AMTA), San Diego, CA, USA, pp. 80–87 (2012)
3. Fleury, S., Zimina, M.: Exploring Translation Corpora with mkAlign. Translation Journal 11(1) (2002), http://accurapid.com/journal/39mk.htm
4. Church, K.W., Hanks, P.: Word Association Norms, Mutual Information, And Lexicography. Computational Linguistics 16(1) (1990)
5. Niwa, Y., Nitta, Y.: Co-occurrence Vectors from Corpora vs. Distance Vectors from Dictionaries. In: Proceedings of the 15th COLING Conference, Kyoto, Japan (1994)
6. Smadja, F.: Retrieving collocational knowledge from textual corpora. An application: Language generation, Doctoral Dissertation, Columbia University (1991)
7. Vossen, P., Peters, W., et al.: The Multilingual design of the EuroWordNet Database, of Lexical Semantic Resources for NLP Applications (1997), http://citeseer.nj.nec.com/cache/papers/cs/343/http:zSzzSzwww.let.uva.nlz.Sz~ewnzSzdocszSzP013.pdf
8. Leacock, C., Miller, G., Chodorow, M.: Using corpus statistics and WordNet relations for sense identification. Comput. Linguist. 24(1), 147–165 (1998)
9. Chiao, Y., Kraif, O., Laurent, D., Nguyen, T., Semmar, N., Stuck, F., Véronis, J., Zaghouani, W.: Evaluation of multilingual text alignment systems: the ARCADE II project. Actes de LREC-2006 (2006)
10. Harris, Z.: La genèse de l'analyse des transformations et de la métalangue. Langages 99, 9–20 (1990)
11. Bourigault, D., Aussenac-Gilles, N., Charlet, J.: Construction de ressources terminologiques ou ontologiques à partir de textes: un cadre unificateur pour trois études de cas. Revue d'Intelligence Artificielle (RIA) – Techniques Informatiques et Structuration de Terminologiques 18(1), 87–110 (2004)
12. Ferret, O.: Utiliser l'amorçage pour améliorer une mesure de similarité sémantique. In: Actes de TALN 2011, Montpellier, pp. 1–6 (2011)
13. Resnik, P.: Selection and Information: A Class-Based Approach to Lexical Relations. Thèse de doctorat, University of Pennsylvania (1993)
14. Caraballo, S.: Automatic acquisition of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics ACL 1999, pp. 120–126 (1999)
15. Baziz, M., Boughanem, M., Aussenac-Gilles, N.: Conceptual Indexing Based on Document Content Representation. In: Crestani, F., Ruthven, I. (eds.) CoLIS 2005. LNCS, vol. 3507, pp. 171–186. Springer, Heidelberg (2005)
16. Sanderson, M.: Word sense disambiguation and information retrieval. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 142–151. Springer (1994)
17. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: Proc. Workshop EACL SIGDAT, Dublin (1995)
18. Huang, X., Robertson, S.E.: Comparisons of Probabilistic Compound Unit Weighting Methods. In: Proc. of the ICDM 2001 Workshop on Text Mining, San Jose, USA (November 2001)

19. Baziz, M., Boughanem, M.: Nathalie Aussenac-Gilles. In: Ding, K., van Rijsbergen, I., Ounis, J. (eds.) The Use of Ontology for Semantic Representation of Documents. Dans: The 2nd Semantic Web and Information Retrieval Workshop(SWIR), Sheffield UK, juillet 29, pp. 38–45 (2004)
20. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research (JAIR) 11, 95–130 (1999)
21. Black, E.: An experiment in computational discrimination of english word senses, dans. IBM Journal of Research and Development 32(2), 185–194 (1988)
22. Zargayouna, H., Sylvie, S.: Mesure de similarité dans une ontologie pour l'indexation sémantique de documents XML. 2. 1 LIMSI/CNRS, Université Paris (2004), http://liris.cnrs.fr/~ic04/programme/articles/Zargayouna-IC2004.pdf
23. Oibeau, T., Dutoit, D., Bizouard, S.: Évaluer l'acquisition semi-automatique de classes sémantiques. In: Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2002). Nancy, France, pp. 37–38 (2002)

When was Macbeth Written? Mapping Book to Time

Aminul Islam, Jie Mei, Evangelos E. Milios, and Vlado Kešelj

Faculty of Computer Science,
Dalhousie University, Halifax, Canada
{islam,jmei,eem,vlado}@cs.dal.ca

Abstract. We address the question of predicting the time when a book was written using the Google Books Ngram corpus. This prediction could be useful for authorship and plagiarism detection, identification of literary movements, and forensic document examination. We propose an unsupervised approach and compare this with four baseline measures on a dataset consisting of 36 books written between 1551 and 1969. The proposed approach could be applicable to other languages as long as corpora of those languages similar to the Google Books Ngram are available.

1 Introduction

Given a book[1], we address the problem of how to predict the time frame (more specifically, a year) the book was written or published. For most books, the time difference between 'year written' and 'year published' is relatively short (e.g., one or two years). Thus, these two terms could be used interchangeably. If year written and year published differ significantly then written years are taken into account. This mapping task has applications in authorship and plagiarism detection [1], identification of literary movements [2], finding literary period of a book, forensic document examination [3], where a key component is to determine the time frame in which it was created.

The proposed approach uses the Google Books Ngram corpus (Version 2) [4,5] and predicts the year a book was written. A collection of 36 English books written between years 1551 and 1969 is used to evaluate the approach. A subset of the Ngram corpus (Version 2), which contains predominantly the English language, is used for this work. However, the proposed approach is general enough to be applicable to the seven other languages (i.e., Chinese, French, German, Hebrew, Spanish, Russian, and Italian) that the Ngram corpus (Version 2) supports. Our contributions are summarized as follows:

- An unsupervised approach to map a book to a year it was written.
- A dataset of 36 books between year 1551 and 1969 is compiled and made publicly available for future research on this type of task.

[1] Though our evaluation datasets are books, the proposed approach is general enough to be applied to document or text of any size.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 73–84, 2015.
DOI: 10.1007/978-3-319-18111-0_6

- An evaluation metric for prediction quality for the task is presented.
- Four baseline measures are introduced to compare with the proposed approach.

The rest of this paper is organized as follows: the Google Books Ngram Corpus is briefly discussed in Section 2. A brief overview of the related work is presented in Section 3. The proposed approach to map a book to a year is described in Section 4. A brief description of the evaluation dataset, evaluation metrics, baseline measures and the experimental results is in Section 5. We summarize contributions and future related work in Conclusion.

2 Google Books Ngram Corpus

The Google Books Ngram corpus (Version 2) [4,5] was generated in July 2012 from 8,116,746 books in eight languages, or over 6% of all books ever published over a period of five centuries. The corpus consists of words and phrases (i.e., ngrams, where the value of n could be one to five) and their usage frequency over time. The English corpus (henceforth, English 2012), a subset of the Google Books Ngram corpus, comprises 468,491,999,592 words from 4,541,627 books predominantly in the English language published in any country between year 1505 and 2008. The 1-gram statistics of English 2012 (henceforth, English 2012 1-grams) without any syntactic information (e.g., part-of-speech tag) are used in this work.

As an example, here are the 25,944,417th and 25,944,418th lines from a file of the English 2012 1-grams (googlebooks-eng-all-1gram-20120701-r.gz):

```
revolutionizer 1865    4       3
revolutionizer 1866    10      10
```

The first line tells us that in 1865, the word "revolutionizer" occurred four times overall, in three distinct books of the sample the corpus was generated.

The English 2012 corpus also contains a file named "total_counts", which records the total number of 1-grams per year contained in the books that make up the corpus. More specifically, the file contains one triplet of values (match_count, page_count, volume_count) per year. This file is useful for computing the relative frequencies of ngrams.

As an example, here are the 15th and 16th lines from the "total_counts" file of the English 2012:

```
1579,203074,1143,3
1581,708458,2824,6
```

The first line tells us that in the year 1579, there were 203,074 words in 1,143 distinct pages and in three distinct books.

3 Related Work

Related works that are relatively close to the task addressed here are the identification of literary movements, genre and authorship classification. For example, in [2] texts, represented by topological metrics of complex networks, are used from books dating from 1590 to 1922 in an attempt to verify changes in writing style. With multivariate statistical analysis of the metrics, six clusters of books are generated where clusters correspond to major literary movements. In [6], text classification by literary period using Prediction by Partial Matching (PPM) statistical model is tested to classify literature style for texts from Brazilian literature. In [7], only simple variations of basic syntactic features of function words and part-of-speech tags are considered for authorship classification. In [8], it is argued that genre detection based on surface cues is as successful as detection based on deeper structural properties.

Another related area is to determine whether a book or text was written by a particular author [9,10,1]. For example, in [9] word usage by Shakespeare is analyzed to answer whether he wrote a newly-discovered poem. In [1], a simplified approach is examined for unsupervised authorship and plagiarism detection which is based on binary bag of words representation where each document is represented as a binary vector that encodes the presence or absence of common words in the text.

4 Proposed Approach

The proposed approach uses the hypothesis that word-use pattern in terms of frequency correlates with a time period. The idea is to find the year that best correlates with the word-use pattern of a book in terms of frequency. Frequencies of unique words in a book and the normalized frequencies of the same unique words present in each year in the corpus are computed. Then, the correlations between frequencies of unique words in the book and the normalized frequencies of the same unique words in each year in the corpus are computed. The approach predicts that the book was written in the year with the highest correlation.

Given a book B containing n unique words[2] (i.e., $W = \{w_1, w_2, \cdots, w_n\}$) and the English 2012 1-grams corpus (or, a corpus similar to this), the task is to find the prediction year y from a *totally ordered set* $(Y, <)$ of m unique years (i.e., $Y = \{y_1, y_2, \cdots, y_m\}$) found in the English 2012 1-grams.

As $y \in Y$, there is a possibility that prediction might get better if it is possible to reduce the prediction space, i.e., the size of Y. That is to find another *totally ordered set* $(Y', <)$ of $m - l + 1$ unique years (i.e., $Y' = \{y_l, y_{l+1}, \cdots, y_m\}$), where $m - l + 1 \leq m$. This could be achieved if a word in the text of a book is not present in the corpus before year y_l. The intuition is that if the word is not available in the corpus before year y_l, it is very unlikely that the book was

[2] In preprocessing step, special characters, punctuation and a set of stop words (33 most frequent words in Google ngram corpus [11]) are removed from the text of the book.

written before that year. Given a book and the set of years (Y) present in the corpus, Algorithm 1 computes y_l and the new set of years (Y'), where it is likely that $|Y'| < |Y|$.

Algorithm 1. Reducing prediction space

input : A book having a set W of n unique words (i.e., $W = \{w_1, w_2, \cdots, w_n\}$), and a *totally ordered set* $(Y, <)$ of m unique years (i.e., $Y = \{y_1, y_2, \cdots, y_m\}$) found in the English 2012 1-grams
output: A *totally ordered set* $(Y', <)$ of $m - l + 1$ unique years (i.e., $Y' = \{y_l, y_{l+1}, \cdots, y_m\}$), where $m - l + 1 \leq m$

```
1 y_l ← 0
2 foreach w ∈ W do
3 |   m_n ← minimum year in the set Y where w appears
4 |   if m_n > y_l then
5 |   |   y_l ← m_n
6 |   end
7 end
8 Y' ← {y|y ∈ Y ∧ y ≥ y_l}
```

Given a book B and the English 2012 1-grams corpus, Algorithm 2 computes the year, y, predicted as when the book was written based on the hypothesis already mentioned. In Line 2 to 5, Algorithm 2 stores the frequencies of all the unique words present in the book in an array. The prediction space, Y, is reduced to Y' in Line 6 using Algorithm 1. Frequencies of the same unique words that are also present in each year in the corpus are computed and normalized by the number of books present in that year in the corpus. These normalized frequencies of all the years are stored in a list of arrays (Line 8 to 15). The Pearson's correlation coefficients between frequencies of unique words in the book and the normalized frequencies of the same unique words in each year are computed. The year with highest correlation is computed (Line 18 to 25). The approach predicts that the book was written in the year with highest correlation.

5 Evaluation and Experimental Results

The procedure of compiling evaluation dataset, evaluation metrics, four baseline measures and the results on the evaluation dataset are discussed in this section.

5.1 Evaluation Dataset

To evaluate the proposed approach, 36 books written between year 1551 and 2000 are selected. To make the dataset balanced, equal number of books are selected from the following nine time periods each with length of 50 years: 1551-1600, 1601-1650, 1651-1700, 1701-1750, 1751-1800, 1801-1850, 1851-1900,

Algorithm 2. Predicting year

input : A book B having a set W of n unique words (i.e., $W = \{w_1, w_2, \cdots, w_n\}$), and the English 2012 1-grams

output: A year $y \in Y$, where Y is a *totally ordered set* $(Y, <)$ of m unique years (i.e., $Y = \{y_1, y_2, \cdots, y_m\}$) found in the English 2012 1-grams

$i \leftarrow 1$
while $i \leq |W|$ **do**
  $L(i) \leftarrow \text{freq}(w_i, B)$ // returns the frequency of w_i in B, and L is an array of length $|W|$
  increment i
end
$Y' \leftarrow$ Algorithm 1(W, Y) // prediction space is reduced using Algorithm 1
$j \leftarrow 1$
while $j \leq |Y'|$ **do**
  $i \leftarrow 1$
  while $i \leq |W|$ **do**
    $L_j(i) \leftarrow \frac{\text{freq}(w_i, y_j)}{\text{Number of books in year} y_j}$ // returns the normalized frequency of w_i in year y_j in the English 2012 1-grams, and L_j is an array of length $|W|$
    increment i
  end
  increment j
end
$max\_r \leftarrow 0$
$j \leftarrow 1$
while $j \leq |Y'|$ **do**
  $c \leftarrow \text{PEARSON}(L, L_j)$ // returns the Pearson's correlation coefficient between L and L_j
  if $c > max\_r$ **then**
    $max\_r \leftarrow c$
    $y \leftarrow Y'(j)$ // returns jth year in the set Y'
  end
  increment j
end
return y

1901-1950, 1951-2000. For each time period, four books are selected based on the "Best Books of the Nth Century" (where $N \in \{16, 17, 18, 19, 20\}$) from goodreads.com [12]. Texts of most books except some 20th century ones are extracted from Project Gutenberg [13]. When selecting a book, some criteria e.g., public availability and English as its written language are taken into account. Publication dates, if present in the texts, are removed. The selected books and years written are shown in the first and second columns, respectively, in Table 1. This dataset is made publicly available[3] for future research.

5.2 Evaluation Metrics

The effectiveness of the mapping of a book to a year is evaluated with a quality measure. Let $Y = \{y_1, y_2, \cdots, y_m\}$ be the set of each distinct year a book could be mapped onto, and $I = \{1, 2, \cdots, m\}$ be the index set of Y. The idea is to consider each year in the set as a node of a path graph or linear graph, G, and measure how close the predicted node is to the actual node compared to all the nodes. The closeness is measured by the number of edges between the predicted node and the actual node. The smaller the number of edges between the predicted node and the actual node, the closer they are. Thus if a document actually written in year y_a is predicted as written in year y_p, then the **P**rediction **Q**uality (PQ) of a book is computed by the complement of the number of edges between a and p (i.e., $|a - p|$) over the total number of edges in G (i.e., $m - 1$) as shown in the following formula[4]:

$$\begin{aligned} \mathrm{PQ}(a, p) &= 1 - \frac{|a - p|}{m - 1} \\ &= \frac{m - |a - p| - 1}{m - 1} \end{aligned} \tag{1}$$

For example, given $Y = \{1987, 1993, 1995, 1999, 2001, 2002, 2004, 2007\}$, and thus $I = \{1, 2, \cdots, 8\}$, if a document actually written in the year 1999 is predicted as written in the year 2004, PQ would be $\frac{8-|4-7|-1}{8-1}$, or 57.1%.

5.3 Baseline Measures

The following four baseline measures are considered in this work:

Baseline Measure 1. If a year is to be chosen, one appropriate choice, at least probabilistically, would be the year with maximum number of books in the corpus. The first baseline measure always chooses this year as the solution year and then use Equation 1 to compute the PQ.

In evaluation, the first baseline measure always chooses the year 2008 with a maximum of 206,272 books present in the English 2012 corpus compared to only one book in the year 1505.

[3] `http://web.cs.dal.ca/~eem/`

[4] These notations are used throughout this section.

Baseline Measure 2. Choosing all possible years as solution years is considered as the second baseline measure. The PQ is computed as the average PQ of all the solution years by the following formula:

$$\begin{aligned} \mathrm{PQ}_{\mathrm{b2}} &= \frac{\sum_{i=1}^{m} \mathrm{PQ}(a,i)}{m} \\ &= \frac{m^2 - m - \sum_{i=1}^{m} |a-i|}{m^2 - m} \end{aligned} \tag{2}$$

Baseline Measure 3. In this baseline measure, the middle member in Y is always chosen as the solution year. The PQ is computed by the following formula:

$$\begin{aligned} \mathrm{PQ}_{\mathrm{b3}} &= \mathrm{PQ}(a, \frac{m}{2}) \\ &= \frac{m - |a - \frac{m}{2}| - 1}{m-1} \end{aligned} \tag{3}$$

In evaluation, the third baseline measure always chooses the year 1796 based on the English 2012 corpus.

Baseline Measure 4. Given a collection of N books published over a period of years denoted by Y, the fourth baseline measure considers a year $y_h \in Y$ as the solution year provided that the total number of books published since year $y_1 \in Y$ till year y_h is approximately $\frac{N}{2}$. This is the year that equally divides all the books in the corpus. It is equally probable that an unknown book could be written before or after this year. Based on the English 2012 corpus, this year is 1985 as shown in Figure 1. The PQ is computed by the following formula:

$$\begin{aligned} \mathrm{PQ}_{\mathrm{b4}} &= \mathrm{PQ}(a,h) \\ &= \frac{m - |a-h| - 1}{m-1} \end{aligned} \tag{4}$$

5.4 Results

The detailed experimental results on the 36 books dataset are shown in Table 1. Book names, years the books were written, and the years predicted by the proposed approach are in the first three columns, respectively. Column four to column eight shows the prediction quality in % for the proposed approach and the four baseline measures. Baseline measure 1, 3, and 4 choose year 2008, 1796, and 1985, respectively, based on the English 2012 corpus. For all the 36 books, the proposed approach and the best among the four baseline measures achieve 82.3% and 75.1% PQ, respectively. One of the reasons that Baseline measure 3 achieves good PQ, specially for the 18th and 19th century books, is that the dataset is balanced and the measure always chooses the middle year from the set of years. It is obvious that Baseline measure 4 does better for the 20th century books. The proposed approach predicts better for the 16th, 17th, and 18th century books. For the 20 books from these three centuries, the proposed approach achieves 89.8% PQ, compared to 73.4% PQ by the best baseline measure. It is

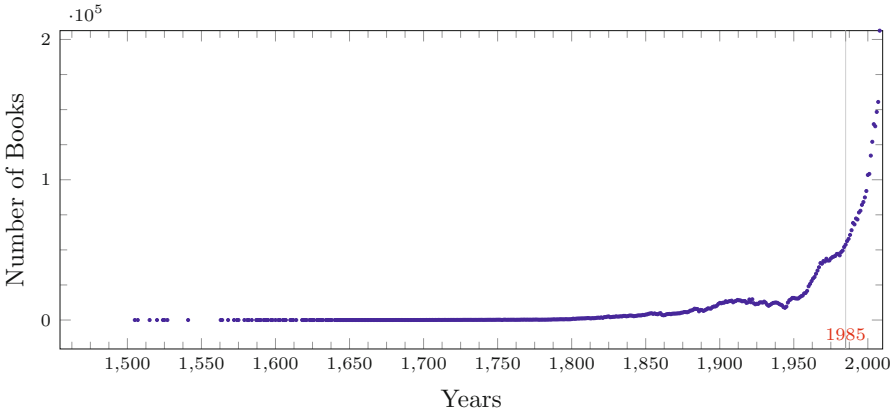


Fig. 1. Number of books versus years in the English 2012 corpus

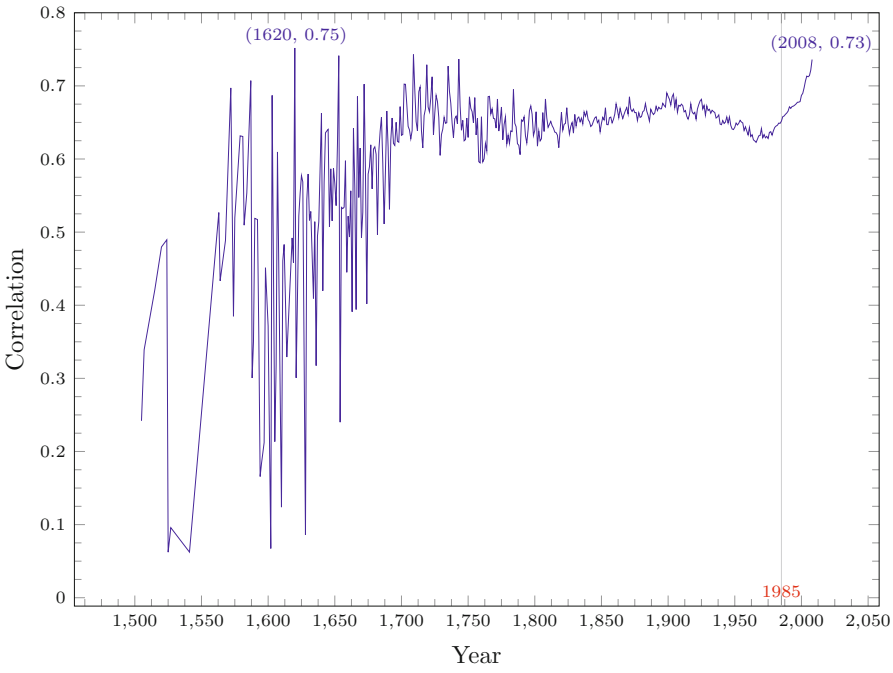


Fig. 2. Year-wise correlations for Macbeth

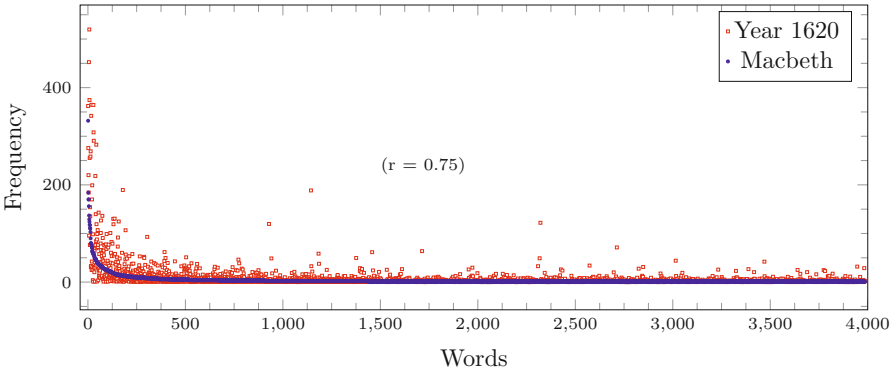


Fig. 3. Correlation for Macbeth in the year 1620

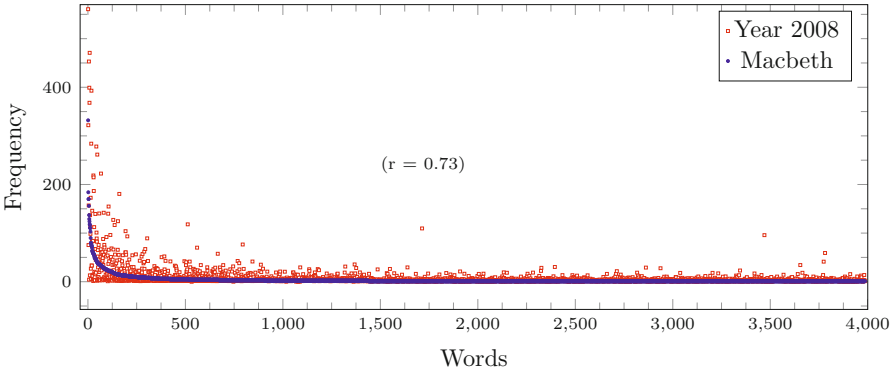


Fig. 4. Correlation for Macbeth in the year 2008

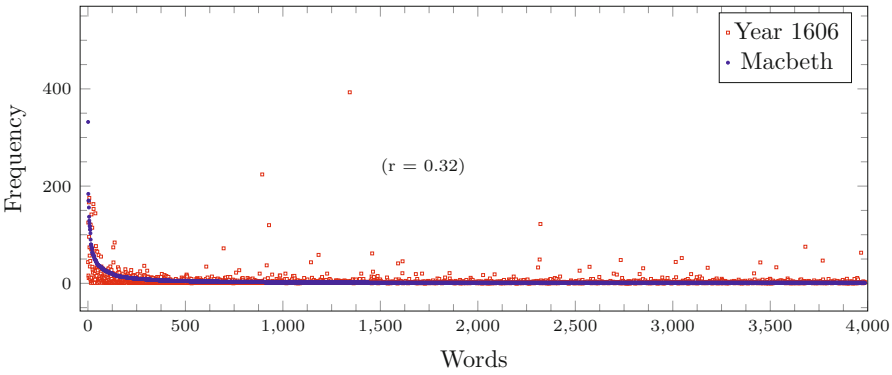


Fig. 5. Correlation for Macbeth in the year 1606

Table 1. Results on the 36 books dataset

Book Name	Written Year	Predicted Year by Proposed Approach	Prediction Quality (PQ) in %				
			Proposed Measure	Base-line 1	Base-line 2	Base-line 3	Base-line 4
Utopia	1551	1590	**96.7**	1.7	51.6	51.7	7.1
Romeo and Juliet	1595	1587	**98.4**	5.9	55.5	55.9	11.3
The Merchant of Venice	1596	1587	**98.4**	5.9	55.5	55.9	11.3
Julius Caesar	1599	1653	**90.1**	6.4	56.0	56.4	11.8
Twelfth Night	1601	1587	**97.6**	6.6	56.2	56.6	12.0
Hamlet	1602	1620	**97.4**	6.8	56.4	56.8	12.3
Othello	1603	1587	**97.2**	7.1	56.6	57.1	12.5
Macbeth	1606	1620	**98.1**	7.5	57.0	57.6	13.0
Leviathan	1651	1651	**100**	15.8	63.3	65.8	21.2
Second Treatise of Government	1689	1756	**84.2**	24.8	68.6	74.8	30.2
An Essay Concerning Human Understanding Volume 1	1689	1751	**85.4**	24.8	68.6	74.8	30.2
The Practice of the Presence of God	1691	1719	**93.4**	25.2	68.8	75.2	30.7
Gulliver's Travels	1726	1587	70.8	33.5	72.2	**83.5**	38.9
A Modest Proposal	1729	1735	**98.6**	34.2	72.5	84.2	39.6
Clarissa Vol. 1	1748	1587	65.6	38.7	73.7	**88.7**	44.1
The History of Tom Jones, a Foundling	1749	2008	38.9	38.9	73.7	**88.9**	44.3
The Wealth of Nations	1776	1776	**100**	45.3	74.7	95.3	50.7
Common Sense	1776	1745	92.7	45.3	74.7	**95.3**	50.7
The History of the Decline and Fall of the Roman Empire	1776	1789	**96.9**	45.3	74.7	95.3	50.7
A Vindication of the Rights of Woman	1792	1774	95.8	49.1	74.9	**99.1**	54.5
Average (20 books between 1551 and 1792)			**89.8**	23.4	65.3	73.4	28.9
Sense and Sensibility	1811	2008	53.5	53.5	74.8	**96.5**	59.0
Pride and Prejudice	1813	2008	54.0	54.0	74.8	**96.0**	59.4
Emma	1815	2008	54.5	54.5	74.7	**95.5**	59.9
Persuasion	1816	2008	54.7	54.7	74.7	**95.3**	60.1
Great Expectations	1861	2008	65.3	65.3	72.6	**84.7**	70.8
Alice's Adventures in Wonderland	1865	2008	66.3	66.3	72.3	**83.7**	71.7
Adventures of Huckleberry Finn	1884	2008	70.8	70.8	70.6	**79.3**	76.2
The Picture of Dorian Gray	1890	2008	72.2	72.2	70.0	**77.8**	77.6
Dubliners	1914	2008	77.8	77.8	67.2	72.2	**83.3**
The Mysterious Affairs at Style	1920	2008	79.3	79.3	66.4	70.8	**84.7**
The Great Gatsby	1925	2008	80.4	80.4	65.7	69.6	**85.9**
Brave New World	1931	2008	81.8	81.8	64.8	68.2	**87.3**
Farenheit 451	1953	2008	87.0	87.0	61.3	63.0	**92.5**
Lord of the Flies	1954	2008	87.3	87.3	61.1	62.7	**92.7**
To Kill a Mockingbird	1960	2008	88.7	88.7	60.0	61.3	**94.1**
Slaughterhouse-Five	1969	1935	92.0	90.8	58.3	59.2	**96.2**
Average (All 36 books between 1551 and 1969)			**82.3**	45.4	66.5	75.1	50.8

obvious that the used hypothesis is not functioning to distinguish 19th and 20th century books.

The correlations between frequencies of unique words in the book Macbeth and the normalized frequencies of the same unique words in each year in the corpus are shown in Figure 2. For Macbeth, year 1620 achieves the highest

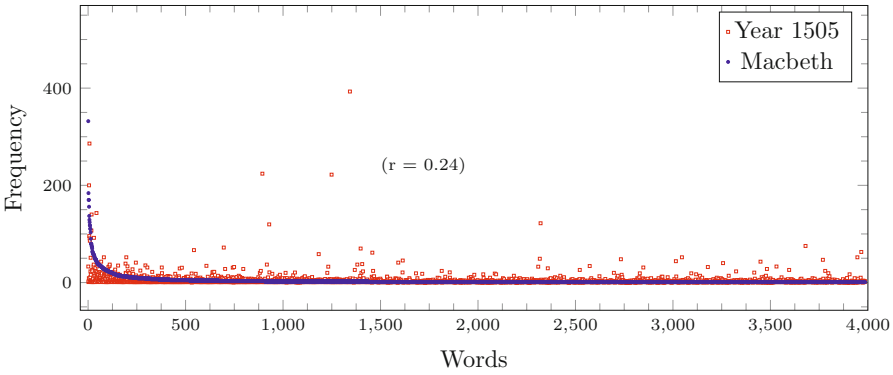


Fig. 6. Correlation for Macbeth in the year 1505

correlation, whereas year 2008 achieves the second highest correlation. Around year 1985 and onwards, the correlation curve sharply increases. One of reasons is the drastic increase in the number of books since year 1985 as shown in Figure 1. The number of books in the English 2012 corpus in 480 years before 1985 equals the number of books in just 23 years after 1985. As a result, word-use in the 19th and 20th century takes shape into an uniform pattern irrespective of the time period. This is one of the reasons why the proposed approach shows highest correlation in the year 2008 for 15 out of 16 books in the 19th and 20th centuries.

As an example, the correlations between frequencies of 3,985 unique words in Macbeth and the normalized frequencies of the same unique words from the corpus in the year 1620, 2008, 1606, and 1505 (out of 425 distinct years in the prediction space) are shown in Figure 3, 4, 5, and 6, respectively. As the correlation in the year 1620 outperforms the correlations in other years, the proposed approach predicts that Macbeth was written in the year 1620 and achieves 98.1% PQ. For two books ("Leviathan" and "The Wealth of Nations"), the PQ achieved by the proposed approach is 100%.

6 Conclusion

The proposed approach for predicting the time a book was written based on the hypothesis that word-use pattern in terms of frequency correlates with a time period achieves a reasonably good prediction quality (89.8%) for the 16th, 17th, and 18th century books compared to the best baseline measure (73.4%). The approach is unsupervised and general enough to be applicable to other languages. One interesting finding is that word-use in the 19th and 20th centuries tends to converge into an uniform pattern irrespective of time period because the number of books in these two centuries is very large compared to the immediate past three centuries. Future work would be to look into a new hypothesis that best correlates recent centuries and achieves better prediction quality for the 19th and 20th century books.

Acknowledgments. This research was funded by the Natural Sciences and Engineering Research Council of Canada and the Boeing Company.

References

1. Akiva, N.: Authorship and plagiarism detection using binary bow features. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop) (2012)
2. Amancio, D.R., Oliveira, O.N., da Fontoura Costa, L.: Identification of literary movements using complex networks to represent texts. New Journal of Physics 14, 043029 (2012)
3. A simplified guide to forensic document examination (2013), http://www.crime-scene-investigator.net/SimplifiedGuideQuestionedDocuments.pdf (accessed: February 7, 2015)
4. Michel, J.B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Team, T.G.B., Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., Aiden, E.L.: Quantitative analysis of culture using millions of digitized books. Science 331, 176–182 (2011)
5. Lin, Y., Michel, J.B., Aiden, E.L., Orwant, J., Brockman, W., Petrov, S.: Syntactic annotations for the Google books ngram corpus. In: Proceedings of the ACL 2012 System Demonstrations, ACL 2012, pp. 169–174. Association for Computational Linguistics, Stroudsburg (2012)
6. Barufaldi, B., Santana, E., Filho, J., van der Poel, J., Marques, M., Batista, L.: Text classification by literary period using ppm-c data compression. In: 2009 Seventh Brazilian Symposium in Information and Human Language Technology (STIL), pp. 125–133 (2009)
7. Kim, S., Kim, H., Weninger, T., Han, J.: Authorship classification: A syntactic tree mining approach. In: Proceedings of the ACM SIGKDD Workshop on Useful Patterns, UP 2010, pp. 65–73. ACM, New York (2010)
8. Kessler, B., Numberg, G., Schütze, H.: Automatic detection of text genre. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL 1998, pp. 32–38. Association for Computational Linguistics, Stroudsburg (1997)
9. Thisted, R., Efron, B.: Did Shakespeare write a newly-discovered poem? Biometrika 74, 445–455 (1987)
10. Thompson, J.R., Rasp, J.: Did C. S. Lewis write The Dark Tower?: An examination of the small-sample properties of the Thisted-Efron tests of authorship. Austrian Journal of Statistics 38, 71–82 (2009)
11. Brants, T., Franz, A.: Web 1T 5-gram corpus version 1.1. Technical report, Google Research (2006)
12. http://www.goodreads.com/ (accessed: January 15, 2015)
13. https://www.gutenberg.org/ (accessed: January 15, 2015)

Tharawat: A Vision for a Comprehensive Resource for Arabic Computational Processing

Mona Diab

Department of Computer Science,
The George Washington University,
Washington, DC, USA
mtdiab@gwu.edu

Abstract. In this paper, we present a vision for a comprehensive unified lexical resource for computational processing of Arabic with as many of its variants as possible. We will review the current state of the art for three existing resources and then propose a method to link them in addition to augment them in a manner that would render them even more useful for natural language processing whether targeting enabling technologies such as part of speech tagging or parsing, or applications such as Machine Translation, or Information Extraction. The unified lexical resource, Tharawat, meaning treasures, is an extension of our core unique resource Tharwa, which is a three way computational lexicon for Dialectal Arabic, Modern Standard Arabic, and English lemma correspondents. Tharawat will incorporate two other current resources namely SANA, our Arabic Sentiment Lexicon, and MuSTalAHAt, our Multiword Expression (MWE) version of Tharwa but instead of listing lemmas and their correspondents, it lists MWE and their correspondents. Moreover, we present a roadmap for incorporating links for Tharawat to existing English resources and corpora leveraging advanced machine learning techniques and crowd sourcing methods. Such resources are at the core of NLP technologies. Specifically, we believe that such a resource could lead to significant leaps and strides for Arabic NLP. Possessing them for a language such as Arabic could be quite impactful for the development of advanced scientific material and hence lead to an Arabic scientific and economic revolution.

1 Introduction

The Arabic language has garnered a lot of attention due to its significance, being the language spoken by over 300M people worldwide but also it is the language spoken by countries of strategic political presence in the world. It is one of the 6 official languages of the United Nations. Moreover the Arabic script is used as the writing script for several non Arabic speaking countries such as Farsi (the language spoken in Iran), and Dari (the language spoken in Afghanistan), also among several of the ex USSR countries. Recently Arabic has drawn even more attention due to the series of political revolutions, aka the Arab Spring, taking place in the Arab world, from Tunisia, to Egypt, to Libya, to Syria, to Yemen.

Social media is ubiquitous in this current information age. With the explosion of social media, the language of Web 2.0 is undergoing fundamental changes: English is

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 85–97, 2015.
DOI: 10.1007/978-3-319-18111-0_7

no longer dominating the web, and user generated content is outpacing professionally edited content. User generated content is re-shaping the way people are consuming and dealing with information, as the user is no longer a passive recipient, but has now turned into an active participant, and in many instances, a source or producer of information. Social media have empowered users to be more creative and interactive, and allowed them to voice their opinions on events and products and exert powerful influence on the behavior and opinion of others. Yet, the current overflow of user generated content poses significant challenges in data gathering, annotation and presentation. Facebook and Twitter virtually replaced traditional news media as a medium for exchanging information. Social media played a crucial role in the Arab Spring as it became the platform for news as well as communication at a high pace between revolutionaries within a country and across country borders. For example, Tunisians were helping Egyptians mitigate violence and tear gas, for instance, during the initial days of the Egyptian 25th of January 2011 revolution.

Needless to say, the electronic media is inundated with data and mining it for information is crucial for understanding the landscape of what is happening. But different from textual/speech or video social media in other languages, for Arabic, the issue is not simply a difference in level of language formality between traditional media outlets and the more informal language used in online social media. Spoken media being a reflection of actual spoken vernaculars in the Arab world represents a treasure trove of documentation of the rich and complex map of the variant forms of Arabic that are significantly different from one another.

The Arabic language is an aggregate of multiple varieties including a standard used in education and official settings known as Modern Standard Arabic (MSA) and a number of spoken vernaculars comprising the dialectal variants of the language, collectively known as Dialectal Arabic (DA). DA are emerging as a significant set of language varieties for textual processing due to their pervasive and ubiquitous presence online especially in the current influx of social media. The differences between DAs and MSA go beyond register differences as is typical in other languages (formal vs. informal). Coarsely, the two varieties of Arabic, MSA and DA, co-exist in a state of diglossia [8], in a relative complementary distribution but crucially they differ significantly from one another on the morphological, phonological and lexical levels of linguistic representation. One wonders how the Egyptian revolutionaries corresponded with the Tunisians or Yemenis during the Arab spring. Probably such communications occurred mostly in MSA. But the interesting aspect of the variations is that a significant proportion of the lexical items look the same when spelled out but in many cases, they have pragmatic usage variations, hence they are used differently. DAs differ from one another significantly but also from MSA. Such differences have a direct impact on Arabic processing tools. Most automatic resources exist for MSA leading to an abundance of tools for processing this variety but given the significant difference between MSA and DA, we note a sharp drop in performance for the tools when applied to DA. Differences on the lexical level are especially interesting since many surface word forms are homographically similar across naturally occurring written Arabic variants in particular in the absence of short vowel representation –aka diacritics. Many of these forms are not semantic cognates which leads to significant deterioration in computational performance.

To date, a notable gap exists for DA resources especially ones that bridge across variants and in turn to English (as a representative of a language with abundant automatic resources). Some computational approaches to dialectal processing such as [2] and [18] have addressed the gap by approximations via extending BAMA/SAMA databases (Buckwalter, 2004; Graff et al., 2009) to accept DA prefixes and suffixes. This is, however, a shallow process that is limited to a subset of the lexicon shared by both MSA and DA.

Hence, the creation of different resources such as lexicons is crucial from a computational point of view. Linguistically, a resource that fills this lexical gap can lead to more thorough analysis of DA content leading to better insights into the nature of these varieties and how they are being used and what is their exact relation with MSA. Moreover, this could potentially lead to interesting research in theoretical linguistics, sociolinguistics, comparative linguistics, lexical semantics, lexicography and discourse analysis. Furthermore, building such resources for an influential language such as Arabic could have a profound impact on the scientific landscape of the Arab world where such resources can serve as precursors to the building home grown locally state of the art technologies for the Arabic speaking world by Arabs in the Arab world leading to real scientific progress, thereby leading to different kinds of revolutions, Thawarat, specifically scientific and technological ones.

Accordingly, we introduce a vision for a new resource that is unified in structure aiming at combining three existing resources for Arabic. The new lexical resource, Tharawat, is designed to be a comprehensive Arabic(s) resource with links to English and corpora in whichever language of interest where available. Tharawat will comprise at its core three different yet connected resources: Tharwa [7]; SANA [1] a large scale multi-genre, multi-dialect lexicon for Arabic subjectivity and sentiment analysis; and, MuSTalAHAt, [14], an extension of Tharwa but instead of comprising lemmas, MuSTalAHAt comprises multiword expressions (MWE). The current more mature versions for all these resources exist for Egyptian Arabic as a representative of DA. We are in the process of independently augmenting all three resources for other Arabic dialects such as Iraqi, Levantine, Tunisian, Moroccan, Gulf and Algerian. In this paper, we will focus on Egyptian as a representative of DA.

In the following, we discuss some related work then we provide a brief description of Egyptian Arabic, followed by a description of elements from the three core resources and how we plan to merge them and augment them with other pertinent information leading to our comprehensive resource Tharawat.

2 Related Work

The most comparable resource to Tharawat in spirit, being a comprehensive resource, is the English Unified Verb Index (UVI) project.[1] The UVI comprises a total of 8537 verbs represented with 6340 VerbNet links, 273 VerbNet main classes, 214 VerbNet subclasses, 5649 PropBank links, 4186 FrameNet links, 4898 Grouping links, in addition to links to WordNet internally. It took several years to compile and it is monolingual.

[1] `http://verbs.colorado.edu/verb-index/`

To our knowledge, there are no three way resources created for the Arabic language specifically. There exist two way resources for DA-English/French/etc, or MSA-English/French/etc, however none for MSA-DA, and less even resources for DA-MSA-English/French/etc. For example, unlike for MSA, EGY has a small number of printed (bilingual or monolingual) Dictionaries. [19] is the first recorded dictionary of the Egyptian dialect, and its modern reproduction [20] contains 12,500 EGY-ENG entries. [5] compiled *A Dictionary of Egyptian Arabic* which is the most comprehensive and complete dictionary in print for EGY, consisting of more than 31K EGY-ENG single word entries. Both dictionaries target English non-Arabic speakers learning EGY. Other paper based dictionaries exist for other Arabic dialects, the most renowned of which is the Georgetown series comprising bidirectional dictionaries for the following dialects: English-Syrian,[2] Iraqi English,[3] and Moroccan English dictionaries.[4]

Machine Readable Dictionaries (MRDs) of EGY have appeared with varying degrees of coverage and linguistic sophistication. The Egyptian Colloquial Arabic Lexicon (ECAL) by the Linguistic Data Consortium (LDC) [15] is a monolingual lexicon of fully inflected words (surface forms) that consists of over 66K monolingual entries. ECAL is used by [10] to produce the CALIMA morphological analyzer for EGY. The Columbia Egyptian Colloquial Arabic Dictionary (CECAD) [16] is a small EGY-MSA-ENG dictionary consisting of 1,752 high frequency words. CECAD is a subset of ECAL manually augmented with MSA and ENG equivalents.

3 Description of Egyptian Arabic

In characterizing DA, EGY stands in a cluster of its own due to its significant difference from MSA and other Arabic varieties [6]. It is one of the most widespread varieties of Arabic due to the fact that it is the native tongue of more than 90 million contemporary Arabs (which makes up for close to one third of the Arabic-speaking world), along with the strategic and cultural importance of Egypt, but also the media impact of Egypt is quite widespread leading to EGY being very well understood by most non-Egyptian Arabs. EGY exhibits considerable differences from MSA at multiple levels of linguistic representation. We will briefly address here only the morphological, phonological, and lexical variation from MSA without touching upon the syntactic differences. For more information on EGY differences from MSA, see [12].

3.1 Phonological Variation

As is the case for many languages and their dialects, the pronunciation of some MSA phonemes have shifted in EGY. Some of the shifts are quite regular such as /q/ (of the letter ق) becoming a glottal stop /'/ except for few words borrowed from MSA or Classical Arabic, e.g., the word قلب 'heart' is pronounced /qalb/ in MSA but /'alb/

[2] http://press.georgetown.edu/book/languages/dictionary-syrian-arabic

[3] http://press.georgetown.edu/book/languages/georgetown-dictionary-iraqi-arabic

[4] http://press.georgetown.edu/book/languages/dictionary-moroccan-arabic

in EGY. Another example is the MSA /θ/ phoneme (of the letter ث) which shifts in some words to /t/ and in others to /s/, e.g., ثلاثة *vlAvp* 'three' is pronounced as MSA /θala:θa/ or EGY /tala:ta/, and ثروة *vrwp* 'wealth, fortune' is pronounced as MSA /θarwa/ and EGY /sarwa/. The differences in the phonology affect how people write, especially given the absence of an orthographic standard for EGY. In our work, we use the conventional orthography for dialectal Arabic (CODA) proposed for EGY by [12], but we recognize common alternative spellings as well.

Table 1 shows some of the regular dialectal phonemic transformations from MSA to EGY:

Table 1. Phonological differences between EGY and MSA

MSA **letter**	EGY **letter**	MSA **example**	EGY **example**	ENG **gloss**
*****	d	*ahab	dahab	gold
v	t	valAvap	talAtap	three
q	>	qalob	>alob	heart

3.2 Morphological Variation

EGY morphology exhibits considerable divergence from MSA in both inflectional and derivational morphology. We note that the derivational differences are more relevant for building a lexical resource such as Tharwa; however we will review some of the inflectional variations. For an extensive discursive of Arabic morphology in NLP, see [11].

Affixation. EGY has some unique prefixes, suffixes and clitic morphemes that are not shared by MSA, e.g., the EGY future tense prefixes +هـ *ha+*[5] and +ح *Ha+* are notably different from the MSA future prefix +س *sa+*. For negation, EGY allows for clitic circumfixation with the prefix ع *mA* and the suffix ع *$* on verbal words, for example, the EGY verb ع *mlEb$*, 'he did not play', is contrasted against the MSA `lm ylEb`, where the negation marker ع *lm* does not affix onto the verb and there is no suffix. Furthermore, EGY, similar to most other DA, allows for an explicit progressive marker to be prefixed to imperfective verbs. The affix is a ع *b*. For example, the verb ع *bylEb*, 'he is playing' is contrasted with the MSA ع *ylEb*.

Case Inflection. While MSA has a complex case system, EGY does not. Different inflected forms in MSA map to the same form in EGY, e.g., MSA موظفون *mwZfwn*, 'employees [nom.]' and MSA موظفين *mwZfyn*, 'employees [acc./gen.]' map to EGY موظفين *mwZfyn*, 'employees'. The inflectional MSA morpheme marking nominative

[5] Arabic transliteration is in the Buckwalter scheme [13].

regular plural number ع *wn* as in ع *mwZfwn*, 'employees', is consistently replaced with the morpheme marking accusative/genitive plural number ع *yn* in EGY regardless of whether the case of the nominal is nominative or accusative/genitive.

Verbal Dual Inflection. Unlike MSA, where verbs are usually inflected for the dual when following dual nouns, This phenomenon, has to a great extent disappeared in EGY, and the verb plural inflection is used for both plural and dual nouns, e.g. the MSA `AlwaladAni >akalA` 'the two boys ate' becomes `Alwaladayn >akaluwA`.

Derivational Differences. MSA and EGY have similar word formation mechanisms, particularly because derivational morphology depends on roots and patterns. However, EGY has some morphological patterns which are not used in MSA such as `AisotaC1aC2C2aC3`, e.g. استخبّى *Aisotaxab~aY* 'to hide'. In addition EGY utilizes non-MSA morphological patterns to represent the passive voice or the unaccusative form of some verbs such as `AitoC1aC2aC3` (e.g. اتكتب *Aitokatab* 'to be written').

3.3 Lexical Variation

The EGY lexicon comprises entries that differ as well as overlap with MSA:

Identical. EGY and MSA words that are identical in all respects phonological, orthographic, morphological, and semantic, e.g. نشيط *na$iyT*, 'active'.

Semantic Cognates. EGY and MSA that share the same meaning but with some regular phonological and/or orthographic variation, e.g., EGY verb لعب *liEib* 'to play' corresponds to MSA verb لعب *laEib*.

Homographs/Homophones. EGY and MSA that have the same orthography and pronunciation but different meanings, e.g. حاجة *HAjap* is 'necessity' in MSA, but could mean both 'thing' as well as 'necessity' in EGY.

Distinct. Words that belong uniquely to only one of the varieties EGY or MSA, e.g. مش *mi$* 'not', بس *bas* 'only, enough', and دغري *dugoriy* 'straight-ahead' are only used in EGY.

On a sentiment level, we believe that the set of emotion expressive words in MSA will all be included in EGY but potentially with different usages and following the four classes listed above.

On the MWE level we should expect to see variation due to significant pragmatic use variation across the two variants of Arabic. For example an MWE such as كتب كتابه *ktb ktAbh* in EGY means 'got married' will correspond to عقد قرانه *Eqd qrAnh* as opposed to the non-MWE MSA homograph of كتب كتابه *ktb ktAbh* meaning 'wrote his book', though etymologically related.

4 Building Tharawat

We aim to merge the two resources Tharwa [7] and SANA [1] and link them through co-indexation to MuSTalAHAt [14] into a unified framework. We refer the reader to the actual detailed description of these three bodies of work. We provide a very brief description of each of them here:

Tharwa is a three-way EGY-MSA-ENG lemma based lexicon augmented with morphosyntactic and morphosemantic information pertaining to the EGY entry. A lemma is defined as a 3rd person singular masculine form for nominals and the perfective 3rd person form for verbal entries. The additional linguistic information is limited to part of speech (POS), gender, number, rationality, morphological pattern, and morphological root. The resource also includes closed class words and some named entities. Since there exist no standard orthography for DA in general, we use a standardized written form for EGY based on CODA [12] as the pivot for an entry in Tharwa, however we include as many orthographic variants for the EGY entry as possible. Moreover, the EGY and MSA entries are fully diacritized to reflect the phonology and morphology explicitly. The number of entries in the Tharwa dictionary is 73,348. Tharwa already links to the SAMA [9] and Call Home Egyptian Databases [15].

SANA is a large-scale multi-genre, multi-dialectal multi-lingual lexical resource for subjectivity and sentiment analysis of Arabic and its associated dialects. Language use varies across genres and SANA caters for that fact by encompassing lexica derived from four main genres: Online newswire, chat turns, Twitter tweets, and YouTube comments. In addition to MSA, where most NLP efforts have been focused for the past few years, SANA also covers EGY, and provides English glosses. A significant portion of SANA entries are listed in diacritized form with part of speech (POS) tags, gender, number, rationality, and genre class features. To date SANA comprises 227K unique entries.

MuSTalAHAt is a MWE lexicon for dialectal (Egyptian) Arabic that covers, among other types, expressions that are traditionally classified as idioms, prepositional verbs, compound nouns, and collocations. The annotation scheme covers the following areas: Phonological and orthographic information; POS tag, based on the observation of how an MWE functions as a whole lexical unit; Syntactic variability and structural composition; Lexicographic types, which includes the classifications followed in the dictionary-writing domain (idioms, support verbs, compound nouns, etc.); Semantic information, which cover semantic fields and relations; Idiomaticity Degree where three levels of rating is adopted to quantify opaqueness levels; Degree of morphological, lexical and syntactic flexibility; Pragmatic information, which includes adding usage labels to MWEs where applicable; Translation, which includes the MSA and English equivalents, either as an MWE in MSA and English if available or as a paraphrase otherwise. To date, MuSTalAHAt comprises 8590 entries for EGY.

We should note that the idea is to augment MuSTalAHAt in a manner similar to Tharawat. Therefore we link entries from Tharawat to entries in MuSTalAHAt, and vice versa, as well as we augment MuSTalAHAt with information as in Tharawat.

4.1 Infrastructure

We view Tharawat being built as a relational database a la current Tharwa framework, with a back end extensive Oracle database, and two front end interfaces: a) search and visualization of results of searching the database and retrieving information from it; and b) for manually augmenting Tharawat entries by annotators. A description of the latter (b) component is provided in [7]. For the Search and Visualization interface it should adhere to some core desiderata:

- We would like Tharawat to be searchable using various input forms for a word/MWE: lemma form/tokenized form/surface form in DA (specified which one from a predetermined set of DA covered in Tharawat)/MSA/EN. In this case there will be a link to a morphological analyzer such as CALIMA [10] to ensure that the surface form or tokenized form of the entered unit can be resolved to the underlying lemma form which is the manner in which entries will be specified in Tharawat;
- The search entity maybe entered in different script encodings (Arabic or Latin), the input if DA maybe entered it in non standard orthography using Arabic script, Arabizi [3], as well as CODA [12]. Arabizi is a form of Arabic written in romanized script also known as Arabish where people use digits intertwined with letters to express words in the dialect. It is a very common way of writing Arabic in social media in particular. A system has been developed for handling Arabizi input as in [3]. CODA is a system devised in Arabic script to conventionalize DA writing. CODA guidelines exist now for Egyptian, Iraqi, Tunisian;
- A user may search by any of the fields in the database such as by POS, or by MWE, or semantic information such as request all the rational entries in the resource, or all the feminine entries, or those pertaining to a specific topic;
- In terms of visualization, we would like for the user to be able to see data in the form of tables, chart statistics, with representative examples.

4.2 Augmentation, Validation and Verification

The initial population will be from the three core resources but then we will be augmenting Tharawat as follows.

Crowd Sourcing. We will harvest MSA and ENG equivalents for existing EGY entries leveraging the power of crowd sourcing. In a spirit similar to that undertaken in Tharwa, we will exploit crowd sourcing for both verification and augmentation. In the verification phase we have *rating* experiments where the annotators are asked to indicate whether a triple EGY-MSA-ENG is correct or not, i.e., a binary decision. We also have *generating* experiments where we provide the annotators with two of the fields and ask them to provide the third. Hence we present the crowd with the EGY and the MSA and ask them to provide the ENG, or the EGY and ENG and ask them to provide the MSA. It is worth noting that we provide the Arabic, EGY and MSA, fully diacritized. In the case of EGY, we exhaustively provide all the orthographic variants we have in Tharwa, not only the CODA form. We submit these variants as separate instances for

the purposes of annotation. In one mode of the experiments we will also provide them with example sentence usages as derived from corpora.

In the augmentation step, we provide the annotators with the MSA and ENG equivalents and ask them to provide the EGY equivalent. We apply the same process to MWEs.

Automatic Augmentation. We will exploit parallel and comparable corpora that exist for EGY-ENG and MSA-ENG in the process of verifying and augmenting the manual process of Tharawat creation. We derive word level correspondents via automatic word-level alignment applied on lemmatized parallel corpora. We also derive MWE correspondents in the same manner after running MWE detection on the parallel corpora. The augmented MWE will be further added to MuSTalAHAt and linked into Tharawat. This approach in principle is similar to that taken by [17] for learning lemma-based dictionaries from parallel data, however we triangulate two sets of parallel/comparable corpora simultaneously.

This process generates many candidates. The resulting candidates are subjected to manual verification and validation via crowd sourcing as well as by lab annotators. Using the triangulation methods we will link the entries in Tharawat to sentences in the corpora generating examples in all the language entries for DA, MSA and EN. The result will be a corpus that is at least a single lemma in the respective language. We will opt for getting as many lemmas disambiguated per sentence hence opting for more redundancy in the example usages.

4.3 Links to Other Dialects

We have already augmented some of the underlying resources such as Tharwa to Iraqi and Levantine leveraging existing paper dictionaries and manual translation. By extension these links will be reflected in Tharawat. We use paper dictionaries for Iraqi that exist for Iraqi English correspondents with POS tags for the English and Iraqi, therefore we link them automatically pivoting on English and its POS information to link to MSA. Levantine, we provided the MSA and ENG equivalents to translators and they were translated into levantine lemmas that were fully diacritized. We will leverage the existing techniques of automatic augmentation and crowd sourcing for both enriching and verifying the extension to other dialects. We will apply the same techniques of crowd sourcing, and automatic augmentation to the MuSTalAHAt resource.

4.4 Links to English Lexical Resources

In our effort to link Tharawat entries to English lexical resources, via the automatic augmentation process we will also run some of the best word sense disambiguation, such as DKpro,[6] and semantic role labeling systems on the English side of the parallel/comparable corpora so automatically inducing wordNet[7] sense labels for the the

[6] `https://code.google.com/p/dkpro-core-asl/`
[7] `http://wordnet.princeton.edu/`

English equivalents as well argument structure frames as listed in FrameNet[8] and Propbank,[9] thereby linking Tharawat entries to the Unified Verb Index.

4.5 Content Desiderata

We expect Tharawat to encompass all the following information:

- **Unique Entry ID**: Each entry in the database is uniquely identified with a unique id;

- **CODA DA Unit**: This could be a lemma or a MWE in CODA, this the diacritized conventional orthography lemma form of the EGY entries [12]. All entries are fully diacritized with short vowels. A lemma is defined as the third person masculine (or feminine if only interpretation) singular (or plural for broken plurals) for nominals, and perfective 3rd person masculine singular for verbs. For illustration, compare the EGY noun lemma عين *Eayn* 'eye' to its broken (irregular) plural inflected form عيون *Euyuwn* 'eyes' (which is also linked to its lemma). Both words will be entries in Tharawat. The MWE's will technically be in MuSTalAHAt.

- **Unit Coindex**: This field links lemmas to one another as in the case of broken plurals to their singular forms and also linking lemmas participating in MWEs to the full MWE entries;

- **Unit DA Variants**: This will list orthographic variants for the DA units given the fact that the DAs have no standard orthography; This field lists alternative naturally occurring orthographic variants of the EGY CODA entries as obtained from their original sources. This field can have multiple variants both diacritized and undiacritized, e.g., EGY entry كثير *kiviyr* 'many, a lot' (pronounced /kitiyr/) has the variant كتير *kitiyr*;

- **MSA Equivalent**: This field will provide the MSA equivalent(s) to the DA unit;

- **ENG Equivalent**: This field will provide the English equivalent(s) to the DA unit;

- **POS**: This would be spelled out for each unit in the respective DAs/MSA/EN. These will be taken from and augmented from the core underlying resources;

- **Lemma Semantic Information**: This will include information such as functional number, functional gender, rationality where applicable to lemmas. The inflected entry عيون *Euyuwn* 'eyes' is marked as feminine, plural and irrational. We follow the conventions for marking these attributes proposed by [4].This information is provided for each of the languages in the resource, namely DAs/MSA/EN;

[8] https://framenet.icsi.berkeley.edu/fndrupal/
[9] http://verbs.colorado.edu/propbank/

- **Morphological Pattern and Root Information**: Morphological pattern and root information where applicable, e.g., the EGY verb lemma استبدل *Aisotabodil* 'to change' has the root *bdl* and pattern `AisotaC1oC2iC3`. We will provide that for both DA and MSA entries;

- **Link to English external resources**: This provides links to English WordNet for English equivalents, English Unified Verb Index entries, Concept Net, etc.;

- **Link to MSA external resources**: This provides links to entries for units in MSA dictionary resources where available such as Arabic wordNet;

- **Example sentence usages**: This will be provided for each of the languages as derived from comparable and parallel corpora;

- **Pragmatic Information**
  1. **Sentiment Type**: This reflects whether the unit has a sentiment value and if so, what is it (similar to the affect or emotion, such as happy, joy, angry, etc.);

  2. **Sentiment intensity**: If unit has a Sentiment Type, then inherent sentiment intensity would be annotated;

  3. **Sentiment usage example in (DAs/MSA/EN)**: If unit has a Sentiment Type and Intensity, then an example usage in each of the languages used in Tharawat where applicable. The assumption is that such information as sentiment intensity and type will hold cross linguistically;

  4. **Level of formality** This lists the level of formality of where a unit is most frequently observed;

  5. **Genre** This lists the genre such as speech, broadcast news, chat, SMS, discussion fora, etc;

  6. **Topic** This lists the possible domains where a unit entry is used such as Economics, Sports, politics, etc.

- **Provenance Information**: Tharawat would list frequency information for the unit as observed in corpora and online, as well as the basic source of the entries: crowd sourced, automatically generated, or from paper dictionaries, etc.

4.6 Quality Control

A large portion of Tharawat will compiled and revised manually by professional linguists. However, it is necessary to make sure that errors are minimized and data backups are regularly maintained. Therefore, to guarantee the quality of Tharawat we employ two types of automatic quality control checks that help annotators minimize errors and data loss.

Version-Control. Tharawat will undergo constant version-control using SVN to backup new versions and to retrieve old versions where needed. We will adopt the developed interface for Tharwa designated between linguists/developers and the SVN tool for checking in updates to the SVN and checking out the latest version. Currently, the tool checks the new version before accepting. For example, if an annotator is assigned specific fields to revise such as POS and rationality, then they are only allowed to modify those specific attributes. If a violation occurs and the annotator modifies other attributes without checking it out officially, the SVN tool rejects the modification and produces a detailed report. This is particularly useful for preventing any unintended changes, and avoiding version conflicts. It is worth noting that there are significant dependencies among the attributes leading to typical multiple attribute check outs simultaneously.

Automatic Consistency Checks. Regarding the EGY and MSA data content, several automatic checks for detecting errors related to improbable spelling or diacritization were developed. We will extend these automatic checks to cover for other dialects. We also implement automatic checks on the ENG data ensuring that proper nouns[10] are capitalized and no spelling errors exist in the ENG data.

Acknowledgments. We would like to thank authors of the Tharwa, SANA and MuSTalAHAt research works as well as the numerous annotators whose feedback helped shape a lot of our vision.

References

1. Abdul-Mageed, M., Diab, M.: Sana: A large scale multi-genre, multi-dialect lexicon for arabic subjectivity and sentiment analysis. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014). European Language Resources Association (ELRA), Reykjavik (2014), http://www.lrec- conf.org/proceedings/lrec2014/pdf/919_Paper.pdf
2. Abo Bakr, H., Shaalan, K., Ziedan, I.: A Hybrid Approach for Converting Written Egyptian Colloquial Dialect into Diacritized Arabic. In: The 6th International Conference on Informatics and Systems, INFOS 2008, Cairo University (2008), http://sites.google.com/site/khaledshaalan/publications/conference-papers/AHybridApproachforConvertingWrittenEgyptian.pdf?attredirects=0
3. Al-Badrashiny, M., Eskander, R., Habash, N., Rambow, O.: Automatic transliteration of romanized dialectal arabic. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning, pp. 30–38. Association for Computational Linguistics, Ann Arbor (2014), http://www.aclweb.org/anthology/W14-1604
4. Alkuhlani, S., Habash, N.: A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011), Portland, Oregon, USA (2011)
5. Badawi, E.S., Hinds, M.: A Dictionary of Egyptian Arabic. Librairie du Liban (1986)
6. Brustad, K.: The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects. Georgetown University Press (2000)

[10] We assume that if an EGY is a proper noun, then its corresponding ENG is also a proper noun.

7. Diab, M., AlBadrashiny, M., Aminian, M., Attia, M., Elfardy, H., Habash, N., Hawwari, A., Salloum, W., Dasigi, P., Eskander, R.: Tharwa: A large scale dialectal arabic - standard arabic - english lexicon. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), pp. 3782–3789. European Language Resources Association (ELRA), Reykjavik (2014), http://www.lrec-conf.org/proceedings/lrec2014/pdf/1161_Paper.pdf, aCL Anthology Identifier: L14-1115
8. Ferguson, C.F.: Diglossia. Word 15(2), 325–340 (1959)
9. Graff, D., Maamouri, M., Bouziri, B., Krouna, S., Kulick, S., Buckwalter, T.: Standard Arabic Morphological Analyzer (SAMA) Version 3.1 (2009), linguistic Data Consortium LDC2009E73
10. Habash, N., Eskander, R., Hawwari, A.: A Morphological Analyzer for Egyptian Arabic. In: NAACL-HLT 2012 Workshop on Computational Morphology and Phonology (SIGMORPHON 2012), pp. 1–9 (2012)
11. Habash, N.: Introduction to Arabic Natural Language Processing. Morgan & Claypool Publishers (2010)
12. Habash, N., Diab, M., Rabmow, O.: Conventional Orthography for Dialectal Arabic. In: Proceedings of the Language Resources and Evaluation Conference (LREC), Istanbul (2012)
13. Habash, N., Soudi, A., Buckwalter, T.: On Arabic transliteration. In: Soudi, A., Neumann, G., van den Bosch, A. (eds.) Arabic Computational Morphology, Text, Speech and Language Technology, vol. 38, ch. 2, pp. 15–22. Springer (2007), http://dx.doi.org/10.1007/978-1-4020-6046-5_2
14. Hawwari, A., Attia, M., Diab, M.: A framework for the classification and annotation of multiword expressions in dialectal arabic. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), pp. 48–56. Association for Computational Linguistics, Doha (2014), http://www.aclweb.org/anthology/W14-3606
15. Kilany, H., Gadalla, H., Arram, H., Yacoub, A., El-Habashi, A., McLemore, C.: Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22 (2002)
16. Maamouri, M., Bies, A., Buckwalter, T., Diab, M., Habash, N., Rambow, O., Tabessi, D.: Developing and using a pilot dialectal Arabic treebank. In: LREC, Genoa, Italy (2006)
17. Saleh, I., Habash, N.: Automatic extraction of lemma-based bilingual dictionaries for morphologically rich languages. In: Third Workshop on Computational Approaches to Arabic Script-based Languages at the MT Summit XII, Ottawa, Canada (2009)
18. Salloum, W., Habash, N.: 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, pp. 10–21 (2011)
19. Spiro, S.: An Arabic-English Vocabulary of the Colloquial Arabic of, Egypt. Al-Mokattam printing office (1895)
20. Spiro, S.: Arabic-English Dictionary of the Colloquial Arabic of Egypt. Librairie Du Liban (1987)

High Quality Arabic Lexical Ontology Based on MUHIT, WordNet, SUMO and DBpedia

Eslam Kamal[1], Mohsen Rashwan[2], and Sameh Alansary[3]

[1] Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
eslamkamal@hotmail.com
[2] The Engineering Company for the Development of Computer Systems, RDI, Cairo, Egypt
mrashwan@rdi-eg.com
[3] Bibliotheca Alexandrina, Alexandria, Egypt
Sameh.alansary@bibalex.org

Abstract. In this paper, we aim to move ontology-based Arabic NLP forward by experimenting with the generation of a comprehensive Arabic lexical ontology using multiple language resources. We recommend a combination of MUHIT, WordNet and SUMO and use a simple method to link them, which results in the generation of an Arabic-lexicalized version of the SUMO ontology. Then, we evaluate the generated ontology, and propose a method for increasing its named entity coverage using DBpedia, English-to-Arabic Transliteration, and Named Entity Recognition. We end up with an Arabic lexical ontology that has 228K Arabic synsets, linked to 7.8K concepts and 143K instances. This ontology achieves a precision of 96.9% and recall of 75.5% for NLU scenarios.

Keywords: Arabic NLP, Ontology-based Arabic NLP, Arabic Language Resources, Arabic Lexical Ontology, Arabic WordNet, Arabic SUMO, Arabic Ontology, Multilingual Dictionary, MUHIT, Arabic Named Entities.

1 Introduction

Natural Language Processing (NLP) is a field of computer science, artificial intelligence and linguistics, concerned with Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks.

An Ontology is defined as an explicit specification of a conceptualization [1]. It describes the concepts and instances of some area of interest along with the relations between them. An Upper Ontology describes generic, top-level concepts that have the same meaning across all domains (e.g. Human, Artifact and Organization concepts). A Domain Ontology describes concepts of a particular domain, so for example, some of the concepts in a computer-domain ontology would be Mouse, CPU, and Antivirus. A Lexical Ontology provide rich lexical information on the words of a given language, in addition to the concepts, instances, and relations.

Ontology-based NLP is a branch of NLP where ontologies are used to support NLU and/or NLG. This branch of NLP started early on 1996 when Nirenburg et al. presented an ontology that was used in the Mikrokosmos machine translation project

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 98–111, 2015.
DOI: 10.1007/978-3-319-18111-0_8

[2]. This ontology was combined with English and Spanish lexicons, along with a two-way mapping between each lexicon and the ontology. This mapping was used to understand the source text (NLU), and to generate the target language text (NLG). Then on 2001, the same group presented Ontological Semantics as a new NLP approach which uses an ontology as the central resource for extracting and representing meaning of Natural Language (NL) texts, reasoning about knowledge derived from texts, as well as generating NL texts based on the representations of their meaning [3].

Since that time, ontology-based NLP has been applied in many applications (such as information retrieval [4], semantic similarity [5], and word sense disambiguation [6]), and became the state of the art in many of them. With the rise of the Semantic Web and ontology standards, such as the Web Ontology Language (OWL), many ontologies are now available. However, these ontologies are usually annotated with few linguistic information, which limits the usefulness of these ontologies to NLP [7].

Ontology-based NLP requires the ontologies to be annotated with at least NL words, which is generally available for English and few of the top-served languages. When it comes to Arabic, there is a lack of comprehensive ontologies that are annotated with Arabic words or linked to machine readable Arabic lexicons.

Under the assumption that ontology concepts and the relations between them are language-independent (although they are culture-dependent), high-quality linking between the concepts of a comprehensive ontology and the senses of an Arabic lexicon should be enough to use this ontology in Arabic NLP applications.

In this paper, we study the existence of such linking, through manual evaluation of a variety of the current language resources. Section 2 discusses the related work, Section 3 introduces a set of potential language resources. Sections from 4 to 7 show the details of our work. Finally, the conclusion and future work in section 8.

2 Related Work

Several attempts have been made to link words of different languages to English ontologies and WordNet [8] as a lexical ontology. For English, this linking may not be required because most of the ontologies are already annotated (lexicalized) with English words or linked to an English lexicon (usually WordNet).

One of the earliest attempts was on 1994 when Akitoshi Okumura et al. [9] proposed a semi-automatic method for associating a Japanese lexicon with an ontology using a Japanese-English bilingual dictionary. They divided the association of ontology concepts with bilingual concepts into four cases: single to single association, single to multiple associations, multiple to single association, and multiple to multiple associations. Then, they processed the different cases using three algorithms; the equivalent-word match, the argument match, and the example match.

Latifur Khan et al. [10] associated a machine-readable Arabic-English lexicon with the nouns of WordNet. Verbs and other parts of speech were not covered. The matching was based on word and definition translations, and a generalization step that replaces a set of matched WordNet concepts (synsets) by their common parent concept. Their algorithm yielded 69% precision on a sample of 200 Arabic lexicon entries.

Benfeng Chen et al. [11] associated Chinese words to FrameNet using a Chinese ontology called "HowNet" that included many information such as part of speech (POS) tags, definitions, semantic relations, and semantic roles. This work is not applicable to Arabic due to the lack of such rich language resource at the best of our knowledge.

Véronique Malaisé et al. [12] anchored words from the Dutch Cultural Heritage Thesauri to WordNet based on POS tags and the lexical description of the terms without considering the words themselves.

Javier Farreres et al. [13] presented the LeOnI methodology which uses a logistic regression model to combine mappings proposed by a set of 17 classifiers that were adopted from previous researches. The method was evaluated on linking Spanish and Thai words to WordNet. The best result for Spanish was a recall of 72% and precision of 91%. For Thai, the recall was 80% and the precision was 76%.

Many language resources are strongly related to our work, including the Arabic WordNet [14], but we preferred to limit this section to the automated and semi-automated work, and leave the details of the language resources to the next section.

3 Language Resources

All of the work presented here depend on some bilingual language resource to achieve the linking with an ontology. The language resources used range from a simple dictionary with word translations and definitions to a full-fledged lexical ontology. In this section, we will go through the different English and Arabic language resources that we considered as potential candidates for combining an Arabic lexical ontology.

3.1 Bilingual Dictionaries

We have considered five online bidirectional Arabic-English dictionaries for our work, namely WordReference [15], Arabdict [16], Almaany [17], Babylon [18] and Google [19]. Table 1 shows a **Y**es/**N**o/**P**artial metadata-based comparison between them, with the following column meanings:

- DEF: Word sense definitions[1]
- EXA: Example phrases or sentences for word senses
- DOM: Domain information for the word sense
- DIA: Arabic diacritics on the word forms
- POS: Part of speech tags (noun, verb, adjective … etc.)
- RNK: Ranked (or sorted on a rank) senses
- SPW: Average number of senses per word[2]

[1] Maybe missing from some senses even if marked Y. The same applies to all columns.

[2] Measured using a unified sample of 1000 frequent nouns on the Arabic-English direction.

Table 1. Metadata-based comparison between 5 online dictionaries

Dictionary	DEF	EXA	DOM	DIA	POS	RNK	SPW
WordReference	N	Y	N	N	Y	Y	4.9
Arabdict	N	N	Y	P	Y	N	9.0
Almaany	Y	N	Y	Y	Y	N	3.9
Babylon	N	N	N	N	Y	N	1.1
Google	N	N	N	N	Y	N	2.7

3.2 WordNet

WordNet [8] is a large lexical database for English by Princeton. It contains information about 147,278 words divided into nouns, verbs, adjectives, and adverbs. The words are then further divided into 206,941 senses, with an average of 1.4 senses per word. Senses are grouped by synonymy into 117,659 unordered sets called synsets. Words in the same synset denote the same concept and are interchangeable in many contexts.

There are also some semantic relations among the synsets including the hypernym and hyponym relations. Thus, WordNet is sometimes interpreted as a lexical ontology. We prefer to deal with WordNet as a lexical database rather than an ontology, because the vast majority of WordNet's relations connect synsets from the same POS. Thus, WordNet is a set of four disconnected sub-nets, rather than one connected ontology.

WordNet has achieved great success and became the dominant English lexicon in NLP applications. Driven by this success, it has been linked to many well-known lexical resources, few of them are BabelNet [20], SUMO [21], OpenCyc [22], DOLCE [23] and DBpedia [24]. WordNet has been also partially translated to many other languages including Arabic [14], or connected to wordnets of other languages.

Table 2. Comparison between Princeton WordNet and Arabic WordNet

	Synsets	Words	Senses	SPW
WordNet (WN)	117,659	147,278	206,941	1.4
Arabic WordNet (AWN)	11,269	13,808	23,481	1.7
AWN / WN Ratio	**9.6%**	**9.4%**	**11.3%**	

3.3 Arabic WordNet

The Arabic WordNet (AWN) is a wordnet for Modern Standard Arabic (MSA) based on the design and contents of WordNet (WN) [14]. Thus, the AWN synsets are directly connected to WN synsets. At the time of this writing, AWN consists of 13,808 non-diacritized Arabic words divided into 23,481 senses that form 11,269 Arabic synsets. All of the synsets are linked to the equivalent English synsets in WordNet.

Table 2 shows a quick comparison between WordNet and AWN in terms of words, senses, synsets, and senses per word. The low AWN/WN ratios suggest low coverage

of Arabic words in AWN, which can be easily verified as some of the commonly used Arabic words are missing, such as the noun 'بطولة' (championship), the verb 'تقابل' (meet), and the adjective 'أفريقي' (African).

3.4 MUHIT

MUHIT (MUltilingual Harmonized dIcTionary) is a multilingual lexical database produced by the UNDL Foundation within the UNL framework [25]. Entries from different languages are interlinked by sense, and word forms are associated to a uniform concept identifier. It contains more than 10 million word forms for 40+ languages. The Arabic share is more than 2 million word forms (about 20% of the whole size).

More importantly, more than 150,000 of MUHIT Arabic senses are linked to about 117,000 WordNet synsets. It also provides a set of linguistic information that makes it useful for morphological, lexical and syntactic Arabic NLP tasks as below:

- Morphological Information
  - Part of speech (POS): noun, verb, adjective, adverb … etc.
  - Lexical structure: sub word, simple word or multiword expression
  - Inflectional paradigm: 343 paradigms representing 10,566 morphological rules
- Morpho-syntactic Information
  - Transitivity: behavior of a verb and the type of its arguments
  - Tense: past, present and future tenses
  - Gender: masculine, feminine, common or variable
  - Number: singular, plural or invariant. Dual in Arabic is a subclass of plural.
  - Person: first person, second person and third person
- Syntactic Information
  - Valence: the number of syntactic arguments required by any predicate
  - Aspect (for verbs): the temporal internal structure of an action, event or state
  - Subcategorization Frame: number and types of the necessary syntactic arguments
- Semantic Information
  - Semantic classification: inherited from WordNet synsets
  - Animacy (for nominal concepts): human or animal

Table 3. WordNet linking in MUHIT

	Synsets	Unique Words	Senses	SPW
WordNet Seed (WN)	114,473	144,743	202,436	1.4
Linked in MUHIT (MH)	114,135	104,934	147,254	1.4
MH / WN Ratio	**99.7%**	**72.5%**	**72.7%**	

As long as MUHIT database is not publicly available, we were permitted to retrieve the set of MUHIT senses that are linked to a comprehensive seed of 115,247 WordNet synsets (the same seed is referred to later on below). Table 3 shows the actual numbers, which indicate a very high WordNet linking coverage in MUHIT. A sample of this data is shown in Section 5, Figure 2 (List 2).

On the other hand, for almost the same number of synsets, numbers of words and senses are much less than those of WordNet. The numbers may not be comparable because they represent different languages, or may indicate low Arabic word coverage in the range that is linked to WordNet. We will come to this later in the next sections.

In terms of linking precision, we have sampled a random set of 500 equivalent synsets and labeled them manually for correct linking. We got 492 correct links, which translates to 98.4% precision. Examples of the few errors are linking of 'semi-public' to 'شبيه' (similar), and 'unaffected' to 'متأثر' (affected).

3.5 SUMO

The Suggested Upper Merged Ontology (SUMO) is a free formal upper ontology owned by the IEEE [26]. At the time of this writing, it consists of the SUMO itself (Merge), the Mid-Level Ontology (MILO), and other 43 domain ontologies. When combined together, SUMO has 27,684 terms divided into 7,772 concepts and 19,912 instances, along with ~80,000 axioms.

One unique characteristic of SUMO is that its terms are linked to most of WordNet synsets. This mapping [21] is manifested into three main link types as follows:

- The '=' link means that the WordNet synset is equivalent in meaning to the SUMO concept (e.g. 'person' in WordNet is equivalent to 'Human' in SUMO).
- The '+' link means that the WordNet synset is subsumed by the SUMO concept (e.g. 'window' in WordNet is subsumed by 'Hole' in SUMO).
- The '@' link means that the WordNet synset is an instance of the SUMO concept (e.g. 'Cambridge' in WordNet is an instance of 'City' in SUMO).

Each WordNet synset has a single link to SUMO. Table 4 shows the count and distribution of the different link types. The Initial Count column shows the counts of each link type as in SUMO 'WordNetMappings' files. However, some of these links are to a special SUMO concept called 'SubjectiveAssessmentAttribute' which means that the WordNet synset lacks objective criteria for its attribution. These links are not useful, so they are subtracted in the Useful Count column. The final overall coverage is 89% of the synsets of WordNet 3.0 at the time of this writing.

Table 4. WordNet to SUMO mapping types

Link Type	=	+	@	**Total**
Initial Count	4,802	99,081	10,593	114,476
SubjectiveAssessmentAttribute	113	10,129	9	10,251
Useful Count	4,689	88,952	10,584	104,225
WN Synset Coverage	4.0%	75.6%	9.0%	88.6%

The '+' mapping is naturally the most frequent, because SUMO concepts are usually more general than those of WordNet. The '+' mapping is not as useful as the '=' mapping, however it should be sufficient for most of the NLU tasks. The '=' mapping is bidirectional by nature, so it has an extra benefit for NLG tasks.

3.6 OpenCyc

OpenCyc [27] is a large free general knowledge base and commonsense reasoning engine. At the time of this writing, it includes ~239,000 terms divided into ~135,000 concepts and ~104,000 instances, with fair amount of linking to other resources including DBpedia (~48,000 links), UMBEL (~21,000 links) and WordNet 2.0 (~11,000 links). The OpenCyc terms are also lexicalized by English words or expressions using the 'prettyString' predicate (e.g. 'people', 'human' and 'individual' are all values of the prettyString predicate for the 'Person' concept).

WordNet links in OpenCyc are equivalent (and hence, bidirectional) as opposed to three link types in SUMO. Table 5 shows a simple metadata-based comparison between SUMO and OpenCyc in terms of concepts, instances and WordNet links.

Table 5. Comparison between SUMO and OpenCyc

	Concepts	Instances	Equivalent Links	Subsumed Links	Instance Links	Total Links
SUMO	7,772	19,912	4,689	88,952	10,584	104,225
OpenCyc	135,243	104,021	11,107	0	0	11,107

4 Feasibility Study

Considering the NL resources above, we have several options for the generation of the Arabic lexical ontology. Our target is to achieve the best quality and coverage for both NLU and NLG scenarios. Figure 1 summarizes our options; blocks are the NL resources, arrows are the links between them, solid arrows are existing links, dashed arrows are options for automation, and the letters of H, M and U stand for high, medium and unknown respectively, and represent the anticipated degree of quality.

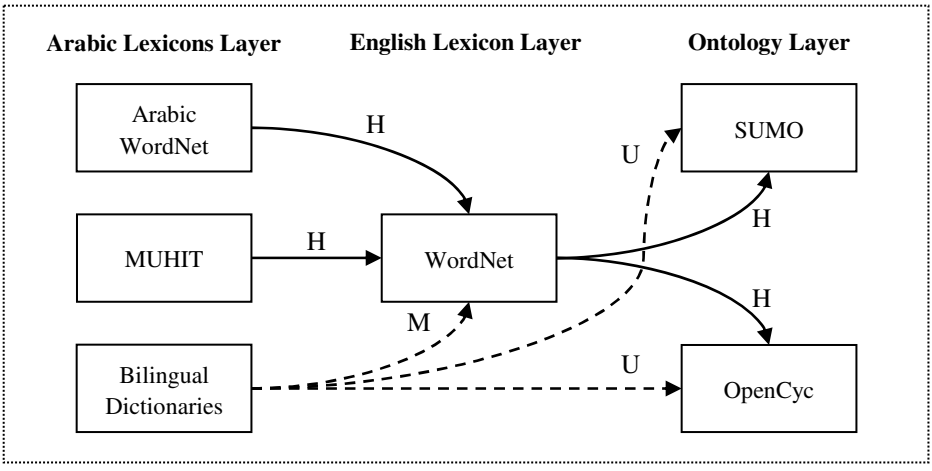


Fig. 1. Options for the generation of an Arabic lexical ontology

The state of the art automated linking was achieved by the LeOnI methodology [13] between Spanish and Thai dictionaries and WordNet. The average precision and recall over the two languages were less than 85% and 80% respectively, which is denoted here by M (medium). The performance of the same methodology on linking between Arabic and SUMO or OpenCyc would not be better, denoted by U (Unknown). With the existence of high-quality links between Arabic lexicons (AWN and MUHIT) and WordNet, and between WordNet and credible ontologies (SUMO and OpenCyc), we will not further consider M and U links, and hence skip the bilingual dictionaries.

In order to assess the H links further, and reduce the amount of manual labelling, we have assumed (temporarily) a 100% precision for H links to realize the recall full potential before moving forward. The count of WordNet synsets is a common divisor for recall estimation. Under this assumption, we computed the recall (equal to coverage) of four possible options for linking from the Arabic lexicons layer to the ontology layer (serving NLU), and other four options for the opposite direction (serving NLG). The results of this computation are shown in Table 6.

Among the eight options we proposed, we have promising recall for only one option from MUHIT to SUMO (even under the unrealistic assumption of 100% precise links). On the other direction, unfortunately, we are left with no promising options. Equivalent links from SUMO to WordNet cannot be comprehensive since the total number of SUMO concepts is 27,684 which puts an upper bound of 23.5% on the recall.

Automatic linking from OpenCyc to WordNet is a viable option because the OpenCyc ontology is large enough that it can contain most of the WordNet synsets in an unlinked state. We would like to consider this option as a future work.

5 Ontology Generation

Based on this feasibility study, we target the creation of an Arabic-lexicalized version of SUMO based on the linking between MUHIT and WordNet, and between WordNet and SUMO. The linking between WordNet and SUMO is publicly provided by SUMO as a set of four files (one for each WordNet POS). The total number of linked WordNet synsets is 114,473. A sample of these links is shown in Figure 2, List 1.

MUHIT has the links to WordNet, but they are not provided publicly. We were allowed to get a list of MUHIT senses that are linked to an input set of WordNet synset IDs. We have provided the list of synset IDs that have links to SUMO, and got a list of 151,123 Arabic senses. A sample of these links is shown in Figure 2, List 2.

The steps we taken to generate the Arabic lexical ontology are as follows:

1. Normalized List 1 by prepending 1, 2, 3 and 4 to the IDs of n, v, s and r POS codes
2. Generated List 2 by retrieving all MUHIT words that correspond to List 1
3. Grouped all List 2 words with the same Synset ID into one Synset entry
4. Inherited the Concept and Link Type from List 1 to List 2 on the same Synset ID
5. Grouped all List 2 words having the same string into one Word entry
6. Assigned a sequential Sense ID to each occurrence of a Word in a different Synset
7. Inherited all of the Concepts, Instances and Axioms from SUMO

Table 6. Bidirectional Arabic lexicon - Ontology potential recall

Option	Phase I Links [3]	Phase I Recall	Phase II Links [4]	Phase II Recall	Expected Recall [5]
	Direction 1: Arabic Lexicon to Ontology				
AWN to SUMO	11,269	9.6%	104,225	88.6%	8.5%
AWN to OpenCyc	11,269	9.6%	11,107	9.4%	0.9%
MUHIT to SUMO	114,135	99.7%	104,225	88.6%	88.3%
MUHIT to OpenCyc	114,135	99.7%	11,107	9.4%	9.4%
	Direction 2: Ontology to Arabic Lexicon				
SUMO to AWN	4,689	4.0%	11,269	9.6%	0.4%
SUMO to MUHIT	4,689	4.0%	114,135	99.7%	4.0%
OpenCyc to AWN	11,107	9.4%	11,269	9.6%	0.9%
OpenCyc to MUHIT	11,107	9.4%	114,135	99.7%	9.4%

As a result, we got an Arabic-lexicalized version of SUMO, where links to MUHIT synsets are appended to the existing WordNet links. Most of MUHIT synsets (as in Table 3) are connected to SUMO concepts and instances (as in Table 6).

List 1: WordNet 3.0 to SUMO Links

04587648	n	window	Window=
15299783	n	window	TimeDuration+
08933621	n	Orly	City@
00730758	v	draw	Reasoning+
00987071	v	draw	describes=
01582645	v	draw	Drawing+
02698145	s	classical	ArtWork+
02966829	s	Brazilian	Nation@
00110533	r	inside	Indoors+

List 2: MUHIT to WordNet 3.0 Links

نافذة	104587648	LEX=N
فترة زمنية	115299783	LEX=N
ضاحية أورلي	108933621	LEX=N
استخلص	200730758	LEX=V
وصف	200987071	LEX=V
تتبع	201582645	LEX=V
كلاسيكي	302698145	LEX=J
برازيلي	302966829	LEX=J
في الداخل	400110533	LEX=A

Fig. 2. Samples of MUHIT, WordNet 3.0 and SUMO links

6 Evaluation

We have already evaluated the precision of the links from MUHIT to WordNet as 98.4%. Now, we evaluate the end-to-end links between MUHIT and SUMO. Given that we have three link types, we have sampled 200 random links for each type and labeled them manually for correctness. We have assigned a special score of 0.5 for the wrong '=' links that would be correct as '+'. The results are shown in Table 7, where the overall precision P is calculated as in Equation 1.

$$P\ (overall) = \frac{Count(=) \times P(=) + Count(+) \times P(+) + Count(@) \times P(@)}{Count(=) + Count(+) + Count(@)} \quad (1)$$

[3] From Arabic Lexicon to WordNet in the 1st half / From Ontology to WordNet in the 2nd half.
[4] From WordNet to Ontology in the 1st half / From WordNet to Arabic Lexicon in the 2nd half.
[5] Computed as Phase I Recall × Phase II Recall.

Table 7. MUHIT to SUMO links precision evaluation

Link Type	Link Count	Sample Count	MUIHT → WN Correct	MUHIT → SUMO Correct	Precision
=	4,689	200	198.5	190	95.0%
+	88,952	200	196	192	96.0%
@	10,584	200	200	197	98.5%
Overall	104,225	weighted average of 3 precision numbers			**96.2%**

In terms of recall, we want to measure the amount of MSA words correctly covered by a term in our ontology. Each correct word must pass the following conditions:

1. The word is found in MUHIT
2. The word sense (in context) is found in MUHIT senses for the word
3. MUHIT sense belongs to a synset that is linked to a SUMO term
4. The link between the synset in MUHIT and the term in SUMO is correct

For the purpose of simplifying the labeling process, and also to keep track of each source of low coverage separately, we divided the evaluation into separate tasks that correspond to conditions (1) and (2) above. Conditions (3) and (4) correspond to the linking coverage and precision respectively. Linking coverage can be computed from Table 6 directly as 91.3% (104,225 out of 114,135), and the precision is 96.2%. For (1) and (2), we need to sample some Arabic text. We were allowed to get a sample of a modern Arabic news corpus that was used for building the Microsoft Research Arabic Toolkit Service (ATKS) [30]. The sample size was 3 million words.

Then, for (1), we sampled a random set of 500 words and counted manually how many of them are covered. This counting cannot be precisely automated as it entails stemming as part of it. For (2), we sampled word senses whose words are found and counted the covered ones. One expected source of low coverage is the named entities (NEs). So, we have labeled each word and sense as either an Arabic or named entity (NE). Table 8 shows the results of this evaluation.

Then, we computed the ontology recall using Equation 2 as 64.0%. Recall of links is computed as the number of correct links (100,264 as the multiplication of the overall count and precision in Table 7) over the number of synsets (114,135 in Table 3).

Table 8. Lexical ontology coverage over Arabic text

	Sample Size	Arabic Count	NEs Count	Arabic Cover	NEs Cover	Arabic Recall	NEs Recall	Overall Recall
Words[6]	500	444	56	23	437	98.4%	41.1%	92.0%
Senses	1000	888	112	771	21	86.8%	18.6%	79.2%
MUHIT	Recall (Words) x Recall (Senses)					85.5%	7.7%	72.9%

$$Recall\ (Ontolgy) = Recall\ (MUHIT) \times Recall\ (Links) \quad (2)$$

[6] The word stem was extracted manually before MUHIT lookup.

7 Named Entity Coverage

With an overall precision of 96.2% and recall of 64.0%, it is clear that our weakness is the recall, and by taking a look at Table 8, it is the coverage of named entities. In order to tackle this problem, we need not only to augment MUHIT with Arabic named entities, but also to link them with the relevant concepts and instances in SUMO.

We have initially considered Wikipedia as a source for named entities [28]. However, we want to link these named entities with SUMO, which requires us to find some classification or parent type for each entity. Thus, using Wikipedia will put the requirement to classify Wikipedia pages into entities and non-entities, then classify entities to different entity types. This process is a source of errors, and we are after highly precise classification, in order not to affect the achieved precision so far.

Considering this, we have moved to DBpedia [25], because its entities are classified according to a small ontology. For example, 'Barack Obama' is a 'Person', 'Microsoft' is a 'Company', and 'Paris' is a 'municipality'. In order to link with SUMO, we have identified the top three named entity classes in DBpedia as 'Person', 'Place' and 'Organization', and manually mapped them to 'Human', 'Region' and 'Organization' in SUMO. The steps of importing entities from the Arabic DBpedia are as follows:

1. Retrieve all Arabic DBpedia entities typed as 'Person', 'Place' or 'Organization'
2. Union the values of 'label', 'name', 'longName' and 'birthName' attributes
3. Ignore any name value that is not pure Arabic script
4. Search for each name value as a word in MUHIT
5. If not found, add a new word with one sense and link it to the relevant SUMO type
6. If found, check if any of the senses is linked to the same target type based on the manual mapping (either directly or to one of the SUMO children of the target type)
7. If not found as a sense, add new sense and link it to the relevant SUMO type

Applying the steps above on the Arabic DBpedia resulted in a set of 48,614 entities, which contributed to MUHIT by 46,578 words, and 47,714 senses. We have randomly sampled 200 new entities along with the linked SUMO concepts, revised them manually, and found them to be 100% correct.

7.1 Arabic Named Entities from English DBpedia

To further extend our NE coverage, we have considered the English DBpedia as another source. The basic idea is to transliterate the English name values to Arabic, and then validate the transliterated output as Arabic NEs. Some entities will remain valid after transliteration like usual Latin person names. Some others will transform to rubbish such as 'United States of America' and 'Egypt'. Some third will collide with common Arabic words like some Chinese names. We have divided this approach into two major steps; Candidate Generation and Named Entity Filtration.

Candidate Generation. In this step, we have applied (1), (2) and (3) above, but for the English DBpedia producing 2 million entities. Then we used the ATKS Transliterator

[30] to transliterate them into the Arabic script. Then, for each name, we have accepted the top 3 transliteration candidates. 'Alabama' was transliterated to 'الاباما', 'ألاباما' and 'الباما'. More candidates (up to 9) were accepted for multi-word names.

Most of these transliterated candidates are rubbish or collide with common Arabic words. As a first cleaning step, we have searched for these candidates in the 5th edition of the Arabic Gigaword Corpus [29] and kept only those candidates that were found at least 2 times. This cleaning resulted in 160,000 remaining candidates.

Named Entity Filtration. In this step, we filter the candidates to end up with a clean list of typed entities. We have proposed and applied the following steps:

1. Collected a list of sentences for each candidate (from 2 up to 100 sentences)
2. Automatically tagged the sentences for named entities using the ATKS NER [30]
3. Accepted candidates where at least one occurrence is detected by NER with the same exact text and type as in the candidates list, reducing the size to 90,540 entities
4. Manually labeled 400 candidates for correctness (both text and type must be correct)
5. Identified some of the discriminating features for correctness such as number of characters, number of agreement with NER, existence as part of another entity, number of distinct NE types for the same NE phrase … etc.
6. Trained a Bayesian Network classifier in Weka [31] using the labeled candidates
7. Classified the entities from (2) using the trained model

Applying this approach, we got a list of 82,207 Arabic entities. 75,904 of them are new to both Arabic DBpedia and MUHIT. Finally, we evaluated them manually for precision on another sample of 100 entities and found them to achieve 95.8% precision.

To summarize, we have augmented MUHIT with 47,614 correct entities from Arabic DBpedia, and 75,904 transliterated entities from the English DBpedia that are 95.8% precise. This resulted in increasing MUHIT linked synsets from 104,225 to 227,743, NE recall from 7.7% to 51.5%, and the overall MUHIT recall from 72.9% to 81.3%. On the ontology level, it increased the overall recall from 64.0% to 75.5%, and also increased the precision from 96.2% to 96.9%.

8 Conclusion and Future Work

In this paper, we have evaluated many of the viable options for generating a high-quality and comprehensive Arabic lexical ontology to be used for ontology-based Arabic NLP. We came up with a recommended combination of language resources along with a simple methodology to merge them into a comprehensive Arabic lexical ontology for Natural Language Understanding scenarios. We also increased MUHIT named entity coverage significantly, and hence the lexical ontology. Our final precision and recall for Arabic are 96.9% and 81.3% respectively in comparison with averages of 85% and 80% on Spanish and Thai as presented in the LeOnI methodology [13].

As for the future work, we want to experiment with applying the generated ontology in an Arabic NLP task such as Word Sense Disambiguation (WSD), and also consider automatic linking of OpenCyc to WordNet in order to unblock the Natural Language Generation scenarios as well.

References

1. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
2. Nirenburg, S., Raskin, V., Onyshkevych, B.: Apologiae ontologiae. Computing Research Laboratory, New Mexico State University (1996)
3. Nirenburg, S., Raskin, V.: Ontological semantics, formal ontology, and ambiguity. In: Proceedings of the International Conference on Formal Ontology in Information Systems, vol. 2001, pp. 151–161. ACM (2001)
4. Kara, S., Alan, Ö., Sabuncu, O., Akpınar, S., Cicekli, N.K., Alpaslan, F.N.: An ontology-based retrieval system using semantic indexing. Information Systems 37(4), 294–305 (2012)
5. Batet, M., Sánchez, D., Valls, A.: An ontology-based measure to compute semantic similarity in biomedicine. Journal of Biomedical Informatics 44(1), 118–125 (2011)
6. Gutiérrez, Y., Fernández, A., Montoyo, A., Vázquez, S.: UMCC-DLSI: Integrative resource for disambiguation task. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 427–432. Association for Computational Linguistics (2010)
7. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)
8. Miller, G., Fellbaum, C.: Wordnet: An electronic lexical database (1998)
9. Okumura, A., Hovy, E.: Lexicon-to-ontology concept association using a bilingual dictionary. In: Proceedings of the First Conference of the Association for Machine Translation in the Americans, pp. 177–184 (1994)
10. Khan, L.R., Hovy, E.: Improving the precision of lexicon-to-ontology alignment algorithms. In: Proceedings of AMTA/SIG-IL First Workshop on Interlinguas, San Diego, CA (1997)
11. Chen, B., Fung, P.: Automatic construction of an English-Chinese bilingual FrameNet. In: Proceedings of HLT-NAACL 2004: Short Papers, pp. 29–32. Association for Computational Linguistics (2004)
12. Malaisé, V., Isaac, A., Gazendam, L., Brugman, H.: Anchoring dutch cultural heritage thesauri to wordnet: two case studies. In: Proceedings of the ACL Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007), pp. 57–64 (2007)
13. Farreres, J., Gibert, K., Rodríguez, H., Pluempitiwiriyawej, C.: Inference of lexical ontologies. The LeOnI methodology. Artificial Intelligence 174(1), 1–19 (2010)
14. Elkateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, A., Fellbaum, C.: Building a wordnet for arabic. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006) (2006)
15. Kellogg, M.: WordReference.com English-Arabic Dictionary (1999), `http://www.wordreference.com` (accessed November 1, 2014)
16. Fouad, Y.: Arabdict Online Dictionaries (2008), `http://www.arabdict.com` (accessed November 1, 2014)

17. almaany.com, Almaany Arabic-English Dictionary (2010), http://www.almaany.com (accessed November 1, 2014)
18. Babylon. Babylon Arabic-English Dictionary (1997), http://translation.babylon.com/arabic/to-english (accessed November 1, 2014)
19. Google, Google Translate (2006), https://translate.google.com/#en/ar (accessed November 1, 2014)
20. Navigli, R., Ponzetto, S.P.: BabelNet: Building a very large multilingual semantic network. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 216–225. Association for Computational Linguistics (2010)
21. Niles, I., Pease, A.: Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In: Proceedings of the IEEE International Conference on Information and Knowledge Engineering, pp. 412–416 (2003)
22. Reed, S.L., Lenat, D.B.: Mapping ontologies into Cyc. In: AAAI 2002 Conference Workshop on Ontologies for The Semantic Web, pp. 1–6 (2002)
23. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Oltramari, R., Schneider, L., Istc-cnr, L.P., Horrocks, I.: WonderWeb deliverable D17. the WonderWeb library of foundational ontologies and the DOLCE ontology (2002)
24. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia-A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)
25. Alansary, S.: MUHIT: A Multilingual Harmonized Dictionary. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), pp. 2138–2145. European Language Resources Association (ELRA) (2014)
26. Pease, A., Niles, I., Li, J.: The suggested upper merged ontology: A large ontology for the semantic web and its applications. In: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, vol. 28 (2002)
27. Matuszek, C., Cabral, J., Witbrock, M.J., DeOliveira, J.: An Introduction to the Syntax and Content of Cyc. In: AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, pp. 44–49 (2006)
28. Alkhalifa, M., Rodríguez, H.: Automatically extending NE coverage of Arabic WordNet using Wikipedia. In: Proc. Of the 3rd International Conference on Arabic Language Processing CITALA 2009, Rabat, Morocco (2009)
29. Parker, R., et al.: Arabic Gigaword Fifth Edition LDC2011T11. Web Download. Linguistic Data Consortium, Philadelphia (2011)
30. Microsoft Research, Arabic Toolkit Service (ATKS) (2015), http://atks.microsoft.com (accessed January 1, 2015)
31. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)

Building a Nasa Yuwe Language Test Collection

Luz Marina Sierra, Carlos Alberto Cobos, Juan Carlos Corrales, and Tulio Rojas Curieux

University of Cauca, Popayán, Colombia
{lsierra,ccobos,jcorral,trojas}@unicauca.edu.co

Abstract. The nasa yuwe is the language of the Paez people in Colombia is currently an endangered language[1]. The nasa community has therefore been reviewing different strategies with the purpose of encouraging 1) the visualization process of the language and 2) the sensibilization of the use of the language, by means of computational tools. With the intention of making a contribution to both of these areas, the building of an information retrieval system (IRS) for texts written in Nasa Yuwe is proposed. This would be expected to encourage writing in Nasa Yuwe and the retrieval of documents written in the language. To implement the system, it is necessary to have a test collection with which to assess the IRS, so that the first step, prior to IRS development, is to build that test collection specifically for Nasa Yuwe texts, something which is not currently available. This paper thus presents the first test collection in Nasa Yuwe, as well as showing its construction process and results. The results allow appreciation of:1) the process of building the Nasa Yuwe test collection, 2) the queries, expert opinions and documents; and 3) a statistical analysis of the data, including an analysis of Zipf's Law[2].

Keywords: test collection, Nasa Yuwe language, information retrieval system, expert judgment.

1 Introduction

The Nasa Yuwe is one of the official languages of the Republic of Colombia. It is spoken by the Paez indigenous people of Colombia (we called Nasa community because is the way than they called themselves and therefore Nasa Yuwe is how they call their language and it's an alternative name for this language[3]) and is the most important ethnic language of the Colombian territories. It is spoken in several nasa reservations and settlements in several municipalities in the departments of Caqueta, Putumayo, Meta, Tolima, Valle del Cauca, Cauca, and Huila[4]. It is estimated that the nasa population now borders on 200,000 people, of which 75% are active speakers of the nasa language[5]. It should be noted however that the sociolinguistic status of Nasa Yuwe is that of an endangered language, since due to a range of cultural, social, geographical and even historical factors it has lost much ground[6]. Added to the above, its alphabet dates from only the twentieth century, and its unification [1] even later. As a result, very few texts are to be found written in this alphabet. For the purposes of carrying out this work, written texts with the unified alphabet have been

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 112–123, 2015.
DOI: 10.1007/978-3-319-18111-0_9

sought[7]. It is also worth clarifying that the description of the language is a work in progress as is shown in[5],[8-10] [11].

The Government of Colombia and indigenous organizations have been developing strategies to promote the visualization of the Nasa language, including the use of technology as a strategic opportunity that includes approaching the nasa way of life in different contexts. As such, possible computational techniques come to mind, such as Information Retrieval Systems (IRS), promoting possibilities for people from the Nasa community to interact in Nasa Yuwe, providing motivation for the writing and collection of documents written in this language, upon which an information recovery model can be operated that supports awareness of its use through computational tools within a range of activities of, for example, a social, educational, or political nature.

Account was taken of the fact that 1) IRS support the search and retrieval of documents in a specific language[12] ; 2) their processing tasks (tokenize, filtering, transformation, stop word removal, stemming, etc.) are highly dependent on the language in which the recovered documents are written, as shown in several studies in the literature [13, 14]; and 3) the preferred method for evaluating the performance of IRS are test collections. As such, each language requires the suitable design of one or more collections to be used for IRS evaluation. In the case of the Nasa Yuwe language it should be stressed that no test collection for IRS exists. It was therefore necessary to build one, which is the issue addressed by this work.

The rest of the paper is organized as follows: In Section 2, background on IRS evaluation and test collections is presented; in section 3, the process of building the test collection for Nasa Yuwe is shown; in Section 4, the results are presented; and finally, Section 5 presents conclusions and future work intentions.

2 Background

2.1 Evaluation in Information Retrieval Systems - IRS

IRS need to be evaluated in such a way that their performance and quality of retrieval can be measured. There are two main types of evaluation: user-based and system-based[12]. Evaluation with users in the present day is extremely valuable, but impractical given the volume of needs that exist, the determination of appropriate controls, and the incorporation of users, where their training and monitoring can take a long time. System-based evaluation involves sending a predefined set of queries to the system and measuring the relevance of results ranked without the intervention of humans in the process steps. As such, they require little effort, are faster, impartial, reliable and combined with the use of standard statistical measures have become the standard for the design and testing of IRS so that comparing the performance of the system is that much easier[12].

2.2 Test Collection

The first IRS evaluation proposal was known as the Cranfield studies in the fifties. The components of the Cranfield experiments were a small collection of documents,

a set of test queries and for each query a set of relevance judgments regarding the documents in the collection. All these items are known as the test collection and this method is still the most widely used for conducting IRS evaluation[15]. The test collections are therefore useful because they allow the control of some variables that affect the performance of the retrieval, increase the strength of comparative experiments, while reducing costs in comparison to other assessment types[15]. Much bigger collections are used today to better simulate the search requirements, but building and working with large collections with sizeable documents is not very easy[15]. In Table 1 below, a comparison is presented between some existing collections[2]:

Table 1. Description of several test collections. Adaptation of [15]

Collection	Language	Topic	No. of Documents	Size
Cranfield II	English	Abstract of scientific articles	1.398	579 KB
TREC-AP	English	AP news services (1988-1990)	242.918	0.7 GB
GOV2	English	Government web pages	25.205.179	426 GB
NTCIR-4[1] PATENT [16]	English, Chinese, Japanese and Korean	Documents of patents	7.000.000	65 GB
CLEF-multi	Dutch, English, Finnish, French, German, Italian, Portuguese, Russian, Spanish, Swiss	Newspapers and news services for a specific time period (1994 –1995)	1.869.564	4.7 GB
RCV1[2]	English	News stories	810.000	2.5 GB
Reuters-21578[3]	English	Stories of news services in 5 different categories [17]	21.578	28 MB
20 Newsgroups [17]	English	Newsgroups in 20 different groups	18.846 newsgroup documents	13.8 MB compressed

Some relevant work that allows an understanding of the need to build test collections in specific languages to evaluate IRS in each language are: 1) Sorani, one of the main branches of the Kurdish language. Its main contribution is the Pewan, the first standard test collection available for assessing Sorani IRS[18]; 2) a test collection in Italian, made up of complete news items, hypothetical queries, actual queries from potential users, a manual classification compiled by experts and an automatic document classification system[14]; 3) one of the first test collections for Farsi, the official language of Iran. It presents the process of building the collection and its comparisons with other Farsi collections[19]; 4) a test collection developed to answer questions in the Macedonian language, which can be used to develop and evaluate IRS. The collection consists of 4 documents and 163 multiple choice questions taken from the History of Informatics and Computer Applications courses that are part of the curriculum of the University. The preliminary results showed that despite being small it can be effectively used for its purpose[20]; 5) A test collection for Hamshahri, a branch of

[1] http://research.nii.ac.jp/ntcir/permission/ntcir-4/perm-en-PATENT.html
[2] http://trec.nist.gov/data/reuters/reuters.html
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/

the Persian (or Farsi) language. It consists of newspaper articles from 1996 to 2002. The size of the documents ranges from short news items (less than 1KB) to long articles on average of 1.8 KB[21]. In Table 2, an overview of the collections presented above is present.

Table 2. Description of collections for specific languages

Collection	Language	Topic	No. of Documents	Size
Pewan[18]	Sorani	News articles (2003 -2013). Additionally a list of prefixes and suffixes, stopwords, translation of Pewan's queries from English.	115.340	97,8 MB
Italian test collection [14]	Italian	A large set of complete documents of news from one year of a newspaper in Italian	70.000 documents	Not available
Mahak [19]	Farsi	ISNA news articles in 12 news categories	3007 documents 216 queries	Not available
To answer questions [20]	Macedonian	Documents of courses of history and computing form the Institute of Informatics at the Faculty of Natural Sciences and Mathematics in the University of Skopje	4 documents y 163 multiple choice questions	Not available
Hamshahri[21]	Farsi (Persian)	Newspaper news articles from 1996 to 2002	166.774 documents and 65 queries	564 MB

Among the most common methods for making relevance judgments in a test collection are: 1) Pooling[19], which creates a subset configured with the list of retrieved documents with the highest score for a query, which have been obtained by a IRS. This method, proposed by Gilbert and Sparck Jones 1979[22] has been used inter alia for TREC-6 and is well known for its efficiency. 2) Move to Front (MTF) Pooling [15], is an improvement on the standard Pooling method using a variable number of documents retrieved by different IRS depending on the performance of each system. 3) Interactive Searching and Judging (ISJ) [15] aims to create a pool of document judgments with minimal effort. To achieve this, a group of searchers are asked to find and judge as many relevant documents as possible in a short period of time. This method is not very popular, because it has the risk that relevance judgments can be biased by the search systems that created them. It is only usually applied therefore as a supplementary method to improve the quality of a test collection.

3 Building the Test Collection of Texts Written in Nasa Yuwe

To build the test collection for texts written in Nasa Yuwe the process was adapted to take account of the current conditions of the language. In Figure 1, a brief outline of the process used to build the collection is presented.

3.1 Selecting Documents

In order to select the documents of the Nasa Yuwe test Collection we followed this steps: 1) To make the document collection, fieldwork was conducted taking into

account the existence of many variants in the Nasa Yuwe language associated with the geographical area in which the reservation of each community is located. It was therefore necessary to delimit the documents in the collection to those written in the unified alphabet[7], which consists of 32 vowels and 61 consonants. It should be noted that few written documents exist, due to the newness of the alphabet and the sociolinguistic status of the language. 2) Review of the documents selected was carried out with expert Nasa Yuwe speakers. The work involved reviewing these texts in both their writing and grammar.

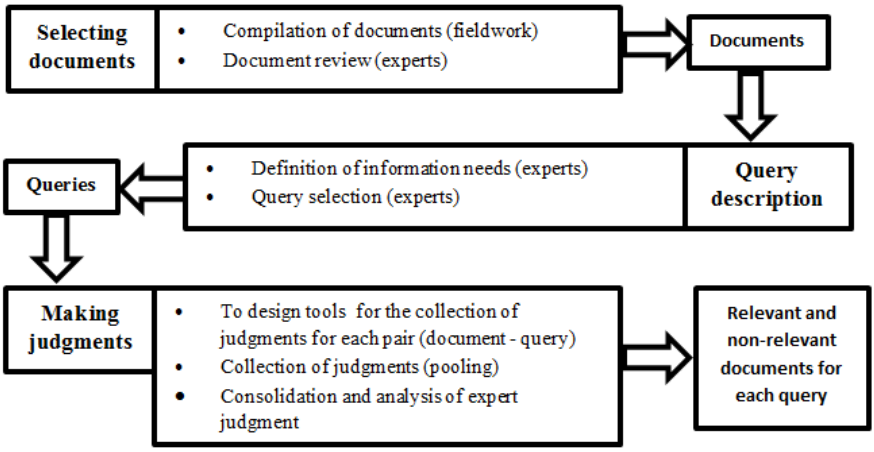


Fig. 1. Description of the process of building the test collection

3.2 Query Description

The Queries for the Nasa Yuwe test Collection were done this: 1) To define the needs for information, the issues discussed in the documents selected were taken into account, producing a list of 20 information needs. 2) Query selection was carried out by prioritizing the issues with the most required information in reference to the selected texts. Work was done with the experts to finally come up with eight queries.

3.3 Making Judgments

- Relating to design of the tools for the collection of relevance judgments for each pair (document-query), two scenarios were considered: 1) Collection of expert judgments virtually, so that a tool was designed to allow collection of this information through a web application. 2) Collection of judgments by printed format where each expert issued their judgment for each pair. The instruments developed for making judgments of each pair (document-query) took into account the definition of a scale of 4 values: Very Relevant (VR), Relevant (R), Barely Relevant (BR) andP Not Relevant (NR), which helped to avoid confusion among the

experts and in collecting the information. For the purposes of this study, scenario 2 was used, given the nature of the participating experts.

- To collect the judgement, the opinions of experts for each document pair query were used, since there were no previous judgements in this respect nor preselections for any IRS. The process of selecting the experts was done taking due consideration of the diversity of variants in speakers of Nasa Yuwe. The majority were teachers interested in the visualization of the language through different strategies.

4 Results from Building the Collection

4.1 Documents

In the first instance, documents relating to stories about the Nasa culture and worldview were selected. Those texts were find in the library of the Indigenous Intercultural Autonomous University - UAIIN[23], those texts are used as teaching materials in different scenarios (schools of nasa people context and nasa training teachers). The texts selected are described below: 1) Area Nasawe'sx Fxinzenxi – **Nasa Stories and Worldview[24]**. This text contains personal stories about the Nasa way of life, written in Nasa Yuwe and with a contextualized translation into Spanish. 2) Eç thegya' ipi'ki' tha'w - **We invite you to read[25]**. Short texts about descriptions of Nasa life. Some activities to support these descriptions, written in nasa Yuwe and with a Spanish translation, are proposed. 3) Nasawe'sx Kiwaka Fxi'zenxi Ẽen. [26]. This text contains an investigation into ancestral knowledge in the Nasa culture written in both Nasa Yuwe and Spanish. 4) Pees kupx fxi'zenxi. **The metamorphosis of life**. [27]. Stories about the Nasa caciques in ancient times were taken from this text.

Secondly, a document review was conducted which included: their digitization and correction to the level of special characters appropriate to the writing of Nasa Yuwe and to the level of grammar of Nasa Yuwe. This process required the help of two teachers who are speakers of nasa yuwe, with whom a preliminary assessment of the texts was made in reference to making judgments, and concluding that the text of [27] despite being written in the unified alphabet was not appropriate for inclusion in the collection given the difficulty presented in its reading, because of the use of words specific to the Toribio variant not commonly used in other variants of nasa yuwe. As such, these stories were taken out of the documents in the collection. Thus remained only the first three texts described above. All documents were put into UTF-8 text format to support the characters of the unified Nasa Yuwe alphabet. In Table 3, a fragment of a document written in Nasa Yuwe is shown. The documents mostly contain a title and a text.

Finally, 97 documents of text written in Nasa Yuwe were obtained for the test collection, having an average size of 1.5 KB per document. In Figure 2, the size distribution of the documents is presented. The number of terms in the documents of the collection ranges from 15 to 500, and the documents have an average of 103 terms.

Table 3. Fragment of a story in Nasa Yuwe: "The origin of the Nasa people". Source: [24]

Text written in nasa yuwe	*Text written in English*
Nasa vxanxi's pta'sxnxi. Txaniteya' kiwe wala u'sene'yũ' aça' khã'sx ũskiweçxane'yũ' mẽh kũh jwed ksxa'w ũskiweyũ'ne'sa', vxite ne'jwe'sxtayu' aça' vxitesa' nuuçxkwẽsane'tayu'. Ne'jwe'sxyũ' puutx ptamne'tayũ'. Piçthẽ'jsa' tayne' yaaseyũ', uysa' umane' yaaseyu'. Naa je'zsa ũusyahtxya' uweçxa naa kiwete kĩhçxahne' peejxsa ũsu' txãatxi's vxitya' takhne'nta: Yu'a's, fxtũu tasxtxi's, kwettxi's, kĩnjwã jxukane'nta vxitsa', nmehte' naa kiwe' nasane' peejxyu' aça' nasa'swa vxitya' yahtxne'nta kxteeçxãh puutx fxi'zekahn jĩçxa.	**The origin of Man.** Long ago when the earth was young, there were only spirits of the wind. Some were higher spirits and others were lesser spirits or subordinates. Each of these had its own partner. One of the higher spirits was called TAY and the other was called UMA. These began to think about how to populate the earth and decided to create the water, the plants, the animals, the stones and all that now exists on earth. Finally they decided to create Man so that he would care for them and that all of us would live in harmony together in one place.

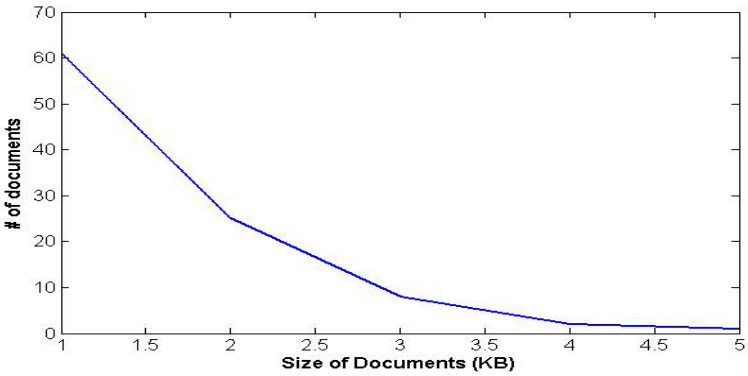


Fig. 2. Document size distribution of the collection

4.2 Queries

Table 4 shows the list of selected queries for the test collection with its description and in the Figure 3, the distribution of terms per query is shown, with an average of 4.4 words per query, a minimum of 3 words and a maximum of 7 words, which allows an understanding of the number of terms required to express information needs in this language.

4.3 Judgment for Each Pair (Document –Query)

The experts selected for making judgments were teachers who spoke Nasa Yuwe and an expert Nasa Yuwe linguist from different variants (Vitoncó (Tierradentro), Jambaló, Munchique-Tigres. Caldono y pueblo nuevo). It should be noted that the process of making judgments by each expert was costly in time and effort; given the scale of the exercise and that no precedents existed in this regard.

To consolidate the judgments, all values were taken to have equal weight and two groups were made: in the first, values of Very Relevant and Relevant (VR+R) obtained in the scale were pooled together, the second group featuring the Barely Relevant and Not Relevant (BR+NR) values.

Table 4. Query descriptions

No.	Need for information	Title	Query description in Nasa Yuwe	No. of \words per query	Narrative
1	To learn about the origen and age (planting) of the corn	Kutxh	kutxh yuwe's jiyuka ki' uh a'te's txãwẽy	7	Relevant documents that could inform the Nasa user of the IRS about origin and age of corn
2	To find out about the aid and work of traditional doctors	Thẽ' walawe'sx	Thẽ' walawe'sx majii	2	Relevant documents that could inform the Nasa user of the IRS about traditional doctors
3	To learn about the phases of the moon	A'te	A'te dxi'j / a'te yuwe's jiyuna	6	Relevant documents that could inform the Nasa user of the IRS about the phases of the moon.
4	Stories about origin/birth - appearance in Nasa culture.	Upxhnxi /vanxi	Upxhnxi yuwe / vxanxi yuwe	5	Relevant documents that could inform the Nasa user of the IRS about the origen and first appearance of the Nasa culture.
5	About the hen and the chicken in the Nasa culture	Atalx	Atalx wejxa's jiyuna	3	Relevant documents that could inform the Nasa user of the IRS about the hen and the chicken in the Nasa culture
6	Stories of the wind in the Nasa culture	Wejxa	Wejxa yuwe's jiyuna	3	Relevant documents that could inform the Nasa user of the IRS about stories of the wind in the Nasa culture
7	Stories about the sun in the Nasa culture	Sek	Sek yuwe's jiyuna /sek dxi'j	5	Relevant documents that could inform the Nasa user of the IRS about stories of the sun in the Nasa culture
8	Stories about caciques in the Nasa culture	Sa'twe'sx	Sa'twe'sx yuwe's jiyuna	3	Relevant documents that could inform the Nasa user of the IRS about stories of the Nasa caciques.

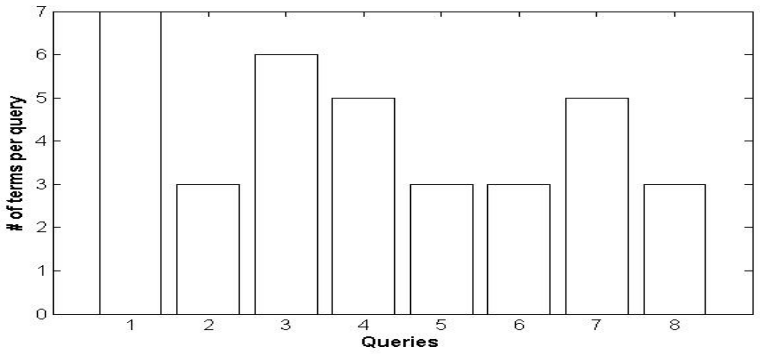


Fig. 3. Query Length

Finally, the values obtained from the aforementioned groups were taken as a reference to determine the relevance or non-relevance of a document for each query, as follows: 1) documents that obtained 60% of relevance are considered relevant in the first group (VR+R) of the total responses from the experts in each query were considered relevant 2) the documents that obtained 70% as a value in the second group (BR+NR) were considered not relevant. Determining the percentage relevance of the documents in 60% allowed differentiation of the scale of relevance level of each relative to each query. In the collection, 63% of documents are relevant for the eight queries. Figure 4 displays the number of relevant documents for each query.

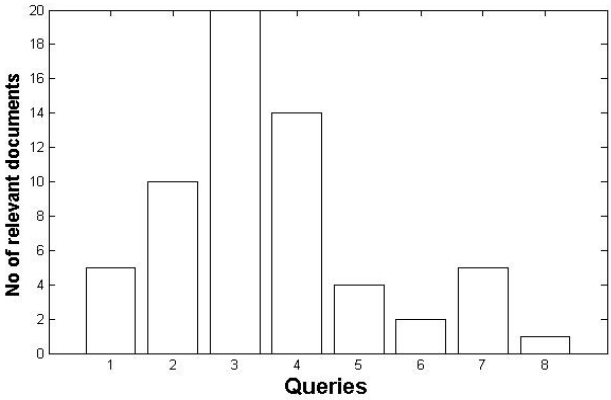


Fig. 4. Number of relevant documents for each query

4.4 Description of the Test Collection

In Table 5, the first version of the Nasa Yuwe test collection is described. There are various attributes of the Nasa Yuwe collection are also summarized. It is published in: http://www.ewa.edu.co/coleccion.

Table 5. Attributes of the nasa yuwe collection

Attributes	Value
Collection size	113 KB
Document formats	Text (UTF-8)
No. of documents	97
Terms by document average	103
Corpus size in numbers of words	9955
Average document size	1.5 KB
No. of queries	8

4.5 Statistical Analysis of the Collection

In Table 6, some of the most frequent terms in the documents of the collection are presented, which is of great importance for the task of stopword removal in building the IRS just as can be seen the length of the words, taking the Nasa unified alphabet [7] into account. About 5,000 different words were found, including conjugations.

Table 6. Most frequent terms in the document collection

Word	Frequency	Length	Word	Frequency	length
Wala	140	4	Teeçx	76	4
Txãa	111	3	txã'w	68	3
Naa	107	3	kwe'sx	59	3
a'te	104	3	ki'	50	2
Nasa	91	4	Luuçx	49	4

The frequency of some terms in the collection behaves as shown in Figure 5. For the first 50 terms, being able to see Nasa Yuwe fulfill Zipf's Law, which applies to the majority of languages.

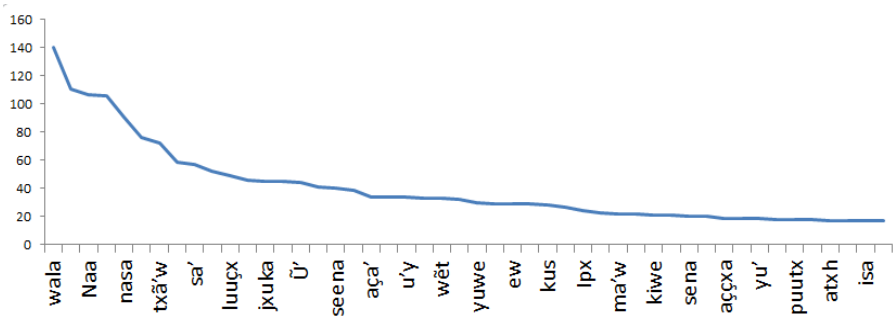


Fig. 5. Frequency of some terms in the collection

5 Conclusion and Future Work

Building a test collection for Nasa Yuwe represents a very important starting point for carrying out further experiments in information retrieval. This paper presents the first test collection developed for the evaluation of IRS on texts written in Nasa Yuwe. The building process, the collection itself, and some general statistics for the purposes of text recovery and language processing are presented in detail. At the same time comparison is made possible with collections in other languages in sociolinguistic situations similar to Nasa Yuwe. Furthermore, a complete reference on other existing test collections is presented that makes it possible to compare the importance of this work in terms of building a test collection suited to the Nasa Yuwe tongue. In addition, Nasa Yuwe test collection is not only useful for the authors of this paper but for teachers and students of different nasa contexts. These contexts are social and family environment, basic schooling and training of new teachers in traditional and

indigeneous universities such as UAIIN (e.g., University of Cauca have different programs in Ethnic Education, anthropology, linguistic and language revitalization), as is the case of the experts who participated in the review and issuance of relevance judgments for each query.

As future work would involve building an IRS for processing queries made in the Nasa Yuwe language. Additionally, there is awareness of the need to expand and improve several aspects that favor the quality of the collection. The development of this experience is expected to encourage writing in Nasa Yuwe and the interaction of the people of the Nasa community involved in this process.

Acknowledgements. We are grateful to Universidad of Cauca and its research groups GTI, GIT and GELPS of the Computer Science, Telematics and Anthropology departments. We are especially grateful to Colin McLachlan for suggestions relating to the English text.

References

1. Rojas Curieux, T.: Por los caminos de la recuperación de la lengua Paéz (nasa yuwe), Popayán Letrarte editores (2006)
2. Manning, C., Raghavan, P., Shütze, H.: An Introduction to Information Retrieval. Cambridge University Press (2009)
3. Moseley, C.: Atlas de las lenguas del mundo en peligro. Ediciones UNESCO, Popayán (2010), Versión en línea: `http://www.unesco.org/culture/en/endangeredlanguages/atlas` (accessed Marzo 2013)
4. Instituto Colombiano de Cultura Hispánica: Geografía Humana de Colombia, Región Andina Central Tomo IV Volumen II, Bogotá: Banco de la República (2000)
5. Rojas, C., Esbozo Gramatical de la, T.: lengua nasa (lengua Paéz). In: El Lenguaje en Colombia, Tomo I: Realidad Lingüística de Colomba, Bogotá, Academía Colombiana de la Lengua e Instituto Caro y Cuervo, pp. 479–495 (2009)
6. Universidad del Cauca, CRIC-PEBI-Comisión General de Lenguas: Estudio Sociolingüistico Fase preliminar. Base de datos - CRIC 01/2007 Lengua Nasa Yuwe y Namtrik. Popayán, Cauca, Colombia (2008)
7. Farfán Martínez, M., Rojas Curieux, T.: Zuy Luuçxkwe kwe'kwe'sx ipx kwetuy piyaaka. Cartilla de aprendizaje de nasa yuwe como segunda lengua, Buenos Aires (2010)
8. Jung, I.: Gramática del Páez o nasa yuwe. Descripción de una Lengua Indígena de Colombia. LINOM GmbH (1984, 2008)
9. CRIC y el Programa de Dllo Rural en la Región de Tierra Dentro Cxhab Wala -PT/CW, Diccionario Nasa Yuwe - Castellano, Primera ed., Popayán: Litografía San José (2005)
10. Rojas Curieux, T., Perdomo Dizu, A., Corrales Carvaja, M.H.: Una Mirada al nasa yuwe de Novirao, Primera ed., Popayán: Sello Editorial Universidad del Cauca (2009)
11. Rojas Curieux, T.E.: La lengua paéz una visión de su gramática, primera ed., M. d. Cultura, Ed., Bogotá: Panamericana Formas e Impresos S.A (1998)
12. Carterette, B., Voorhees, E.M.: Overview of Information Retrieval Evaluation. In: Current Challenges in Patent Information Retrieval, pp. 69–85. Springer (2011)
13. Jadidinejad, A.H., Mahmoudi, F., Dehdari, J.: Evaluation of Perstem: A Simple and Efficient Stemming Algorithm for Persian. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 98–101. Springer, Heidelberg (2010)

14. Agosti, M., Bacchin, M., Ferro, N., Melucci, M.: Improving the Automatic Retrieval of Text Documents. In: Peters, C., Braschler, M., Gonzalo, J. (eds.) CLEF 2002. LNCS, vol. 2785, pp. 279–290. Springer, Heidelberg (2003)
15. Peters, C., Braschler, M., Clough, P.: Evaluation for Multilingual Information Retrieval Systems. In: Multilingual Information Retrieval, pp. 129–169. Springer (2012)
16. NTCIR Project, NTCIR Project 2007 (En línea), http://research.nii.ac.jp/ntcir/permission/ntcir-4/perm-en-PATENT.html (Último acceso: December 5, 2014)
17. Ribeiro-Neto, B., Baeza-Yates, R.: Modern Information Retrieval -the concepts and technology behind search, 2nd edn. Addison Wesley, Harlow (2011)
18. Sheykh Esmaili, K., Salavati, S., Yosefi, S.: Building A Test Collection For Sorani Kurdish. In: ACS International Conference on Computer Systems and Applications (AICCSA), Ifrane (2013)
19. Esmaili, K., Abolhassani, H., Neshati, M., Behrangi, E.: Mahak: A Test Collection for Evaluation of Farsi Information Retrieval Systems. In: IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2007, pp. 639–644. IEEE (2007)
20. Armenska, J., Tomovski, A., Zdravkova, K., Pehcevski, J.: Information Retrieval Using a Macedonian Test Collection for Question Answering. In: Gusev, M., Mitrevski, P. (eds.) ICT Innovations 2010. CCIS, vol. 83, pp. 205–214. Springer, Heidelberg (2011)
21. AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., Oroumchian, F.: Hamshahri: A standard Persian text collection. Knowledge-Based Systems 22(5), 382–387 (2009)
22. Kuriyama, K., Kando, N., Nozue, T., Eguchi, K.: Pooling for a Large-Scale Test Collection: An Analysis of the Search Results from the First NTCIR Workshop. Information Retrieval 5(1), 41–59 (2002)
23. Consejo Regional Indígena del Cauca – Programa de Educación Bilingüe e Intercultural (PEBI - CRIC): Universidad Autónoma Indígena Intercultural –UAIIN (2015), http://www.pebi-cric.org/uaiin.html(accessed Marzo 2015)
24. Consejo Regional Indígena del Cauca – Programa de Educación Bilingüe e Intercultural (PEBI - CRIC): Cuentos y Cosmovisión Nasa. Area Nasawe'sx Fxinzenxi, Segunda ed., Popayán (2010)
25. Consejo Regional Indígena del Cauca – Programa de Educación Bilingüe e Intercultural (PEBI - CRIC): Te invitamos a leer. Eç thegya' ipi'ki' tha'w, Primera ed., Cali: Grafitextos (2007)
26. Asociación de Cabildos Ukawe'sx Nasa Çxhab, Consejo Regional Indígena del Cauca – Programa de Educación Bilingüe e Intercultural (PEBI - CRIC): NASAWE'SX KIWAKA FXI'ZENXI ÊEN, Primera ed., Cali: Grafitextos (2006)
27. Yule Yatacue, M., Vitonas Pavi, C.: Pees kupx fxi'zenxi. La metamorfosis de la vida, Tercera ed., Toribio, Cauca: Grafitextos (2012)
28. Consejo Regional Indígena del Cauca – CRIC, Programa de Educación Bilingüe e Intercultural.: Sistema Educativo Indígena Propio -SEIP. Primer Documento de Trabajo (2011)

Morphology and Chunking

Making Morphologies the "Easy" Way

Attila Novák[1,2]

[1] MTA-PPKE Hungarian Language Technology Research Group,
[2] Faculty of Information Technology and Bionics
Pázmány Péter Catholic University
50/a Práter street, 1083 Budapest, Hungary
novak.attila@itk.ppke.hu

Abstract. Computational morphologies often consist of a lexicon and some rule component, the creation of which requires various competences and considerable effort. Such a description, on the other hand, makes an easy extension of the morphology with new lexical items possible. Most freely available morphological resources, however, contain no rule component. They are usually based on just a morphological lexicon, containing base forms and some information (often just a paradigm ID) identifying the inflectional paradigm of the word, possibly augmented with some other morphosyntactic features. The aim of the research presented in this paper was to create an algorithm that makes the integration of new words into such resources similarly easy to the way a rule-based morphology can be extended. This is achieved by predicting the correct paradigm for words not present in the lexicon. The supervised machine learning algorithm described in this paper is based on longest matching suffixes and lexical frequency data, and is demonstrated and evaluated for Russian.

Keywords: morphology, paradigm prediction, Russian.

1 Introduction

Morphological analysis is an important task in any natural language processing chain, preceding any further analysis of texts. It is also unavoidable in information retrieval, or indexing algorithms, where the lemma of words are to be used in order to have a robust representation of the information present in the documents. In this case, the morphosyntactic features identifying the specific member of the paradigm of the lexical item are irrelevant, only the lemma is required.

Large-scale computational morphologies are usually created using a morphological grammar formalism that minimizes the amount of information necessary to include in the source lexicon about each lexical item by providing some rule-based method of formalization of the morphological behavior of words. This allows an easy extension of the morphology with new lexical items. This approach also gives the creator of the morphology complete control over the quality of the resource. Building rule-based morphological grammars, however, requires threefold competence: familiarity with the formalism, knowledge of the morphology,

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 127–138, 2015.
DOI: 10.1007/978-3-319-18111-0_10

phonology and orthography of the language, and extensive lexical knowledge. Many morphological resources, on the other hand, contain no explicit rule component. Such resources are created by converting the information included in some morphological dictionary to some simple data structures representing the inflectional behavior of the lexical items included in the lexicon. The representation often only contains base forms and some information (often just a paradigm ID) identifying the inflectional paradigm of the word, possibly augmented with some other morphosyntactic features. With no rules, the extension of such resources with new lexical items is not such a straightforward task, as it is in the case of rule-based grammars. However, the application of machine learning methods may be able to make up for the lack of a rule component. In this paper, we intend to solve the problem of predicting the appropriate inflectional paradigm of out-of-vocabulary words, which are not included in the morphological lexicon. The method is based on a longest suffix matching model for paradigm identification, and it is showcased with and evaluated against an open-source Russian morphological lexicon.

The context in which we explored the possibilities of automatic paradigm identification, was the following task. We needed to make a pop-up dictionary capable of handling and correctly lemmatizing all inflected word forms of the vocabulary of a specific Russian–Hungarian dictionary. The morphological engine integrated in the dictionary program is Humor ([12,10]), a constraint-based morphological analyzer, which was first developed for Hungarian morphology. Instead of creating a Humor-based Russian morphology from scratch, we decided to adapt an LGPL-licensed Russian resource, available from www.aot.ru ([13]). The core vocabulary of this morphology is based on Zaliznyak's morphological dictionary [16]. It contains 174 785 lexical entries, each of which are classified into one of 2 767 paradigms. The resource was converted to the Humor formalism, and its coverage needed to be extended to cover the whole vocabulary of the dictionary. For the evaluation of the performance of the paradigm assignment algorithm, we used various disjunct parts of the aot resource. In addition, we used the frequency distribution of Russian lemmas, taken from Serge Sharoff's Russian internet frequency list.[1]

The paper is structured as follows: after a short summary of related work, the features used for predicting Russian inflectional paradigms are described in Section 3. This is followed by description of the suffix model and the ranking algorithm we use. Finally, in Section 6, the performance of the system is evaluated, followed by an error analysis.

2 Related Work

Morphological paradigm prediction has been a field of interest, especially for researchers dealing with inflectional, or at least compounding languages. Such languages have a complex morphology, which cannot be covered by hand-made

[1] `http://corpus.leeds.ac.uk/frqc/internet-ru.num`

lexical resources. Some studies aim at solving this problem by learning inflectional paradigms from raw text corpora by clustering word forms in the corpus and analyzing the resulting clusters ([9,8,3]). Other unsupervised methods applied to morphology induction are that of [15], [6] and [5], the latter using morphemes to encode a corpus by grouping morphemes into structures, called signatures, representing inflectional paradigms. These models, however, mainly aim at only segmenting word forms into stems and affixes: stem alternations cause paradigms to be scattered into unrelated subparadigms. However, the performance of unsupervised methods is far behind those using existing resources either as an inventory of inflectional pattern rules, or as annotated data for supervised machine learning algorithms.

Raw text corpora are also used in approaches where word form statistics are used to validate inflectional forms generated by a predicted paradigm candidate for a given word. If the resulting word forms are not represented in a corpus, then the paradigm is not valid. Some examples for such methods are described in [4] and [11]. The algorithm of [7] exploits both lexical features and corpus-based information to determine inflectional behavior by analogy. The author of [14] also defines string-based and corpus-based features used for a support vector machine classifier to decide if a predicted paradigm is valid or not. The most similar approach to our method is the one used in [1], implemented in parallel with our research, however they emphasize paradigm generalization.

Our approach differs from most of the previous ones in that we use a morphological lexicon as annotated data and the frequency distribution of raw text corpora. We address the problem of predicting inflectional paradigms based on the lemma and some given lexical features which are usually available in some less-sophisticated dictionaries. Based on the information coming from the dictionary, the morphological lexicon can be extended in a more robust manner than in cases when only raw word form corpus frequency data is available, and lemma, categorial features and the paradigm all need to be estimated from that data.

3 Features Affecting the Paradigmatic Behavior of Russian Words

When attempting to predict the inflectional paradigm for Russian words, certain grammatical features of the lexical item need to be known in order to have a good chance of guessing right. Lemma and part of speech are obviously necessary features, although part of speech can be guessed from the lemma for adjectives and verbs with rather good confidence. Nevertheless, we assumed these to be known, as these properties of words are present in any dictionary.

For nouns, a number of additional features (gender, countability and animacy) play a role in determining the morphosyntatctic feature combination slots which make up the paradigm of the given lemma. There are also nouns, which are undeclinable. Of these features, gender is indicated for each headword in any dictionary, and undeclinable nouns are also usually marked as such.

Certain abstract, collective and mass nouns (and, in the aot resource, also many proper names) lack plural forms, while there are also pluralia tantum, which have no singular. Some of the latter, however, are easier to recognize, due to their lemma exhibiting typical plural morphology.

Animacy affects the nominal paradigm in a manner that does not influence the actual set of possible word forms. However, there is a case syncretism in Russian, which depends on animacy. For animate nouns, plural accusative coincides with genitive (for masculine nouns, the same applies also to singular). For inanimate nouns, on the other hand, the form of accusative matches that of the nominative. This difference is still present in the case of homonyms, where one of the senses of the word is animate, and another form is inanimate. This phenomenon is illustrated in Figure 1 with the word *ёж* 'hedgehog: animal', and 'Czech hedgehog: a static anti-tank obstacle'. Note, however, that the animacy feature, although it is present in the aot lexicon, is not generally made explicit in other dictionaries, because a human user can infer this information from the meaning of the word. We thus have not used this information.

ёж[num:Sg.cas:Nom]	ёж[num:Sg.cas:Nom]
ежа[num:Sg.cas:Gen]	ежа[num:Sg.cas:Gen]
ежу[num:Sg.cas:Dat]	ежу[num:Sg.cas:Dat]
ежа[num:Sg.cas:Acc]	ёж[num:Sg.cas:Acc]
ежом[num:Sg.cas:Ins]	ежом[num:Sg.cas:Ins]
еже[num:Sg.cas:Prp]	еже[num:Sg.cas:Prp]
ежи[num:Pl.cas:Nom]	ежи[num:Pl.cas:Nom]
ежей[num:Pl.cas:Gen]	ежей[num:Pl.cas:Gen]
ежам[num:Pl.cas:Dat]	ежам[num:Pl.cas:Dat]
ежей[num:Pl.cas:Acc]	ежи[num:Pl.cas:Acc]
ежами[num:Pl.cas:Ins]	ежами[num:Pl.cas:Ins]
ежах[num:Pl.cas:Prp]	ежах[num:Pl.cas:Prp]
(a) ёж[N.gnd:Mas.ani:**Ani**][:8];	(b) ёж[N.gnd:Mas.ani:**Ina**][:9];

Fig. 1. Differences in case syncretism of the lemma (*ёж* 'hedgehog') depending on whether it is animate (a) or inanimate (b)

Similarly, the set of valid morphosyntactic feature combinations for verbs depends on verbal aspect and transitivity/reflexivity. Thus, these properties need to be known for verbs, and, indeed, they are listed in dictionaries. E.g. nontransitive verbs lack passive participles; verbs of perfective aspect lack present participle forms; and many verbs of imperfect aspect lack past participial (especially passive) forms. The adverbial participial forms a verb may assume also depend on aspect (and also on other idiosyncratic lexical features).

Defectivities of the adjectival paradigm, e.g. the lack of short predicative forms and synthetic comparative and superlative forms depend on semantic and other, seemingly idiosyncratic, features of the lexeme. E.g. relational adjectives

usually lack these forms. Such properties, however, were not made explicit in the aot lexicon, neither are they present in normal dictionaries, so we did not use any lexical features for adjectives beside part of speech.

Thus, when defining the feature set for predicting inflectional paradigms of words, we assumed that the lemma and the lexical properties mentioned above: part of speech, gender, verb type, etc., are known. Other morphological characteristics relevant for inflection that cannot be derived neither from a simple dictionary, nor from the surface form of a word, such as animacy, optional stress variation, idiosyncratic orthographic variations, or other irregularities were not made available to the system. Thus, our model is not necessarily able to predict paradigmatic behavior depending on such features.

The other set of features we used are n-character-long suffixes of the lemma for various lengths n. The maximum suffix length is a parameter of the algorithm. It was set to 10 in the experiments reported in this paper. In order to exploit this information, a suffix model is created based on the lexicon. An illustration of how this model including both the endings and the lexical features is generated is shown in Figure 2.

4 Creation of the Suffix Model

A suffix trie is built of words input to the training algorithm in the form shown in the right column of Figure 2.

```
мумиё[N.n.*.-];prd:25          мумиё|n*[N.n-25]
остриё[N.n.-];sfx:ё;prd:1709   остри#ё|n[N.n-1709]
бабьё[N.n-];sfx:ё;prd:210      бабь#ё|ns[N.n-210]
дубьё[N.n-];sfx:ё;prd:210      дубь#ё|ns[N.n-210]
свежевьё[N.n-];sfx:ё;prd:210   свежевь#ё|ns[N.n-210]
цевьё[N.n.-];sfx:ьё;prd:1433   цев#ьё|n[N.n-1433]
жнивьё[N.n];sfx:ё;prd:1103     жнивь#ё|n[N.n-1103]
суровьё[N.n];sfx:ё;prd:210     суровь#ё|ns[N.n-210]
мостовьё[N.n];sfx:ё;prd:210    мостовь#ё|ns[N.n-210]
```

Fig. 2. A portion of the suffix model. The format of the right column is: `lem#ma|lex-features[PosTag-paradigmID]`, where `ma` is a required ending of the lemma for all items in the paradigm identified by `paradigmID`.

The lemma is decorated with the following features (from right to left):

– The tag in brackets consists of two parts: part of speech (and, in the example in Figure 2: gender) is followed by the appropriate paradigm ID from the aot database; the two are separated by a hyphen. This is the information to be predicted by the algorithm for unknown words. After processing the training data, terminal nodes of the suffix trie link to a data structure representing the distribution (relative frequency) of tags for the given suffix.

- A suffix following a vertical bar is attached to the end of the lemma. This represents the available lexical knowledge about the lexical item in an encoded form (n: neuter noun, *: undeclinable, s: singular only).
- Some paradigms are restricted to lemmas ending in a specific suffix. There is a hash mark at the beginning of the suffix of the lemma that is required by the given paradigm ID to be valid. The given paradigm ID is not applicable to words not having that ending. E.g. all lemmas in paradigm 1433 must end in *ъё*.

5 Ranking

The suffix-trie-based ranking algorithm that we used was inspired by the suffix guesser algorithm used in Brants' TnT tagger to estimate the lexical probability of out-of-vocabulary words ([2]). However, that model did not prove to perform well enough in this task. So we modified the model step-by-step until we arrived at a model that turned out to be simpler, yet to perform much better. The paradigms are predicted by assigning a score to each paradigm for each word. Then, the higher this score is for a paradigm tag for a certain word, the more probable it is that the word belongs to that paradigm. We select the top-ranked paradigm to be the predicted inflectional class.

The score for each paradigm is calculated for all suffixes of the word, including the lexical properties, from shortest to longest. For all tags, the rank is calculated iteratively according to Formula 1.

$$rank^{i+1}[tag] = sign \times len\_sfx \times rel\_freq + rank^{i}[tag] \tag{1}$$

where

$sign$	is negative if the suffix is shorter than the minimal suffix required by the given paradigm
len_sfx	is the length of suffix not including lexical properties
rel_freq	is the relative frequency of tag for the suffix
$rank^{i}[tag]$	is divided by len_sfx if $len\_sfx > 1$ is negated if $sign > 0$ and $rank^{i}[tag] < 0$ before calculating $rank^{i+1}[tag]$

The applied ranking score clearly prefers the most frequent paradigm for the longest matching suffix. Some examples for the ranked candidates are shown in Figure 3.

6 Evaluation

Evaluation of the ranking algorithm was performed on different training and test set combinations. In each case, we applied five-fold crossvalidation. In order to see how the performance of the algorithms is affected by the frequency of

```
курба|f [N.f]   [N.f:50]#2.857270 [N.f:175]#0.756756 [N.f:48]#0.293840
                [N.f:105]#0.175658 [N.f:88]#0.098045 [N.f:103]#0.051742
                [N.f:396]#0.03995 [N.f:611]#0.039730 [N.f:69]#0.029693
                [N.f:121]#0.021167

дурака|f  [N.f]    [N.f:88]#4.466005 [N.f:15]#1.341181 [N.f:273]#0.904291
                   [N.f:36]#0.738748 [N.f:50]#0.467147  [N.f:16]#0.443249
                   [N.f:39]#0.300179 [N.f:105]#0.175658 [N.f:96]#0.155983
                   [N.f:103]#0.051742
```

Fig. 3. The ten highest ranked paradigm candidates for the input words курба|f and дурака|f. The candidates are listed sorted by their rank, with the calculated score separated by the # mark for each tag.

the lemmas in the training and test sets, we split the aot lexicon into parts that contained rare words (LT10; not more than 10 occurrences in the Internet corpus; 91,770 words), average words (LT100; between 11 and 100 occurrences; 33,990 words), and frequent words (MT1000; more than 1000 occurrences; 9,650 words). Moreover, we also evaluated performance on a random 20% sample of the lemmas disregarding frequency (RAND; 159,935 words).

We used standard evaluation metrics for measuring performance. *First-best accuracy* measures the ratio of having the correct paradigm ranked at the first place. This reflects the ability of the system to automatically classify new words to paradigms. In addition, the accuracy values for 2^{nd} to 9^{th} ranks were also calculated. *Recall* is the ratio of having the correct paradigm in the set of the first ten highest ranked candidates. Following the metrics used by [7], precision was calculated as *average precision at maximum recall*, i.e. $1/(1+n)$ for each word, where n is the rank of the correct paradigm. This measures the performance of the ranking algorithm. As it might be the case that paradigm prediction is used to aid human classification, this metric reflects the ratio of noise a human must face with when verifying the results. Finally, *f-measure* is the harmonic mean of precision and recall.

We evaluated our algorithm comparing it to two baseline methods. The first one uses Brants' suffix guesser model ([2]) instead of the longest suffix matching method. This model uses a θ factor to combine tag probability estimates for endings of different length in order to get a smoothed estimate. θ is set as the standard deviation of the probabilities of tags. First, the probability distribution for all suffixes is generated from the training set, then it is smoothed by successive abstraction according to Formula 2.

$$P(t|l_{n-i+1},...l_n) = \frac{\hat{P}(t|l_{n-i+1},...l_n) + \theta_i P(t|l_{n-i},...l_n)}{1+\theta_i} \tag{2}$$

for $i = m...0$, with the initial setting $P(t) = \hat{P}$, where

$\hat{P}$ are maximum likelihood estimates from the frequencies in the lexicon

θ_i weights are the standard deviation of the unconditioned maximum likelihood probabilities of the tags in the training set for all i

The other baseline assigns the most frequent paradigm identifier to each word based on its part of speech and the additional features available (e.g. gender, aspect, etc.). The results of these baselines compared to our system are shown in Table 1. As expected, the second baseline, choosing the most frequent tag, has a rather low accuracy, however, our longest suffix method outperforms the first baseline as well. A key difference between the two models is that Brants' model assigns more weight to unconditioned tag distributions and ones conditioned on shorter suffixes than those conditioned on longer ones. This is just the other way round in the longest suffix algorithm.

Table 1. First-best accuracy of paradigm identifiers achieved by the longest suffix match algorithm, Brants' model, and by assigning the most frequent paradigm tag

	Longest suffix	Brants' model	Most frequent tag
MT1000	0.768	0.587	0.410
LT100	0.876	0.593	0.473
LT10	0.887	0.698	0.480
RAND	0.862	0.632	0.466

The tags containing paradigms ID's as well as detailed PoS and subcategorial features define a very sophisticated classification of words. However, some of the features that distinguish two different paradigms are not relevant from the aspect of their inflectional behavior, such as the subtype of a non-inflecting adverb. Moreover, some of these features cannot even be predicted. In many cases, there is stress variation, which does not affect the set of orthographic forms in the paradigm, however, it yields a different paradigm ID. Moreover, some paradigm differences are irrelevant from the point of view of our dictionary lookup task, because they do not affect the set of word forms in the paradigm. The case syncretism differences between animate and inanimate nouns are examples of such differences. To see how the algorithms perform in our original lemmatization task, equivalence classes of paradigms were generated, and a prediction was considered correct if the set of inflected forms generated by the predicted paradigm was identical to the set of word forms generated by the correct paradigm. Of the 2 767 different paradigms, 921 non-unique paradigms could be collapsed into 283 equivalence classes. Table 2 shows the results for each setup, where rows FULL, ID and EQUIP correspond to full tag, paradigm ID, and equivalence class evaluations, respectively. In the rows marked by ID, instead of full tag agreement, which might include hard-to-predict information like that the word is the name of an organization, only the paradigm identifiers were considered. Thus `[N.n._nam:Org.--49]`, and `[N.n.--49]` were considered as equivalent.

Table 2. Results on full tag agreement (FULL), paradigm identifiers (ID) and equivalent paradigm classes (EQUIP). The results are measured by first-best accuracy, precision, recall and f-measure.

	MT1000	LT100	LT10	ALL/MT1000	ALL/LT100	ALL/LT10	RAND
FULL	0.752	0.849	**0.879**	0.759	0.855	0.872	0.848
	0.819	0.903	0.926	0.823	0.910	0.923	0.903
	0.903	0.979	0.991	0.923	0.989	0.994	0.982
	0.859	0.940	**0.958**	0.870	0.948	0.957	0.941
ID	0.768	0.876	**0.887**	0.771	0.872	0.885	0.862
	0.830	0.920	0.934	0.834	0.924	0.933	0.915
	0.905	0.980	0.992	0.926	0.990	0.994	0.983
	0.866	0.949	**0.962**	0.878	0.956	0.962	0.948
EQUIP	0.819	0.889	**0.892**	0.813	0.884	0.890	0.875
	0.869	0.929	0.937	0.866	0.932	0.936	0.924
	0.929	0.984	0.993	0.951	0.993	0.995	0.988
	0.898	0.956	**0.964**	0.907	0.961	0.965	0.955

The three columns on the left show results where the models were trained only on words in the same frequency class they were tested on. The test set was always 20% of the lemmas in the given frequency range. Results in the next four columns were obtained by training the models on the complement of the test set w.r.t. the whole lexicon.

As the numbers show, our system performs best on rare words, while it achieved the worst results on very frequent words. This is not very surprising, as irregular words tend to be frequent words, while rare words have regular inflectional behavior. Correctly predicting the exact paradigm of an unknown personal pronoun or an irregular verb is indeed a rather difficult task. Since our aim was to extend existing morphological lexicons, and such resources already contain the most frequent words of the language, the results obtained for rare words are the ones which are relevant for our task.

Also note that beside similar recall values, precision and first-best accuracy are higher when equivalent paradigms are collapsed. The prediction algorithm works reasonably well for extending resources for tasks that do not require full morphological analysis such as indexing for information retrieval or dictionary lookup.

Table 3 shows the first-best paradigm ID accuracy results for all words, nouns, verbs and adjectives separately. The exact paradigm of verbs and adjectives turned out to be more difficult to guess than that of nouns. The results achieved for adjectives seem to be especially contradictory to the overall performance, which can be explained by the unpredictable behavior of adjectives. Semantic factors and hard-to-predict stress variation affecting paradigmatic classification are explained in the next section of this paper.

Table 3. First-best accuracy of paradigm ID prediction in the case of all types of words, nouns, verbs and adjectives

	ALL	NOUNS	VERBS	ADJECTIVES
MT1000	0.768	0.814	0.702	0.683
LT100	0.876	0.935	0.802	0.772
LT10	0.887	0.968	0.869	0.732
RAND	0.862	0.947	0.848	0.682

7 Error Analysis

The most frequent confusions of the longest suffix algorithm for infrequent words are due to failure to correctly predict

- whether an adjective has synthetic comparative, superlative and/or short predicative forms
- whether a *-ние*-final abstract noun has an alternative *-нье* spelling
- whether a noun has a second genitive (used in partitive constructions) or locative form
- stress in past passive participles of certain verb classes and in short and comparative forms of certain adjectives, or other optional stress variation across the paradigm (this results in an *e*∼*ё* contrast not normally reflected in orthography)
- whether a non-inflecting noun can be interpreted as plural
- whether an imperfective verb has past passive participle forms

Except for stress-related issues and semantically motivated or idiosyncratic defectivity, incorrect forms are very rarely predicted by the algorithm. Humans would probably make similar mistakes for words they do not know, especially if they do not know the meaning of the word either. The system sometimes highlights inconsistencies in the original aot data that even the author of this article, who is not a native or even advanced speaker of Russian, can identify as errors, e.g. that while the name of the energy company *Кубаньэнерго* is categorized as lexically non-plural, the similarly formed *Сахалинэнерго* does not have this property.

When looking at errors the algorithm makes when applied to frequent words, we find that the types of errors are similar. Nevertheless, failure to predict superlatives, comparatives, second genitives or special locative forms is more prevalent for this data, as a much higher proportion of very frequent words have these "irregular" forms.

The most frequent errors of Brants' original suffix guesser algorithm, on the other hand, include absurd errors that would not be made even by beginning learners of Russian. This is due to overemphasizing distributions conditioned on shorter suffixes over those on longer ones. The top-ranked candidate paradigm is often totally inapplicable to words having the ending the given lexical item has, such as the paradigm of *-кий*-final adjectives to *-ный*-final ones (the most frequent error of that algorithm for infrequent words).

8 Conclusion

In this article, we presented and evaluated a suffix-trie-based supervised learning algorithm capable of predicting inflectional paradigms for words based on the ending of their lemma and some basic lexical properties. The algorithm can be used to automatically extend the vocabulary of computational morphologies lacking an independent rule component, which is often the case for resources based on a morphological dictionary. The experiments were demonstrated for Russian, however, with minimal adaptation the tool can be used for any language provided there is a morphological resource available. Moreover, we assumed that a dictionary with some lexical features is also available, thus such features could be used for disambiguating paradigm candidates. The results showed that our method can correctly identify the paradigm of unseen words with an accuracy of about 90%, achieving the best performance on relatively rare words, which are good candidates of being absent in the original lexicon. For rare nouns, the paradigm identification accuracy is 96.8%.

We found that assigning more weight to distributions conditioned on longer suffixes than on shorter ones yields much better prediction performance, not only in terms of the number of exact predicted paradigm matches, but especially when taking into account what sorts of errors the system makes. While the baseline suffix guesser algorithm often proposes paradigms inapplicable to the given lexical item, our algorithm makes errors that arise due to the lack of lexical semantic information. Humans would make similar errors in similar situations.

Acknowledgement. The author of this article would like to thank Borbála Siklósi for her help, especially in evaluation.

References

1. Ahlberg, M., Forsberg, M., Hulden, M.: Semi-supervised learning of morphological paradigms and lexicons. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, April 26-30, pp. 569–578 (2014), `http://aclweb.org/anthology//E/E14/E14-1060.pdf`
2. Brants, T.: Tnt - a statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing, ANLP 2000. Seattle, WA (2000)
3. Dreyer, M., Eisner, J.: Discovering morphological paradigms from plain text using a dirichlet process mixture model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 616–627. Association for Computational Linguistics, Stroudsburg (2011)
4. Forsberg, M., Hammarström, H., Ranta, A.: Morphological lexicon extraction from raw text data. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 488–499. Springer, Heidelberg (2006)
5. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Comput. Linguist. 27(2), 153–198 (2001)
6. Hammarström, H., Borin, L.: Unsupervised learning of morphology. Comput. Linguist. 37(2), 309–350 (2011)

7. Linden, K.: Entry generation by analogy encoding new words for morphological lexicons. Journal Northern European Journal of Language Technology, 1–25 (2009)
8. Monson, C., Carbonell, J., Lavie, A., Levin, L.: Paramor: Finding paradigms across morphology. In: Peters, C., Jijkoun, V., Mandl, T. (eds.) CLEF. LNCS, vol. 5152, pp. 900–907. Springer, Heidelberg (2007)
9. Nakov, P., Bonev, Y., Angelova, G., Gius, E., von Hahn, W.: Guessing morphological classes of unknown German nouns. In: Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R. (eds.) RANLP. Current Issues in Linguistic Theory (CILT), vol. 260, pp. 347–356. John Benjamins, Amsterdam (2003)
10. Novák, A.: What is good Humor like? [Milyen a jó Humor?]. In: I. Magyar Számítógépes Nyelvészeti Konferencia, pp. 138–144. SZTE, Szeged (2003)
11. Oliver, A., Tadic, M.: Enlarging the croatian morphological lexicon by automatic lexical acquisition from raw corpora. In: LREC. European Language Resources Association (2004)
12. Prószéky, G., Kis, B.: A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL 1999, pp. 261–268. Association for Computational Linguistics, Stroudsburg (1999)
13. Sokirko, A.V.: Morphological modules at the site www.aot.ru. In: Dialog 2004 (2004)
14. Šnajder, J.: Models for predicting the inflectional paradigm of croatian words. In: Slovenščina 2.0, pp. 1–34 (2013)
15. Wicentowski, R.: Modeling and learning multilingual inflectional morphology in a minimally supervised framework. Tech. rep. (2002)
16. Zaliznyak, A.A.: Russian grammatical dictionary – Inflection. Russkij Jazyk, Moskva (1980)

To Split or Not, and If so, Where? Theoretical and Empirical Aspects of Unsupervised Morphological Segmentation

Amit Kirschenbaum

Natural Language Processing Group,
Leipzig University, Germany
amit@informatik.uni-leipzig.de

Abstract. The purpose of this paper is twofold: First, it offers an overview of challenges encountered by unsupervised, knowledge free methods when analysing language data (with focus on morphology). Second, it presents a system for unsupervised morphological segmentation comprising two complementary methods that can handle a broad range of morphological processes. The first method collects words which share distributional and form similarity and applies Multiple Sequence Alignment to derive segmentation of these words. The second method then analyses less frequent words utilizing the segmentation results of the first method. The challenges presented in the theoretical part are demonstrated exemplarily on the workings and output of the introduced unsupervised system and accompanied by suggestions how to address them in future works.

1 Introduction

Unsupervised, knowledge free approaches analyse raw, unannotated data without any previous knowledge about the language they are applied on. In the context of morphology, which is at the focus of the present study, this can comprise various tasks like paradigm extraction, detection of related sets of words, or morphological segmentation. In the last half of a century of research, several dozens of algorithms addressing these tasks have been developed (see [15] for an overview), with the result that "high accuracy by ULM systems is presently only achievable if the language has small amounts of one-slot concatenative morphology" [15, p.335].

In this paper, we introduce a two-method system performing morphological segmentation, i.e. splitting word forms of a given language into their basic units carrying meaning - the morphemes. The first method utilizes Multiple Sequence Alignment (MSA). The approach has its origin in bioinformatics, where it is used to align sequences of DNA, RNA, proteins, etc. MSA is used to identify conserved regions that play functional or structural roles in collections of biosequences that are assumed to be related. The most common way to align multiple sequences is progressive alignment. In this approach, the most similar pair of sequences is aligned first, and then more distant sequences are added progressively .[23]. The method has the important characteristics that it can detect discontinuous patterns which equips it with the potential to successfully handle more complex structures with non-concatenative properties.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 139–150, 2015.
DOI: 10.1007/978-3-319-18111-0_11

2 Theoretical Considerations

In this section we briefly survey the challenges that unsupervised morphological segmentation methods face when coping (a) with the subject of the task, the language; (b) with the methodological issues and implementations; and (c) with the evaluation of the segmentation results. The issues discussed in this section are then instantiated in the empirical part in Section 3.

2.1 Language Challenges

The utmost aim of unsupervised, knowledge-free algorithms is to analyse any given language. However, world languages display a variety of morphological processes and phenomena, so that designing one system that can successfully handle them is a very ambitious challenge. Depending on the predominance of particular morphological process, a language can be classified as agglutinative, inflectional (with the subclass of introflectional), isolative, or polysynthetic. In general, languages are of mixed morphological types, so that only a method that can principally handle any language type can also handle any morphological process that might occur in a language.

Each morphological process poses a different challenge to unsupervised segmentation. Whereas in polysynthetic languages the task of morphological segmentation overlaps to a great deal with word segmentation, strictly isolative languages like Chinese are not relevant for morphological segmentation, since they do not contain any morphemes that could be split. Especially agglutination and inflection thus challenge the unsupervised language analysis, each of them having properties that either complicate, or alleviate the morphological segmentation task.

Agglutinative languages exhibit clear boundaries between morphemes which may be advantageous for unsupervised segmentation. On the other hand, several affixes are often agglutinated to one stem, which lowers the frequency of occurrence of the same word forms in the corpus, and thus has negative implications for finding contextually similar words. As stated earlier, present unsupervised methods perform best on one slot linear morphology.

Inflectional languages have typically smaller number of affixes which can be added to a root, due to accumulation of several functions on one affix and frequent syncretism among inflected forms. However, stem alternants and affix allomorphy are also more frequent phenomenon in these languages, which makes it more difficult to determine where the segmentation point should be and whether formally similar word forms are also close morphologically. A special case are introflective languages with so called non-concatenative morphology like e.g. Hebrew or Arabic, which often contain discontinouos morphemes both as stems (e.g. root consonants) and inflections (introflection).

As mentioned above, phenomena that characterize one language type are typically present also in languages assigned to another predominate type. For example, German, the languages on which we demonstrate our method and the challenges to the unsupervised segmentation, is usually classified as a an inflectional language. In addition to the inflection ,however, German features other morphological processes. Similarly to English, grammatical relations between words can be expressed both through inflectional suffixes, e.g., *Werthers Leiden* 'Werther's suffering', or through prepositions, which is

typical for isolative languages, *das Leiden von Werther* 'the suffering of Werther'. There are also agglutinative phenomena like in the word *Kind-er-n* child-PL-DAT, where each affix carries only one grammatical function; -er: plural -n: dative. Moreover, German compounding that can result in very complex words is a property that links German with polysynthetic languages. German irregular verbs (e.g. *singen* - *sangen* , sing.PRS-PL-sing.PST-PL) represent an example of introflection in this language, since grammatical category (here tense) changes depending on the vowel inserted into the discontinuous stem morpheme. Circumfixation, another non-concatenative phenomenon, is present in German as well, e.g. in *ge-lauf-en* PTCP-run-PTCP, where the circumfix ge-en marks the participle form.

Methods that aim at an adequate and complete analysis of any given language should handle successfully all morphological structures. However, as stated also by [15, p. 310, p.332], unsupervised methods tend to exhibit an implicit or explicit bias towards a certain kind of languages. Many unsupervised algorithms for morphological analysis assume concatenative morphology, and design their methods and data structures in ways that are suited to describe such phenomena (e.g. [25,26,10,4]).

In the empirical part of this paper we present a method that tries to avoid such bias by applying MSA that had been shown to be able to deal with both concatenative and non-concatenative morphology. The method differs from the previous approaches in that it thrives to be language independent (cf. [24] with strong language specific bias for Arabic, or [8] for stem variation in German) and that it focuses on morphological segmentation (cf. [3] who addresses morphology induction with very low F-scores (below 0.10) for the non-concatenative Arabic). Our approach is similar to features-and-classes method of [7] in its attempt to design a general system that can address the whole range of morphological phenomena in any given language.

In addition to the typological division of languages and morphological processes outlined above, another relevant distinction is between inflectional and derivational morphology. Inflection modifies a word to create new word forms that express different grammatical categories (e.g. number, case, tense, etc.). Regular inflection is typically very productive and therefore easier to detect automatically. Derivation, on the other hand, is a less productive process that creates new lexemes out of existing bases. It involves change in the core meaning and often also change in the word class. Lower productivity and the involved change of meaning are two aspects of derivational morphemes that make them more difficult to detect in an unsupervised manner. Some methods therefore resign on this aim completely, for example [25,26]. [13,14], on the other hand, decompose word forms into stems and suffixes, using the Minimum Description Length (MDL) principle, and groups them into signatures, each is a structure that denote a set of stems that can co-occur with a set of affixes. This method handles inflectional and derivational morphology without making a clear distinction between them. However, it is, again, restricted to concatenative morphology. The qualitative analysis of the data produced by the system described below shows that our approach can segment more complex inflectional and derivational morphemes (see also Table 3).

2.2 Algorithmical and Resources Challenges

Morpheme is the smallest unit of language that carries meaning, i.e. it is a particular form bound to a particular meaning or function. The mapping between them is not always one-to-one (cf. allomorphy) and the form does not need to be linear (see above). However, given this twofold nature of a morpheme it is obvious that unsupervised methods ignoring the meaning/function aspect of morphemes are doomed to fail: Languages abound of strings that formally overlap but do not have the status of a morpheme.

A particular challenge related to unsupervised, knowledge free morphological analysis thus is how to approximate the meaning. One possibility that is exploited also by the method presented in this paper is to use context. In line with the Distributional Hypothesis [16] words that appear in the same context are semantically similar. However, in order to compute distributional similarity reliably, a sufficient amount of contexts in which the word forms occur is needed. Consequently, most current unsupervised methods are very resource intensive, in that they require corpora of a very large size. The need for huge corpora is so acute that even a corpus of 500000 running forms is considered small by some methods [9]. For resource rich languages, large corpora and possibly also training sets for supervised algorithms are not a problematic issue. However, a field that could profit substantially from unsupervised language analysis are resource poor and/or endangered languages for which sometimes only small, unannotated corpora are available. Clearly, such settings do not provide enough input for context based methods to reliably detect distributionally and thus semantically similar words.

In parallel to the typological problem described in the previous section, where a failure of a method to deal with a particular language type also means a failure to deal with some morphological features in a given language (since language types are mixed), the data sparseness problem described above does not affect the performance of the algorithms only on small corpora, but also on corpus low frequent word forms in a large corpus. The system presented in this paper attempts to find an solution that delivers adequate analysis also for corpus low frequent words.

2.3 Evaluation Challenges

The most widely used evaluation method is the automatic comparison of the computed results against adequate linguistic reference, i.e. the gold standard. The alternative, a direct manual evaluation by the language expert(s) is both time and work consuming and unrealistic in many settings. Depending on the task (and also the available gold standard), various evaluation methods have been proposed. Their overview can be found in [28]. In the area of morphological segmentation, the most straightforward evaluation is the calculation of how well the automatically detected segmentation boundaries correspond to the morpheme boundaries in the gold standard (e.g., [6,20]).

The first challenge to the automatic segmentation evaluation is the availability of adequate reference analyses, the gold standard, which typically exist only for resource rich languages. The MorphoChallenge competition (since 2005) provided gold-standard evaluation data for English, German, Finnish, Arabic, and Turkish and for task-based Information Retrieval evaluation data for English, German, and Finnish. Though the MorphoChallenge series without doubt greatly supported the research in unsupervised

morphological analysis and contributed to the evaluation standardization and comparability, it is also evident that given the diversity of world languages, the offered sample cannot be viewed as representative (see also [15, p.335]). It has been acknowledged (see e.g. [28]) that unsupervised methods cannot come with results that exactly correspond to those designed by linguists. However, it is not only the limitations of the unsupervised methods but also of the gold standards quality that challenges the evaluation. Their reference analyses often do not correspond to complete linguistic analyses. One typical problem are derivational affixes. Whereas most gold standards for morphological segmentation contain all or close to all boundaries separating inflectional affixes, segmentation of derivational affixes can be missing or incomplete. Consequently, a method that is able to detect boundaries also between roots and derivational affixes can be disadvantaged compared to a method focusing solely on inflection when compared to such a gold standard. Even more intriguing is the problem of introflection. Changes on stems are typically not grasped by the gold standards. In German, word forms like singen and sangen are analysed only with respect to their inflectional suffix, i.e. *sing-en* and *sang-en*. A method that correctly identifies the vowel change within a discontinouos stem and performs the analysis as *s-i-ng-en* and *s-a-ng-en* is penalized because the additional splits are scored as incorrect (cf. Table 3).

In addition to the challenges related directly to the gold standards, there are challenges related to how the evaluation is performed by individual authors. Low frequent word forms (which can mean up to ten occurrences in this context, cf. [26], or all other words except the most frequent ones, cf. [10,8], are often excluded from either the analysis itself, or from the evaluation, or from both. Moreover, the unsupervised algorithms often perform poorly on the most frequent words. As an example, the algorithm presented in [2] delivers worse results without the trimming of the word forms with a corpus frequency above 0.01% of the total token count. The authors argue that these tend to be function words that are of little interest for morphological analysis. The evaluation of the algorithms thus often differs not only with respect to which gold standard is used and what its properties are, but also with respect to the corpus portion that had been analysed and reported.

3 Empirical Instantiations

When designing the unsupervised segmentation system described below, we carefully considered the challenges outlined in the theoretical part. Given the two-sided nature of a morpheme, we decide for a system that takes into account not only morpheme's formal aspects, but also its meaning/function. In order to approximate meaning, we decided to exploit the distributional hypothesis following the thesis that words that occur in similar contexts have also similar meaning/function. Since such approach can be successfully applied only on word forms with sufficient corpus frequency, we designed a system comprising of two methods: One is using context similarity directly, the other indirectly through utilizing the results of the first method.

As already mentioned, various morphological processes are involved in word form construction. In order to capture this morphological variance, we based the system on MSA that has the potential to address any of the morphological processes. There are

only few biologically inspired methods reported for the task unsupervised of morphological segmentation: [11,18] employ genetic algorithm to obtain the optimal solution within the space of all possible word segmentation into stems and suffixes. These methods use fitness functions which can be viewed as simplified forms of MDL: They seek the absolute minimum of characters [18] or elements [11] in the sets of stems and suffixes, that describe the language, rather than using the information-theoretic criterion, which is based on conditional probabilities, as in [13]. [12] enhance the above idea to detect derivational paradigms, with a strategy that takes into account a property of the language, i.e., the fact that different stems may be combined with the same set of suffixes. The method first generate hypothesized stems and suffixes from a list of words; for each stem all possible paradigms are detected, i.e., the sets of suffixes repeated for that stem. Binary chromosomes represent solutions, where each gene (index in the chromosome) encodes a hypothesized stem or suffix. The genetic algorithm is then applied to an initial population of randomly generated chromosomes. As can be observed also here, these methods focus their efforts on analyzing concatenative morphology, identifying suffixation patterns. Previous work utilized MSA for morphological segmentation, but handled this task differently. [27] aligns orthographically similar words, and uses third-party analysis [21] as a guide, to search for a set of segmentation columns. It determines its segmentation decisions by maximizing the F-score against the analysis of the third-party system. A more closely related work is presented in [19], where semantically related words are used to identify patterns which are assumed to be morphemes. However, similarly to other approaches relying on contextual information, also this approach analyses only those word forms in the corpus that appear with sufficient frequency.

4 The First Method M_1

M_1 is based on the idea that morphologically related words are both formally and semantically similar. We assume that recurring patterns within such words correspond to the morphological relations among them. We identify overlapping patterns within such word with the assistance of MSA, and insert predicted morpheme boundaries in those words accordingly.

In the first step, distributionally and othographically similar word forms are extracted and clustered into sets of presumably morphologically related word forms according to a method described in [19]. In the second step, patterns are extracted from the sets using MSA. Word forms in these sets are aligned using progressive alignment: First, the two most similar sequences are aligned and then less similar ones are added in a cumulative way to construct the final alignment. In the context of morphological segmentation, selected sets of distributionally and orthographically similar words are treated as sequences that are to be aligned. The first sequence of the alignment is the input word, and the similarity criterion means, in this case, similarity of a related word to the input word. The alignment method is based on the one appears in the BioJava package [17], modified for our purpose. Table 1 demonstrates the alignment set for the word *umgedreht*. The "-" signs indicate gaps which are inserted during the alignment process to unify the lengths of the sequences.

Table 1. An example for an alignment of the word form *umgedreht* and its related word forms

umgedreht---
abgedreht---
um--dreht---
um--drehte--
...
umzudrehe--n
umgesied-elt
um--drehe--n
...

Next, M_1 compares the aligned sequences to find a pattern which matches the alignment best: Identical fragments are extracted from pairs of aligned sequences, and are considered as candidate patterns for this alignment. Each candidate pattern is stored with the number of corresponding sequences with which it matches. In the example above the pair of aligned sequences constructed from the word forms *umgedreht* and *abgedreht* generates the candidate pattern `-gedreht`, whereas the pair of aligned sequences consisting of the word forms *umgedreht* and *umdrehen* generates the candidate pattern `um-dreh` that contributes to the correct and complete analysis of the word form `um-ge-dreh-t`. Each candidate pattern $pattern_i$, of the alignment set is given a score which balances between the relative frequency of the pattern in the given alignment and the length of the pattern, and is calculated as follows:

$$score(pattern_i) = \frac{2}{\frac{count(pattern_i)}{\sum_j count(pattern_j)} \log(size) + \frac{1}{length(pattern_i)}} \,. \qquad (1)$$

Here, $count(pattern_i)$ is the number of aligned sequences which match this candidate pattern, $size$ is the number of sequences participating in this alignment and $length(pattern_i)$ is the number of characters which $pattern_i$ consists of. Patterns are ranked based on their scores, and the pattern that got the highest score is selected as the one that describes best the members of the alignment set, and the word forms which formed that alignment set are segmented accordingly. This candidate segmentation for each word from is recorded along with the respective score. A word form may be a member of several alignment sets since it can match the condition of both distributional and form similarity for more than one input word form, and it can be an input word form itself. Therefore a word form can have several candidate segmentations from the different alignments, some of which can be identical. To select the best segmentation for each word form, the scores recorded for each candidate segmentation are tallied, and a ranked list of possible segmentations for each word form is constructed based on those scores.

The method was applied on a corpus of three million German sentences obtained from the Wortschatz collection[1] at the University of Leipzig (Germany).[2] Overall, out

[1] `http://corpora.inforamtik.uni-leipzig.de`

[2] These sentences were used in MorphoChallenge comptetions.

of 1294071 M_1 was able to analyse 196852 (15.2%) word forms, of which 58213 were found in CELEX [1] which is used as a gold standard.

Since M_1 returns a ranked list of segmentation options for each word, we report the top-1, -2 and -5 results. The results are summarized in Table 2 and show the precision (P), recall (R), and F-measure (F) values for each of these cases.

Table 2. Results for M_1

Top-n	P	R	F
1	0.48	0.46	0.47
2	0.57	0.56	0.56
5	0.60	0.59	0.59
Baseline	0.22	0.49	0.30
Morf.	0.60	0.43	0.50

The results were compared to a baseline which assigns segmentation points to the input word forms randomly. Our method performs well above the baseline which achieved F-score of 0.30. The results were also compared to Morfessor [5] which represents the current state of the art. The comparison shows that the presented method achieves good results that for top-1 are close to the state of the art with a potential for further improvement as indicated by the top-2 and top-5 results. It should be pointed out that the top-2 and top-5 analyses hypotheses do not present mutually exclusive solutions. Instead, they typically comprise several solutions that differ in how close they are to the complete linguistic analysis.

A qualitative analysis of the data confirms that the method can deal with different morphological processes that are sometimes not grasped by Morfessor, or by the gold standard, or by both. Table 3 gives overview of such examples.

Table 3. Examples of morphological processes analysed by M_1 with their corresponding F-scores when compared to the gold standard: D - derivation, I - inflection, /Intr/ - introflection, Cfix - circumfixation, Aggl - agglutination, P/C - polysynthesis/compunding

complete analysis	Process	M_1	Morfessor	gold standard
alban-isch	D	alban-isch (1.0)	albanisch (0.0)	alban-isch
hebrä-isch	D	hebrä-isch (0.0)	hebräisch (1.0)	hebräisch
Raps-öl	P/C	Raps-öl(1.0)	Rapsöl (0.0)	Raps-öl
ge-wähl-t	Cfix	ge-wähl-t (1.0)	gewählt (0.0)	ge-wähl-t
k/a/nn	/Intr/	k-a-nn (0.0)	kann (1.0)	kann
zu-ge-ruf-en	D+Cfix	zu-ge-ruf-en (1.0)	zu-gerufen (0.3)	zu-ge-ruf-en
Stief-kind-er-n	D+Aggl	Stiefkind-er-n (0.5)	Stiefkind-er-n (0.5)	Stief-kind-ern

5 The Second Method M_2

M_1 analyses word forms for which both distributionally and formally similar word forms could be retrieved. This approach requires enough characteristic contexts to create a reliable contextual representation of a word from. Words forms with low frequency can therefore achieve only inaccurate representations and the degree of semantic relatedness among them is typically rather weak. Consequently, the probability that a morphologically related words would be among them is also lower.

Method M_2 presented in this section was designed to handle the word forms with such context constraints. In order to avoid an approach that would take into account solely the form aspects of morphemes, the meaning/function based segmentation results from the first method were utilized to compute the segmentation of the so far unanalysed words.

For each unanalysed input word (focus word), M_2 first collects a list $\{w_k\}$ of previously analysed words that are formally similar to it. Form similarity between the focus word and w_k is calculated as $1 - d_k$, where d_k is Needlman-Wunsch distance [22] with affine gap penalties, normalized to the range [0,1]. $\{seg_{kl}\}_{l=1}^{5}$ is formed by retrieving the top-5 analyses for each w_k. The focus word form is then compared to each seg_{kl} to find matching segments and M_2 generates segmentation hypotheses h_{kl} for the focus word.

In our experiments with this method we considered several parameters. The implementation with the so far best results included (a) the degree of form similarity between the focus word and each of the words in $\{w_k\}$, as described above; (b) the coverage of segments found in the focus word with respect of to segments in seg_{kl}, that is, the ratio of the segments found in the focus word and the segments of seg_{kl}. Let $\{s_m\}$ be the set of segments of a given segmentation hypothesis seg_{kl} of w_k, and let $\{s_n\} \subset \{s_m\}$ be the set of segments discovered in the focus word, then the ratio between the sizes of these two sets is used as a measure of segments coverage. The score for a single segmentation hypothesis of a focus word then is:

$$score(h_{kl}) = \frac{|\{s_n\}|}{|\{s_m\}|} \times (1 - d_k) \ . \tag{2}$$

The results of this experiment are presented in Table 4.

Table 4. Results for M_2

Top-n	P	R	F
1	0.49	0.44	0.46
2	0.62	0.56	0.59
5	0.68	0.63	0.65
Baseline	0.21	0.50	0.29
Morf.	0.74	0.52	0.61

In our future work, we want to include other parameters in our experiments and investigate the potentials of various parameter combinations to optimize both the top-1 and top-n results. One possibility would be including a "segment validity parameter" that can be computed as a function of segment frequency among different seg_{kl}s.

6 Overall Results and Evaluation

The evaluation of the whole corpus including words analyzed by both M_1 and M_2 is presented in Table 5.

Table 5. Results for the whole corpus

Top-n	P	R	F
1	0.48	0.46	0.47
2	0.59	0.56	0.57
5	0.64	0.61	0.63
Morf.	0.67	0.48	0.56

The results show that the system delivers useful results when applied on data with different degree of sparseness. Though top-1 analyses are still subject to improvement, the top-5 results show that the method can achieve promising results. The qualitative analysis of the results confirms (see also Table 3) that the evaluation is negatively affected by some properties inherent to the gold standard. Introflective aspects of German that include e.g. stem vowel changes in the conjugation of irregular or auxiliary verbs or in pluralization of nouns are not captured by the gold standard but are often analysed by our system. Consequently, segmentation boundaries that are actually correct are scored as false positives due to their absence in the gold standard. Similarly, not all derivation morphemes are segmented in the gold standard either (see the examples Hebräisch and Albanisch in Table 3). Due to these deficits in the gold standard, the qualities of the presented system that distinguish it from other approaches sometimes fall short compared to methods that deliver results more conform to the (imperfect) gold standard. It can be however assumed that its actual performance is higher than the evaluation reveals.

It would be probably unrealistic to expect that any existing gold standard of sufficient size for the evaluation of unsupervised methods could contain a complete and perfect annotation. The more important it then seems for the comparability of the results that the specifics of the gold standard used for evaluation would be at least partly as well described as the evaluation method itself. It would be further useful if the quantitative analysis was accompanied by at least a brief qualitative analysis of the method's output, so that the reader can get insights into its scope. As an example, compared to methods that are biased or designed to perform (only) the segmentation of inflectional affixes, systems such as the one presented in this paper may not achieve equally high results on the regular and frequent inflectional phenomena, but might be able to address a larger scope of morphologically different processes.

7 Conclusions

In the first part, the paper surveyed the theoretical context and challenges of unsupervised, knowledge free morphological segmentation. In the second part it described a system grounded in the theoretical considerations and presented its result on German. The method utilizes MSA to analyse formally and distributionally similar words, and uses these context based results to assist the analysis of less frequent words. The results show that the method can handle a broad range of morphological processes in a quality close to the present state of the art approaches and has potential for further improvement.

References

1. Baayen, R.H., Piepenbrock, R., Gulikers, L.: The CELEX lexical database (release 2). CD-ROM (1995)
2. Baroni, M., Matiasek, J., Trost, H.: Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In: Proceedings of the ACL 2002 Workshop on Morphological and Phonological Learning, pp. 48–57. Association for Computational Linguistics (July 2002)
3. Bernhard, D.: Morphonet: Exploring the use of community structure for unsupervised morpheme analysis. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 598–608. Springer, Heidelberg (2010)
4. Bordag, S.: Unsupervised and knowledge-free morpheme segmentation and analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 881–891. Springer, Heidelberg (2008)
5. Creutz, M., Lagus, K.: Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. Tech. Rep. Report A81, Helsinki University of Technology (March 2005)
6. Creutz, M., Lindén, K.: Morpheme segmentation gold standards for finnish and english. Publications in Computer and Information Science, Report A 77 (2004)
7. De Pauw, G., Wagacha, P.W.: Bootstrapping morphological analysis of gıkuyu using unsupervised maximum entropy learning. In: Proceedings of the Eighth Annual Conference of the International Speech Communication Association (2007)
8. Demberg, V.: A language-independent unsupervised model for morphological segmentation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 920–927. Association for Computational Linguistics (June 2007)
9. 1Fisher, D., Riloff, E.: Applying statistical methods to small corpora: Benefitting from a limited domain. In: Probabilistic Approaches to Natural Language, a AAAI Fall Symposium. pp. 47–53, technical Report FS-92-04 (1992)
10. Freitag, D.: Morphology induction from term clusters. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, CONLL 2005, pp. 128–135. Association for Computational Linguistics (2005)
11. Gelbukh, A.F., Alexandrov, M., Han, S.: Detecting Inflection Patterns in Natural Language By Minimization of Morphological Model. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 432–438. Springer, Heidelberg (2004)

12. Gelbukh, A.F., Sidorov, G., Lara-Reyes, D., Chanona-Hernández, L.: Division of spanish words into morphemes with a genetic algorithm. In: Kapetanios, E., Sugumaran, V., Spiliopoulou, M. (eds.) NLDB 2008. LNCS, vol. 5039, pp. 19–26. Springer, Heidelberg (2008)
13. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27(2), 153–198 (2001)
14. Goldsmith, J.: An algorithm for the unsupervised learning of morphology. Natural Language Engineering 12(04), 353–371 (2006)
15. Hammarström, H., Borin, L.: Unsupervised Learning of Morphology. Computational Linguistics 37(2), 309–350 (2011)
16. Harris, Z.S.: Distributional Structure. In: Fodor, J.A., Katz, J.J. (eds.) The Structure of Language: Readings in the Philosophy of Language, pp. 33–46. Prentice-Hall (1964)
17. Holland, R.C.G., Down, T.A., Pocock, M.R., Prlic, A., Huen, D., James, K., Foisy, S., Dräger, A., Yates, A., Heuer, M., Schreiber, M.J.: BioJava: an open-source framework for bioinformatics. Bioinformatics 24(18), 2096–2097 (2008)
18. Kazakov, D.: Unsupervised Learning of Naïve Morphology with Genetic Algorithms. In: Daelemans, W., van den Bosch, A., Weijters, A. (eds.) Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks, pp. 105–112 (1997)
19. Kirschenbaum, A.: Unsupervised segmentation for different types of morphological processes using multiple sequence alignment. In: Dediu, A.-H., Martín-Vide, C., Mitkov, R., Truthe, B. (eds.) SLSP 2013. LNCS, vol. 7978, pp. 152–163. Springer, Heidelberg (2013)
20. Kurimo, M., Creutz, M., Varjokallio, M., Arisoy, E., Saraclar, M.: Unsupervised segmentation of words into morphemes-challenge 2005: An introduction and evaluation report. In: Proceedings of the PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes (2006)
21. Monson, C., Hollingshead, K., Roark, B.: Probabilistic ParaMor. In: Working Notes for the CLEF 2009 Workshop (2009)
22. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3), 443–453 (1970)
23. Notredame, C.: Recent progresses in multiple sequence alignment: a survey. Pharmacogenomics 3(1) (2002)
24. Rodrigues, P., Ćavar, D.: Learning arabic morphology using statistical constraint-satisfaction models. In: Benmamoun, E. (ed.) Perspectives on Arabic Linguistics XIX, pp. 63–75. John Benjamins (2007)
25. Schone, P., Jurafsky, D.: Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In: Proceedings of the 4th Conference on Computational Natural Language Learning, vol. 7, pp. 67–72. Association for Computational Linguistics (2000)
26. Schone, P., Jurafsky, D.: Knowledge-free induction of inflectional morphologies. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, NAACL 2001, Association for Computational Linguistics (2001)
27. Tchoukalov, T., Monson, C., Roark, B.: Morphological Analysis by Multiple Sequence Alignment. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 666–673. Springer, Heidelberg (2010)
28. Virpioja, S., Turunen, V.T., Spiegler, S., Kohonen, O., Kurimo, M.: Empirical comparison of evaluation methods for unsupervised learning of morphology. Traitement Automatique des Langues 52(2), 45–90 (2011)

Data-Driven Morphological Analysis and Disambiguation for Kazakh

Olzhas Makhambetov, Aibek Makazhanov, Islam Sabyrgaliyev, and Zhandos Yessenbayev

Nazarbayev University Research and Innovation System
Astana, Kazakhstan
{omakhambetov,aibek.makazhanov,islam.sabyrgaliyev,
zhyessenbayev}@nu.edu.kz

Abstract. We propose a method for morphological analysis and disambiguation for Kazakh language that accounts for both inflectional and derivational morphology, including not fully productive derivation. The method is data-driven and does not require manually generated rules. We leverage so called "transition chains" that help pruning false segmentations, while keeping correct ones. At the disambiguation step we use a standard HMM-based approach. Evaluating our method against open source solutions on several data sets, we show that it achieves better or on par performance. We also provide an extensive error analysis that sheds light on common problems of the morphological disambiguation of the language.

1 Introduction

Morphological analysis (MA) and disambiguation (MD) are crucial steps in automated processing of any language, and, in the case of agglutinative languages (ALs), it is hard to overestimate their importance. Agglutination causes words to acquire complex meanings, effectively transforming them into whole phrases. To "expand" such words into phrases (e.g., to translate into another language) one needs to perform MA. Usually there is an ambiguity, i.e. a word can have multiple analyses and meanings. Thus, MA is typically followed by MD, in order to choose the analysis that best fits the context. In this paper we describe a way of applying such a two-step processing to Kazakh, an agglutinative Turkic language.

Before continuing further, we would like to define key terms with which we are going to operate throughout the paper. Given an analysis [*alma*-N-POSS.3SG-ACC] we define segmentation as [*alma*_N-*sy*_POSS.3SG-*n*_ACC], i.e. an analysis with preserved surface form. Given this analysis-segmentation pair, we will refer to the following as: *alma* – root; *alma*-N – pos-labeled root; N – POS tag; *n*_ACC – morpheme; ACC – morpheme tag; *sy*_POSS.3SG-*n*_ACC – morpheme chain; POSS.3SG-ACC – tag chain.

To give readers an intuition of a kind of difficulties that may arise in morphological disambiguation of ALs, let us consider the example given in Table 1. The table contains possible analyses of a Kazakh word *almasyn*. Each row of the table contains analyses that can be stemmed from a particular root. As we can see the main source of ambiguity is the fact that the word can be stemmed from six different roots. Additionally, for every

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 151–163, 2015.
DOI: 10.1007/978-3-319-18111-0_12

Table 1. Possible analyses of a Kazakh word [*almasyn*]

#	Analysis	Root-POS	English translation
1,2	al-VB-NEG-PTCP-*POSS.3SG/PL*-ACC	al-VB, *take*	*his/her/their* not taking
3,4	al-VB-NEG-OPT-*3SG/PL*		let *him/her/them* not take
5,6	al-VAUX-NEG-PTCP-*POSS.3SG/PL*-ACC	al-VAUX, *be able*	*his/her/their* not being able
7,8	al-VAUX-NEG-OPT-*3SG/PL*		let *him/her/them* not be able
9,10	alma-N-*POSS.3SG/PL*-ACC	alma-N, *apple*	(ate) *his/her/their* apple
11,12	Alma-NP-*POSS.3SG/PL*-ACC	Alma-NP, *female name*	(saw) *his/her/their* Alma
13,14	almas-N-*POSS.3SG/PL*-ACC	almas-N, *sword*	(swung) *his/her/their* sword
15,16	Almas-NP-*POSS.3SG/PL*-ACC	Almas-NP, *male name*	(saw) *his/her/their* Almas

nominal root (N, NP) or stem (PTCP) the 3rd person possession marker can be either singular or plural[1], which results in 12 analyses. Similarly, optative mood constructions, that stem from a main and an auxiliary verb *al*, agree with either singular or plural 3rd person subject (1st and 2nd rows, analyses 3,4 and 7,8). The ambiguity in agreements gives four additional analyses, yielding a total of 16 possibilities[2].

Traditionally the MA problem has been approached by building finite state transducers (FST) [1–3] based on a formal description of the morphology [4] of a language. These approaches are attractive due to their efficiency and precision. Although there are open source tools that effectively implement transducers [5,6], building a grammar that accounts for most aspects of the language is a quite challenging endeavor. Derivational morphology is a good example of an aspect frequently avoided (at least for Kazakh) by formal method-based approaches.

Derivational morphology in Kazakh is largely non-productive, i.e. almost all derivational morphemes attach preferentially. For example, a verb *basta* (*begin, lead* – literally *make head*) can be said to consist of a noun root *bas* (*head*) and a noun-verb derivative *-TA*[3]. However, the very same *-TA* does not attach to a noun *shash* (*hair*) to make (by the same logic) something like *grow one's hair*. Now, imagine building a transducer

[1] In general, in Kazakh any nominal marked by a 3rd person possession marker yields at least two analyses. The same goes for any finite verb that agrees with a 3rd person subject. As we will show later, combined with ambiguity in present and future indefinite markers, these cases cause disambiguation methods most of the troubles.

[2] As far as ambiguity goes, we would like to emphasize two very interesting cases in our example. Notice how analyses pairs 9,10 and 13,14 have similar inflectional paradigms and differ only in roots. Disambiguation of such cases goes beyond distinguishing between morpho-syntactic patterns. It involves semantics. In other words, a successful disambiguation method ought to know that normally swords cannot be eaten and apples cannot cut flesh. Same goes for analyses pairs 11,12 and 15,16: here a choice is between feminine and masculine personal names. While it seems that gender information associated with names might help, it may, actually, add to the ambiguity, because there is no grammatical gender in Kazakh.

[3] Capitalization represents a group of allomorphs that match a regular expression [tdl][ae].

Table 2. Characteristics of the morphological sub-corpus of the KLC

Characteristic	Value
words, total	584 839
words, unique	77 060
analyses, unique	85 806
analyses per word, avg.	1.11 ± 0.38
analyses per word, max.	7
roots, total	23 667
morpheme tags, total	110
morpheme tags, derivatives	46
morphemes, total	1 646
derivational morphemes, total	1 172

that accounts for non productive derivation (NPD) and does not over-generate/-analyze. According to Table 2, to completely cover a 585k word sub-corpus[4] of the Kazakh Language Corpus (KLC) [7], one has to check the validity of attachment of 46 derivatives with 1172 allomorphs to various sub-sets of a set of 23.7k roots. Unfortunately, the task cannot be automated beyond accepting root-morpheme combinations that produce valid analyses found in the data (85.8k cases). Thus, several million unseen words will have to be checked manually. Clearly, NPD requires ad-hoc introduction of exception rules, which defeats the whole purpose of formal methods. On the other hand, given enough training data one could try to apply statistical models to tackle the problem. That is exactly what we are trying to do.

In this paper we describe a purely data driven approach that accounts for both inflectional and derivational morphology. Our method is oriented strictly on disambiguation, in that we do not set the goal of finding all possible analyses for a given word. It is rather that for a given word in a given context we try to find a set of analyses that contains the in-context-correct one. Such a formulation may seem odd, given a rather extreme ambiguity of the example outlined in Table 1. However, context puts sensible limits on ambiguity. Indeed, in the aforementioned KLC sample we found only three occurrences of our example word *almasyn*, two of which corresponded to the analysis #7, and one – to #5 (cf., Table 1). Thus, the in-context ambiguity for *almasyn* is only 1.5, as opposed to the in-language ambiguity of 16. As we gather from Table 2, for the sample in general, there are only 1.1 analyses per word on average, and – seven at most. This empirical evidence justifies our strategy of ranking analyses generated by our analyzer, and using only top five for disambiguation.

Our method implements a two-step analysis-disambiguation pipeline. The method requires morphologically segmented, analyzed and disambiguated training data. At the analysis step we employ a mixture of methods. Namely, we use: (i) a trivial look-up – to "analyze" seen words; (ii) a recursive procedure with a simple pruning heuristic – to segment unseen words; (iii) a bigram language model built on morpheme tags – to

[4] The largest morphologically disambiguated data set for Kazakh, but, of course, not a representative sample of the language by any means.

rank obtained segmentations. For disambiguation we build a first order HMM, treating words as observations and tag chains as hidden states.

We evaluate our analyzer and disambiguator separately, comparing them respectively to an FST-based analyzer [1] and Morfette [8], a language independent morphological disambiguation tool. In both experiments we compare the methods on a range of metrics, and for disambiguation, we report results achieved on several data sets. Our analyzer outperforms an FST-based analogue in both, precision and recall, achieving F-measure of 98.40% for the best setting. For disambiguation our method performs better than Morfette on smaller data sets, and achieves on par performance on the largest one. We have analyzed the errors made by both methods, and found that, apart from genuine ambiguity, inconsistency in data annotation greatly affected the performance.

Lastly, it should be noted that we do not account for compounding and orthographic distortions of roots and morphemes caused by morphophonemic rules. We plan to address these issues in the future.

1.1 Our Contribution

Our contribution lies in the development of a morphological disambiguation method for Kazakh language that:

- considers not fully productive derivation;
- needs no manual rule generation;
- was evaluated on the largest data set available for the language;
- is the first disambiguator for the language.

2 Related Work

Statistical approaches to morphological disambiguation have been successfully applied in the past. Hakkani-Tur et al. [9] model the distribution of morphological analyses of Turkish by breaking up analyses into smaller units, so called inflectional groups (IGs). Such an approach considerably reduces analyses sparsity improving predictions of a statistical model (a second order HMM). Additionally, it provides four different options of representing sequences of analyses through IG sequences. The authors show that the simplest representation which assumes that "... the presence of inflectional groups in a word depends only on the final inflectional groups of the last two words" [9] yields the highest performance. For morphological disambiguation of Czech, Hajič et al. [10] also use a second order HMM that runs on inputs partially disambiguated by a rule-based system. Chrupała et al. [8] cast the problem into a classification task, training two maximum entropy classifiers that provide probability distributions over analyses and word-lemma pairs. The authors use a language independent set of features, and show that their system performs well, achieving respective accuracies of 97%, 94%, and 82% for morphologically-rich languages, such as Romanian, Spanish, and Polish.

Along with supervised methods several unsupervised approaches were proposed [11, 12]. In a work by Creutz and Lagus [11] words are initially segmented using a baseline algorithm, which is based on a recursive minimum description length (MDL) model.

Then, initial segmentations are reanalyzed by more advanced models formulated in a maximum a posteriori probability, a maximum likelihood or an MDL framework. The authors refer to this collection of models as the Morfessor. A slightly modified version of the Morfessor was presented by Kohonen et al. [12], who implemented a semi-supervised extension to the baseline algorithm.

Recently there have been attempts to develop formal methods for morphological analysis of Kazakh. While Sharipbayev et al. [13] address the problem of generating Kazakh nominals, employing semantic neural networks, a number of works [1, 14–18] resort to finite state approaches. Washington et al. [1] develop three transducers for three Turkic languages of the Kypchak group, namely Kumyk, Tatar, and Kazakh. For Kazakh they report 85.6% coverage on a 25.6M Wikipedia corpus. They also evaluate the transducer on a 1000-word manually analyzed corpus, and achieve 98.6% precision and 57.9% recall. The transducer itself is freely available in the framework of the Apertium project[5]. We use it in the present study for comparison purposes. Upon inspection of the output of the transducer, we found that a small portion of not fully productive derivatives has been accounted for. Also, cases of attribution, substantivization and adverbialization of nominals, gerunds, and participles, have been implemented as conversions, i.e. null-morpheme derivations. However, stems derived from the bare roots such as [*bas-ta*], [*bas-tyq*], [*bas-ym*], etc. are not analyzed.

Kessikbayeva et al. [14] also resort to an FST-based approach, and formalize the nominal and verb paradigms. The authors provide schematic representations of nominal-verbal-nominal derivations, but do not discuss derivation in detail. Using the Xerox finite state toolkit [19] the authors conduct experiments on a set of 2000 randomly chosen analyses and report an overall data coverage of 96% (precision was not reported). Kairakbay et al. [16] present a formalization of the nominal paradigm, but does not conduct any direct evaluation. Finally, Makazhanov et al. [20] investigate the impact of using morphological information on the performance of POS-tagging for Kazakh. The authors, compare performances of two statistical taggers in two settings: (i) when only bare POS-tags are used, and (ii) when POS-tags are represented as tag chains that contain only inflectional morphemes. Thus, essentially, in the second case the authors address the problem of morphological disambiguation that ignores derivation. For this task the best performing tagger achieves an accuracy of 83%. Which is lower than the results for derivation-aware disambiguation achieved in the present study.

The main differences, that distinguish the present study from the aforementioned works on Kazakh morphology, constitute the basis of our contribution, and are listed in sub-section 1.1.

3 Methodology

In this section we describe our approaches to the tasks of morphological analysis and disambiguation respectively. Before discussing the approaches in detail, let us first explain basic assumptions under which our methods operate.

First, we want our methods to generalize on agglutinative languages (ALs), or, at the very least, on Turkic languages (TLs). Therefore, we do not make use of any knowledge

[5] `https://www.apertium.org`

of morphotactics and morphophonemics of Kazakh. Thus, our analyzer is prone to generation of ungrammatical analyses. To account for this, we introduced a simple sampling scheme that ranks analyzer outputs, and returns top five ranked analyses.

Second, we assume that a word in AL can be represented as a sequence of morphemes, in which a given morpheme depends only on the previous one, and the morpheme immediately following the root depends only on the POS tag of the root. In the present study we assume no prefixes, i.e. the root of a word is always its first morpheme (applicable to TLs). In the future, this assumption may be relaxed, and both, prefix and suffix, agglutination may be parameterized.

Third, we assume that data contains segmented analyses, and that derivation is explicitly marked. For instance, the KLC annotation uses a [POS1-POS2] convention for marking derivation, where the initial word class is denoted by POS1 and the derived class – by POS2. Similarly, general convention for Turkish [9, 21] is [DB+POS] where DB denotes a derivational boundary, and POS - the derived class.

3.1 Morphological Analysis

For MA we employ a two-step segmentation-ranking strategy. At the first step, given a word we try to segment it. If the word was segmented, we rank obtained analyses, and return top five (or less, if there were less segmentations to rank). If segmentation fails, we treat the word as a root, and return a list of POS-labeled roots.

In order two segment a word in a setting where morphotactics are not provided, one has to learn "what follows what" patterns from the data. Let us consider the following three segmentations: [*bala*_N-*lar*_PL] (*children*); [*bas*_N-*ym*_POSS.1SG] (*my head*); [*bala*_N-*lar*_PL-*ym*_POSS.1SG] (*my children*). We can conclude that plurality marker *lar* and the 1st person possession marker *ym* can follow a noun root, and that *ym* can follow *lar*. Clearly, one can parse the training set and retrieve all kinds of similar patterns, representing the morphotactics by means of a root lexicon and a graph or a table of morphemes that stores "X follows Y" patterns. Then, such a representation can be used to segment words by a recursive matching of suffixes and constant checking if prefixes (part of a word without suffixes) match known roots. However, this approach is prone to under-analysis, i.e. it may fail to obtain some segmentations. For instance, let our training set consist of segmentations [*bas*_N-*ym*_POSS.1SG] and [*bala*_N-*lar*_PL], and we have learned that "*lar*_PL follows N" and "*ym*_POSS.1SG follows N". Suppose we need to segment the word *balalarym*. The rules we have learned fail to segment *balalarym*, because, although we can match *ym*_POSS.1SG, we do not have a rule for *ym*_POSS.1SG following *lar*_PL, neither we have a noun root *balalar*_N in our root lexicon. To account for such cases we propose a notion of *transition chains*.

Given a segmentation, we define a transition chain as a morpheme chain whose morpheme tags are represented as POS to POS transitions. For example, for a segmentation [*al*_VB–*ma*_NEG–*s*_PTCP–*y*_POSS.3SG–*n*_ACC], we obtain the following transition chain: [*ma*_(VB_NEG)–*s*_(NEG_PTCP)–*y*_(PTCP_PTCP)–*n*_(PTCP_PTCP)]. This can be read as: *a verb is derived into a negative verb, which is derived into a participle, which is inflected by possessiveness marker* (not *derived* further, thus remains a PTCP),

which is marked by an accusative case marker (similarly, remains a PTCP[6]). Note, that under the third assumption given in the opening to this section, derivatives directly correspond to transitions, and inflections can always be identified. Thus, for any given segmentation we can construct a transition chain.

Having defined transition chains, we use a different representation for morphotactics. Instead of a root lexicon and a morpheme graph or table, we store three structures, namely, root and transition lexicons, and a transition-morpheme map. In the context of our previous example, after parsing the training set, we obtain a root lexicon {*bala*_N, *bas*_N}, a transition lexicon {*lar*_(N-N), *ym*_(N-N)}, and a transition-morpheme map {*lar*_(N-N):*lar*_PL, *ym*_(N-N):*ym*_POSS.1SG}. Now morphotactics becomes fairly simple: given two transitions A_P1_P2 and B_p1_p2, B can follow A if its left-hand side POS matches A's right-hand side POS (and vise versa), i.e. $p1 = P2 \implies Bfollows A$. Following this definition, we derive "*lar*_(N-N) follows *ym*_(N-N)", and after mapping transitions to conventional morphemes, we obtain the correct segmentation [*bala*_N-*lar*_PL-*ym*_POSS.1SG].

The segmentation module is implemented as a depth first search recursion, that uses transition chains in a manner described above. We use a max depth threshold of five, abandoning segmentations with more than five morphemes. If a prefix matches a known root we accept a segmentation, otherwise, if its length (in characters) becomes equal to one, a segmentation is abandoned. After transitions are converted back to morphemes, we pass the obtained segmentations to the ranker.

To rank segmentations we assign them probabilities estimated by a Markov chain model under the second assumption given in the opening to this section. Thus, a probability of a given segmentation S is computed as follows:

$$P(S) = P(r\_t) \prod_{i=1}^{n} P(m\_t_i | m\_t_{i-1})$$

where r_t is a POS-tag-labeled root, and m_t_i is the *i*-th morpheme of S. To estimate morpheme bigram probabilities we use MLE with the Laplace smoothing:

$$P(m\_t_i | m\_t_{i-1}) \approx \frac{N(m\_t_i, m\_t_{i-1}) + \alpha}{N(m\_t_{i-1}) + \alpha |M|}$$

where $N(m\_t_i, m\_t_{i-1})$ is the count of a given morpheme bigram, $|M|$ denotes the cardinality of the set of unique morphemes, and $\alpha = 0.1$ (estimated empirically).

The probability of the root is estimated as:

$$P(r\_t) \approx \frac{N(r\_t) + \alpha}{N + \alpha |R|}$$

where $N(r\_t)$ is the count of a given POS-tag-labeled root, N is the total number of all words in the training set, and $|R|$ is the size of a root lexicon. As in the previous case parameter α is estimated empirically to be equal to 0.1.

[6] Thus, in transition chains inflectional morphemes are represented as transitions of POS to themselves.

3.2 Morphological Disambiguation

We model a sequence of analyses given a sequence of words using HMM, and try to maximize the posterior probability, $P(T|W)$:

$$P(T|W) = \frac{P(T)P(W|T)}{P(W)}$$

where $W = w_1, w_2, ...w_k$ is an observation sequence (input words) and $T = t_1, t_2, ..., t_k$ is a state sequence, i.e. tag chains provided by the analyzer for each word. The denominator $P(W)$ remains constant for all segmentations, and thus can be dropped.

We compute $P(T)$ using a chain rule:

$$P(T) = \prod_{i=1}^{n} P(t_i|t_{i-1})$$

The transition probabilities of tag chain bigrams are estimated as:

$$P(t_i|t_{i-1}) = \frac{N(t_i, t_{i-1}) + \beta}{N(t_{i-1}) + \beta|V|}$$

where $N(t_i, t_{i-1})$ denotes the count of a given bigram, $N(t_{i-1})$ is the count of tag t_{i-1}, $\beta = 0.1$ (estimated empirically), and $|V|$ is the cardinality of a set of all unique tag chains in the training set. We compute emission probability $P(w_i|t_i)$ as follows:

$$P(w_i|t_i) \approx \frac{N(w_i, t_i) + \beta}{N(t_i) + \beta|W|}$$

where $N(w_i, t_i)$ is a number of times w_i was tagged by t_i, $N(t_i)$ is the count of tag t_i, and $|W|$ denotes the cardinality of a set of all unique wordforms found in the training set.

To find the most likely sequence of hidden states we utilize Viterbi algorithm. However, as opposed to a traditional approach, we do not consider all possible states (tag chains). Instead we consider only those tag chains that were obtained from the analyses returned for a given word, because, naturally, words accept only valid analyses,

4 Experiments and Evaluation

We evaluate our models on an annotated subset of the Kazakh Language Corpus [7]. All the experiments are performed in a 10-fold cross-validation setting. In order to check performance of our methods on various volumes of training data, we split the data into three data sets as shown in Table 3. DATA-k, the smallest data set, consists of 100 sentences and 1140.6 tokens per test fold. DATA-10k contains 1000 sentences an 11725.3 tokens per test fold. Lastly, the set DATA-ALL covers the whole data at hand, and consists of 4971.7 sentences 58483.9 tokens per test fold. Average ambiguity is calculated as an average number of analyses per word. As one would expect, the ratio of unseen tokens drops with increase in data volume, and the ambiguity increases.

Table 3. Characteristics of data sets

Data set	# sentences	# tokens	avg. ambiguity	unseen tokens, %
DATA-k	100.00±0.00	1140.60±0.49	1.06±0.27	43.62±1.69
DATA-10k	1000.00±0.00	11725.30±0.46	1.09±0.36	17.18±0.61
DATA-ALL	4971.70±0.46	58483.90±1.30	1.11±0.40	7.25±0.14

We start by evaluating our analyzer, on the DATA-ALL set, in terms of coverage, average ambiguity, precision, recall, and f-measure. Coverage is defined as a number of words, for which at least one analysis is returned. In other words, it is a fraction of segmented words. Ambiguity is an average number of analyses returned per word. Precision is defined as a fraction of *segmented* words, for which a correct analyses was given. Similarly, recall is a fraction of *all* words, for which a correct analyses was given. Given precision (P) and recall (R), f-measure is calculated as: $F = (2PR)/(P + R)$.

We compare the performance of our method (OM) to Apertium project's HFST-based analyzer [1] (AHFST) and a look-up baseline (LU). We ran AHFST on our test sets and converted its output to the KLC format in order to be able to evaluate the method against the KLC-formatted golden truth[7]. As for LU, for seen words the baseline retrieves analyses from the data set, and renders unseen words unsegmented.

In order to assess the impact of ranking on the performance of our analyzer, we consider two additional settings: OM-R (without ranking) and OM+LU (with look-up). One would expect that ranking may hurt performance, because some lower-ranked and thrown away analyses may have been correct. Thus, we would expect our method perform better in OM-R setting, where all obtained analyses are evaluated. Similarly, given a relatively low ratio of unseen tokens (7%, cf. Table 3), one would expect LU to perform well. Thus, for OM+LU setting, where seen words are not analyzed, but retrieved from the training set, it is also expected to see increase in performance.

Table 4 shows the results of the experiment. In terms of coverage OM achieves the best performance in all three settings. In fact, the sets of segmented inputs are equal for OM and OM-R, and the corresponding set for LU is almost entirely in OM's set. Hence, we get equal coverages for the three. Given that the sets of segmented inputs and all inputs almost completely (99.8%) overlap, precision and recall values are so closed for each OM setting. Coverage of LU is simply a fraction of seen words in the data. AHFST displays the lowest performance, covering only 85.2% of the data.

When it comes to ambiguity both AHFST and LU perform well, yielding no more than 2.59 and 1.76 analyses per word. As we have anticipated, without ranking our

[7] Here, it should be noted that the AHFST should not be directly compared to the other methods. First, there is a tag set conversion issue. The tag set used in the KLC differs dramatically from the one used by the AHFST. The latter marks copulas and null-morphemes (e.g. nominal case, substantivization), the former does not. Second, a lot of analyses in the KLC stem from bare roots, and contain non-productive derivatives, which are completely absent from AHFST lexicon. We tried to account for all these issues, and make the conversion as general as possible, performing one-to-many mappings from the AHFST output to the KLC data. Lastly, as we later discovered, the data contained certain amount of typos, which must have prevented AHFST from generating correct analyses.

Table 4. Morphological analysis evaluation

Method	coverage, %	avg. ambiguity	precision, %	recall, %	f-measure, %
OM	**99.77** ± 0.02	4.43 ± 0.01	89.99 ± 0.19	90.00 ± 0.19	90.00 ± 0.19
OM+LU	**99.77** ± 0.02	1.98 ± 0.01	97.43 ± 0.08	**97.43** ± 0.08	**97.43** ± 0.08
OM-R	**99.77** ± 0.02	594.15 ± 3.74	96.25 ± 0.26	96.03 ± 0.26	96.14 ± 0.26
AHFST	85.21 ± 0.19	2.59 ± 0.01	74.75 ± 0.16	63.69 ± 0.16	68.78 ± 0.14
LU	92.75 ± 0.14	**1.76** ± 0.01	**98.94** ± 0.04	91.77 ± 0.13	95.22 ± 0.08

analyzer generates extremely ambiguous outputs (594 analyses per word), and, if ranking is introduced, the ambiguity reduces in more than 100 times. Of all three settings OM+LU achieves the lowest ambiguity of 1.98 analyses per word.

In terms of precision the baseline performs the best, and our method achieves a slightly lower precision of 97.43%, when coupled with the look-up baseline. Again, as we expected, ranking resulted in a considerable 6% drop in precision (OM vs OM-R). However, combining the method with the look-up baseline, reflects positively on performance. The same is also true for recall, where ranking introduces a 6% drop, and look-up adds another 1.4%. Recall is an important metric in this experiment, as it gives an upper bound on accuracy of disambiguation. Thus, because the highest recall was achieved by our analyzer in combination with the look-up baseline, we will use OM+LU setting at the MA step of our next experiment on disambiguation.

Let us move on to disambiguation, where we compare our method with an open-source language-independent tool, Morfette [8] and a look-up baseline that assigns all seen words their most frequent tag and tags all unseen words as nouns. We evaluate the methods in terms of accuracy as a fraction of correctly disambiguated words.

Table 5 shows the results grouped by data sets and all, seen, and unseen words. As it can be seen, the "most frequent + noun" strategy works exceptionally well for seen words, as the LU baseline achieves the best performance, and is closely followed by our method. For unseen words Morfette performs better and with the increase in data volume its accuracy grows faster than that of our method's, which outperformed Morfette only on the smallest data set. Ultimately, for all words our method yields better performance, on two smaller data sets, and performs in par with Morfette on the complete data volume, losing only 0.03%.

From the experiment it can be seen that as data volume grows the respective performances of our method and Morfette grow slower, and it may become harder to beat a look-up baseline on a large enough data sets, given the same 90/10 train/test split.

In the following subsection we discuss some of the frequent errors we encountered, and practical issues we had with some of the methods we used.

4.1 Error Analysis and Practical Issues

Our error analysis relies on visual examination of erroneous outputs generated by our method and Morfette. We classify errors into three broad categories: (i) contextual; (ii) root inconsistency; (iii) morpheme chain inconsistency. Contextual errors were the most common for both methods. As we explained in the introduction, errors of this

Table 5. Morphological disambiguation evaluation

Method / Data set	acc. all, %	acc. seen, %	acc. unseen, %
DATA-k			
Our method	**68.43 ± 1.36**	86.15 ± 1.47	**45.60 ± 2.58**
Morfette	65.11 ± 0.93	81.21 ± 1.26	43.84 ± 0.85
LU baseline	54.59 ± 2.08	**86.20 ± 1.13**	13.88 ± 3.00
DATA-10k			
Our method	**81.23 ± 0.56**	87.72 ± 0.39	49.92 ± 1.09
Morfette	80.11 ± 0.53	85.79 ± 0.40	**52.69 ± 0.69**
LU baseline	74.97 ± 0.89	**88.12 ± 0.56**	11.56 ± 0.59
DATA-ALL			
Our method	85.85 ± 0.16	88.75 ± 0.12	48.72 ± 1.05
Morfette	**85.88 ± 0.11**	88.16 ± 0.10	**56.68 ± 1.00**
LU baseline	83.67 ± 0.14	**89.33 ± 0.12**	11.23 ± 0.73

type mostly relate to ambiguity in a 3rd person possession and agreement markers, as these morphemes have the same surface forms regardless of numbers of possessors and subjects. Same goes for indefinite future-present tense ambiguity, e.g. [bar_V-a_*FUT/PRES*-myn_1SG] I (*will*) go. Our bigram model may not have captured contextual clues (number of a possessor and a subject, temporal expressions, etc.), because it was not found in the immediate preceding context.

Root inconsistency errors were the second most common. These errors directly relate to the discussion, that was given in the introduction, on non-productive derivation. In some cases words were stemmed from bare roots by the analyzers, but were kept as lemmata in the data and vice versa. These cases are due to data inconsistency and only thing that can be done about it is to set up a convention on dealing with such cases and re-annotate bits of data containing them.

Morpheme chain inconsistency errors were less frequent, but had different consequences depending on which method had erred. Our method often joined otherwise separate morphemes and vice versa, e.g., in the data, the word *berushi* might have been segmented as [ber-V_-u_(V-GER)-shi_(GER-N)], while our method would favor [ber-V_-ushi_(V-N)]. Both cases could have appeared in the data, so the error may be addressed to the data inconsistency. For the Morfette, however, we have received some analyses whose morpheme chains corresponded to impossible segmentations. For instance: for the word [maqsat-tar-y-nyng] (of his/her/their dreams) we have received the analysis [maqsat-VB-GER-PL-GEN] which does not apply for the surface form. We do not want to speculate on as to what caused this error, but we are almost certain that the reason was not in the data.

Lastly, we want to comment on some practical issues. Whilst acknowledging Morfette's out-of-the-box convenience, we have to note, that in our experiments on complete data set (526k/58k tokens train/test) it took us 30+ CPU hours and 26GB of memory to train, and 5-6+ CPU hours to test. We have initialized maxPrefix variable to 0 (there are no prefixes in Kazakh) in the source code (Lemma.hs, POS.hs), and otherwise ran the tool on default parameters. Compared to that our method runs in a matter of minutes for training and seconds for testing, and uses no more than 1GB of memory.

5 Conclusion and Future Work

We have developed a data-driven method for morphological analysis of Kazakh that accounts for both inflectional and derivational morphology. The method does not require formalization, because morphotactics are induced directly from labeled data in the form POS-to-POS morpheme transitions. Our experiments suggest that such a representation of morphotactics help in pruning many false segmentations, while keeping correct ones. We have evaluated our method against open source solutions on several data sets, and showed that our method achieves better or on par performance. We also have to note that, comparing to some of the existing solutions, our method is less time and resource consuming. However, while our method is efficient and achieves a good performance relative to others, disambiguation accuracy could and should be improved. Our future work will be dedicated to introducing necessary adjustments to the method to facilitate compound-aware analyses that account for morphophonemics of the language. We also plan to utilize more sophisticated models for morphological disambiguation. Lastly, as we discovered issues with annotation inconsistency in the annotated sub-corpus of the Kazakh Language Corpus [7], we plan to use our method for semi-automatic noise reduction in this data set.

Acknowledgments. This work has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan.

References

1. Washington, J., Salimzyanov, I., Tyers, F.: Finite-state morphological transducers for three kypchak languages. In: Calzolari N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014. European Language Resources Association (ELRA), Reykjavik (May 2014)
2. Oflazer, K., Güzey, C.: Spelling correction in agglutinative languages. In: ANLP, pp. 194–195 (1994)
3. Sak, H., Güngör, T., Saraçlar, M.: A stochastic finite-state morphological parser for turkish. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort 2009, pp. 273–276. Association for Computational Linguistics, Stroudsburg (2009)
4. Koskenniemi, K.: A general computational model for word-form recognition and production. In: Proceedings of the 10th International Conference on Computational Linguistics, pp. 178–181. Association for Computational Linguistics (1984)
5. Hulden, M.: Foma: a finite-state compiler and library. In: Lascarides, A., Gardent, C., Nivre, J. (eds.) EACL (Demos), pp. 29–32. The Association for Computer Linguistics (2009)
6. Lindén, K., Axelson, E., Hardwick, S., Pirinen, T.A., Silfverberg, M.: HFST-Framework for Compiling and Applying Morphologies. In: Mahlow, C., Piotrowski, M. (eds.) SFCM 2011. CCIS, vol. 100, pp. 67–85. Springer, Heidelberg (2011)
7. Makhambetov, O., Makazhanov, A., Yessenbayev, Z., Matkarimov, B., Sabyrgaliyev, I., Sharafudinov, A.: Assembling the kazakh language corpus. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1022–1031. Association for Computational Linguistics, Seattle(2013)

8. Grzegorz Chrupała, G.D., van Genabith, J.: Learning morphology with morfette. In: Calzolari, N., Khalid Choukri, B.M.J.M.J.O.S.P.D.T. (eds.) Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008. European Language Resources Association (ELRA), Marrakech (May 2008), http://www.lrec-conf.org/proceedings/lrec2008/
9. Hakkani-Tur, D.Z., Oflazer, K., Tur, G.: Statistical morphological disambiguation for agglutinative languages. Computers and the Humanities 36(4), 381–410 (2002)
10. Hajič, J., Krbec, P., Květoň, P., Oliva, K., Petkevič, V.: Serial combination of rules and statistics: A case study in czech tagging. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL 2001, pp. 268–275. Association for Computational Linguistics, Stroudsburg (2001)
11. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Transactions on Speech and Language Processing (TSLP) 4(1), 3 (2007)
12. Kohonen, O., Virpioja, S., Leppänen, L., Lagus, K.: Semi-supervised extensions to morfessor baseline. In: Proceedings of the Morpho Challenge 2010 Workshop. Aalto University School of Science and Technology Faculty of Information and Natural Sciences Department of Information and Computer Science, Espoo, Finland (September 2010)
13. Sharipbayev, A., Bekmanova, G., Ergesh, B., Buribayeva, A., Karabalayeva, M.K.: Intellectual morphological analyzer based on semantic networks. In: Proceedings of the OSTIS 2012, pp. 397–400 (2012)
14. Kessikbayeva, G., Cicekli, I.: Rule based morphological analyzer of kazakh language. In: Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM, pp. 46–54. Association for Computational Linguistics, Baltimore (2014)
15. Altenbek, G., Xiao-long, W.: Kazakh segmentation system of inflectional affixes. In: Joint Conference on Chinese Language Processing, CIPS-SIGHAN, pp. 183–190 (2010)
16. Kairakbay, B.M., Zaurbekov, D.L.: Finite state approach to the Kazakh nominal paradigm. In: Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, pp. 108–112. Association for Computational Linguistics, St Andrews (2013)
17. Makazhanov, A., Makhambetov, O., Sabyrgaliyev, I., Yessenbayev, Z.: Spelling correction for kazakh. In: Gelbukh, A. (ed.) Proceedings of the 2014 Computational Linguistics and Intelligent Text Processing. LNCS, vol. 8404, pp. 533–541. Springer, Heidelberg (2014)
18. Zafer, H.R., Tilki, B., Kurt, A., Kara, M.: Two-level description of kazakh morphology. In: Proceedings of the 1st International Conference on Foreign Language Teaching and Applied Linguistics, FLTAL 2011, Sarajevo (May 2011)
19. Ranta, A.: A multilingual natural-language interface to regular expressions. In: Proceedings of the International Workshop on Finite State Methods in Natural Language Processing, FSMNLP 2009, pp. 79–90. Association for Computational Linguistics, Stroudsburg (1998)
20. Makazhanov, A., Yessenbayev, Z., Sabyrgaliyev, I., Sharafudinov, A., Makhambetov, O.: On certain aspects of kazakh part-of-speech tagging. In: 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–4 (October 2014)
21. Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G.: Building a turkish treebank. In: Treebanks, pp. 261–277. Springer (2003)

Statistical Sandhi Splitter for Agglutinative Languages

Prathyusha Kuncham, Kovida Nelakuditi, Sneha Nallani, and Radhika Mamidi

IIIT Hyderabad
{prathyusha.k,nelakuditi.kovida,
sneha.nallani}@research.iiit.ac.in, radhika.mamidi@iiit.ac.in

Abstract. Sandhi splitting is a primary and an important step for any natural language processing (NLP) application for languages which have agglutinative morphology. This paper presents a statistical approach to build a sandhi splitter for agglutinative languages. The input to the model is a valid string in the language and the output is a split of that string into meaningful word/s. The approach adopted comprises of two stages namely Segmentation and Word generation, both of which use conditional random fields (CRFs). Our approach is robust and language independent. The results for two Dravidian languages viz. Telugu and Malayalam show an accuracy of 89.07% and 90.50% respectively.

1 Introduction

Agglutinative languages are rich in morphology. There are many agglutinative languages such as Dravidian languages, Turkic languages etc. In these languages many words combine to form a compound word. In this process, morphophonological changes i.e. fusion of final and initial characters occur at word boundaries. This is termed as "Sandhi".

Examples of sandhi in compound words:

(a) Compound Nouns:
'vixyAlayaM'[1] → 'vixya' + 'AlayaM'
university *education* *temple*

(b) Compound Verbs:
'kUdabeVttu' → 'kUdu' + 'peVttu'
to accumulate *be gatherer* *keep*

(c) Other type of compound words:
'rAmudeVkkada' → 'rAmudu + 'eVkkada'
Where is Ramudu *Ramudu* *where*

If this word is given as an input to a Question Answering system, it is very important to identify the question word 'eVkkada' (where) for proper

[1] Words are in wx format (sanskrit.inria.fr/DATA/wx.html). All the examples given in the paper are from Telugu language.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 164–172, 2015.
DOI: 10.1007/978-3-319-18111-0_13

functioning of the system which can be obtained only by splitting the compound word.

As observed from the above examples, one has to split (c) as it is morphologically unanalysable and if not split, degrades the performance of NLP applications [1]. It is not necessary to split (a) and (b) as these are frequently occurring collocations in the language and are also handled by the existing morphological analyser [2]. Therefore, we focused on handling words of type (c) in this paper.

We developed a statistical sandhi splitter for agglutinative languages. Our approach uses CRF which is one of the most successful statistical learning methods in NLP for labeling and segmenting sequential data [3]. Our approach consists of two stages namely Segmentation [4], [5], [6] and Word generation as discussed in section 4.

2 Related Work

Sandhi splitting can be done using (a) Rule based techniques (b) Statistical techniques and (c) Hybrid approaches.

(a) ***Rule based systems:***
[7] and [8] developed a rule based system to split compound words into meaningful sub-words in Malayalam and Marathi respectively. However, the main drawback of this type of systems is that they require a lot of manual effort and time to prepare rules. Moreover, the system is language dependent.

(b) ***Statistical systems:***
[9] built a Finite state transducer (FST) which was used to identify possible words for a given compound word with 80.3% accuracy. This approach fails for out-of-vocabulary (OOV) words i.e. where the base word doesn't exist in the FST. [10] used statistical methods like Dirichlet process and Gibbs Sampling for Sanskrit sandhi splitting.

(c) ***Hybrid systems:***
These systems combine both statistical and rule based techniques. [11] gives an accuracy of 91.1% for Malayalam. Hybrid systems are also not language independent.

Segmentation in our approach has been inspired from [5]. Our model is purely statistical and language independent. When compared to rule based and hybrid approaches, our model is robust, faster and requires less effort.

3 Dataset

There was no available sandhi annotated data in agglutinative languages except for Malayalam. We have prepared sandhi annotated dataset for Telugu language. Following decisions were made while annotating the training data.

(A). Context dependent particles: In cases where contextual information is required to decide whether or not to split a word, a decision to not split the word was made. The following examples in Telugu give an insight into these occurrences.

(i) "gA"
"gA" can act as question particle or it can mean "while" depending on the situation.
a. "unnAvugA" → 'unnAvu' + gA
(you are) present, aren't you? *present* *aren't you?*
b. "undagA" → undu + gA
while present *present* *while*

(ii) Clitics [12]
We look at Telugu clitics 'e','o' which are ambiguous.
Examples:
a. 'Amek**e** puswakam kAvAli' *(What book does she need?)*
Here, 'e' acts as a question marker.
b. 'nenu Amek**e** iccAnu' *(I gave only to her (not others))*
Here, 'e' is emphatic clitic.

a. 'rAjuk**o** BArya uMxi' *(King has one wife)*
Here, 'o' acts as a quantifier 'one'.
b. 'rAjuk**o** rAniko kala vacciMxi' *(either king or queen got a dream)*
Here, 'o' is an indefinite clitic.

Ideally, we need (a) cases to be split and (b) cases not to be split, the decision whether to split or not can be made with contextual information obtained from words of compound word itself or sentence. Such cases which require this contextual information to disambiguate the sense are decided not to be split.

(B). Dialectal influence: The base form of a word may change with dialect. Therefore, only words from the standard written language are considered while preparing the training data.

Example:
In standard Telugu, 'vaccAraMxaru' *(all came)* is 'vacciMdraMxaru' in Telangana Telugu dialect. So 'vaccAraMxaru' → vaccAru *(came)* + aMxaru *(all)* is included in training data but not 'vacciMdraMxaru'.

4 Our Approach

Our approach consists of two stages viz., Segmentation and Word Generation. A flow chart of the system is shown in Figure 1.

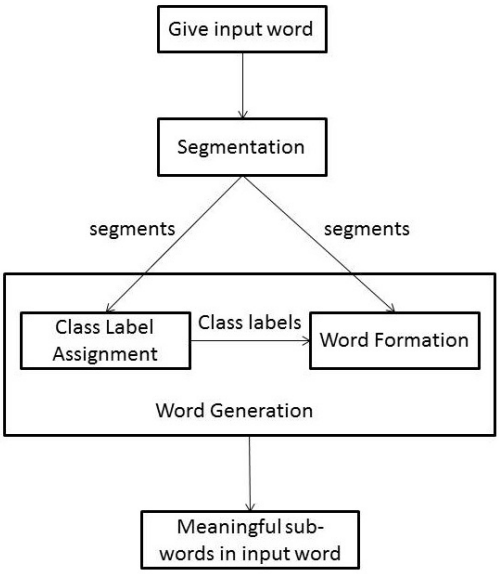


Fig. 1. Flow chart of the Sandhi Splitter module

4.1 Segmentation

In this stage, at each character in a word, CRF model decides whether or not to split at that point. Thereby, we identify the boundaries between different words i.e. the points where the morphphonological changes occur in the compound word. This is formulated as a two-class classification problem. The input for this task is a word and the output is the segments that show the boundary/split points in the input. The resulting segments may or may not be meaningful words.

Example:

Input: 'pUjayyAkA' *(after having finished the prayer)*
Output: 'pUj'-'ayyAkA'
Here, 'pUjayyAkA' → 'pUja' + 'ayyAkA'.
prayer *finished*

We can observe the morphophonological change (a → a + a) at the word boundaries. In the above output, the segments are "pUj" and "ayyAka" where "pUj" is not a meaningful word in Telugu.

At this stage, CRF was trained with following feature set.

Feature Set:
Characters*:* Morphophonological changes occur at character level. So this feature is important to identify where and what type of morphophonological changes take place in the word.

Character Tags: Every character is given a tag based on its type of sound (consonant/short vowel/long vowel). This is important to capture the information of types of vowel/consonant clusters that occur during morphophonological changes.

4.2 Word Generation

This stage majorly deals with the generation of meaningful words from the segments obtained in segmentation stage. The input to this stage is the segments of the compound word and the outputs are the different meaningful words.

Example:

Input: 'pUj'-'ayyAkA'
Output: 'pUja' *(prayer)*
'ayyAkA' *((after having) finished)*

Word generation has two components:

- Class label assignment
- Word Formation

4.2.1 Class Label Assignment

The morphophonological changes of addition or deletion of characters in sandhi are finite and form a class space. From the training data collected, automatically 41 such classes are extracted for Telugu and 49 for Malayalam.

Input:	'baMXuvoVkaru' *(one relative)*		
Segmentation:	'baMXuv-oVkaru'		
Class label assignment:	'baMXuv' _u	→	'baMXuvu' *(relative)*
	'oVkaru' NULL	→	'oVkaru' *(one)*

In the above example the segments are 'baMXuv' and 'oVkaru'. 'baMxuv' will be meaningful if 'u' is added at the end and 'oVkaru' is itself a meaningful word. So these two words fall into '_u' and 'NULL' classes respectively.

Having generated these classes we prepare the training data for this stage with segments and class labels. CRF was trained with following feature set.

Feature Set:

Segments: This feature is important because segments which precede or follow decide the class label for a current segment in some cases.

Example:

a.

Input:	'manixxariki' *(for both of us)*		
Segmentation:	'man-ixxariki'.		
Class label assignment:	'man' _a	→	'mana' *(our)*
	'ixxariki' NULL	→	'ixxariki' *(for both)*

b. Input: 'manixxaraM' *(we both)*

Segmentation:	‘man-ixxaraM’		
Class label assignment:	‘man’ _aM	→	‘manaM’ *(we)*
	‘ixxaraM’ NULL	→	‘ixxaraM’ *(both)*

From this example, the class label for the segment ‘man’ is decided based on its succeeding segments.

Prefix & Suffix Characters: The prefix and suffix of a segment plays an important role in deciding the class label which can be seen in Table 2.

4.2.2 Word Formation

This step deals with generating a meaningful word from a segment using the information from the class label. The segments that have same class label adopt same method for formation of words.

As continuation to the example ‘baMXuvoVkaru’, discussed in section 4.2.1, we have

Word formation:	‘baMXuvoVkaru’	→	‘baMxuvu’ + ‘oVkaru’	
	One relative		*relative*	*one*

We add ‘u’ at the end of ‘baMxuv’ and the resulting word is ‘baMxuvu’. In case of ‘oVkaru’, as its class label is ‘NULL’, no change is made. So ‘baMXuvoVkaru’ is split into two meaningful words ‘baMxuvu’ and ‘oVkaru’.

5 Experiments and Results

Our model was trained on 1267 Telugu words. Development and test sets have 800 and 1151 words respectively. Test data contains words which have sandhi (Split) and which do not (Non-split). If CRF is used, one has to choose a proper feature template for accurate performance of the system. Table 1 and Table 2 show the results when the model is trained on different feature templates for Segmentation and Class label assignment stages respectively. From Table 1 and Table 2, we can observe that the template 4 gives better accuracy in both the tasks.

Table 1. Results of different feature templates in Segmentation task on development data

Template	C	T	C&T	Precision	Recall	F-Measure
1	2	0	No	97.14	95.33	96.22
2	2	2	Yes	97.14	94.07	95.58
3	3	0	Yes	97	95.51	96.25
4	3	3	Yes	96.94	96.09	**96.51**
5	3	2	No	97.07	95.32	96.19
6	3	1	No	97.15	95.39	96.26
7	2	1	No	97.34	95.14	96.23

- If ‘C’ = k, k characters on the left and right to the current character are included as features.
- If ‘T’ = k, k tags on the left and right to the current tag are included.
- An example for ‘C&T’ is C-1/T-1 which means previous character and previous tag is included.

Table 2. Results for different feature templates in Class label assignment task on development data

Template	Pw-C	C	Nw-C	Word Accuracy
1	1-0	3	1-0	96.61
2	1-2	3	1-2	96.87
3	2-2	3	2-2	96.69
4	2-3	3	2-3	**96.95**
5	1-3	3	1-3	96.61

- If ‘P-C’ is ‘n-k’, n previous segments along with k, k-1..., 1 character/s from the beginning (prefix) and ending (suffix) of the corresponding segments are included.
- If ‘N-C’ is ‘n-k’, n next segments along with k, k-1…, 1 character/s from the beginning and ending of the corresponding segments are included.
- If ‘C’ is k then k, k-1..., 1 character/s from the starting and ending of the current segment are considered as features.
- ‘Word Accuracy’ gives the percentage of words which were correctly class labelled.

After feature template selection for the two stages, the same templates were used for testing on Malayalam. The overall system when tested on Telugu data gave 89.07% accuracy and 90.5% on Malayalam data.

Table 3. Overall accuracy of system on Telugu and Malayalam test data

Language	#Train	#Test	#Split in Test	#Non-split in Test	Accuracy
Telugu	1267	1151	286	865	89.07
Malayalam	1926	1000	260	740	90.50

Comparison with Other Systems

We compare our system with a few existing Statistical and Hybrid sandhi splitting systems for Telugu and Malayalam languages.

- As mentioned in section 2, [11] gives an accuracy of 91.1% whereas our system gives 90.5% for the same dataset of Malayalam language. Though the difference of accuracies is very small, unlike their system our system can be easily adaptable to any language as it is purely statistical.
- We could not compare our system with [9] as the dataset they used is unavailable. Compared to their system, our system is dynamic as it can handle OOV words

because the features are not solely defined with respect to the vocabulary but are also derived from characters in word/s.

Due to the unavailability of parallel corpus for sandhi-split words for other agglutinative languages, we couldn't test on languages other than Telugu and Malayalam. But, we are confident that it will work for other agglutinative languages as well on the basis of their common typological features.

6 Conclusion

We have presented our efforts in building a statistical sandhi splitter. Our model is language independent mainly because of automatically extracted class labels. Even though we have handled sandhi in one type of compound words, our model can be readily adapted to other types as well. The model has been tested on Telugu and Malayalam. Through this work, we have also prepared a standard dataset for Telugu language consisting of a corpus of agglutinated words and its parallel corpora of sandhi-split words. Showing the impact of sandhi splitting on NLP applications like Machine Translation, Parsers, Dialogue System etc., is part of our immediate future work. As discussed in section 3, a few words require contextual information to split and a few words show dialectal influence. We plan to extend our model by taking contextual information and non-standard language into consideration in future.

Acknowledgements. This work is supported by Information Technology Research Academy (ITRA), Government of India under, ITRA-Mobile grant ITRA/15(62)/ Mobile/VAMD/01.

References

1. Kolachina, S., Sharma, D.M., Gadde, P., Vijay, M., Sangal, R., Bharati, A.: External sandhi and its relevance to syntactic treebanking. Polibits (43), 67–74 (2011)
2. Bharati, A., et al.: Natural language processing: a Paninian perspective, ch. 3. Prentice-Hall of India, New Delhi (1995)
3. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labelling sequence data (2001)
4. Nguyen, C.-T., Nguyen, T.-K., Phan, X.-H., Nguyen L.-M., and Ha, Q.-T.: Vietnamese word segmentation with crfs and svms: An investigation. In: Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation, PACLIC 2006 (2006)
5. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 562. Association for Computational Linguistics (2004)
6. Xue, N., et al.: Chinese word segmentation as character tagging. Computational Linguistics and Chinese Language Processing 8(1), 29–48 (2003)

7. Nair, L.R., Peter, S.D.: Development of a rule based learning system for splitting compound words in malayalam language. In: 2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 751–755. IEEE (2011)
8. Joshi Shripad, S.: Sandhi splitting of Marathi compound words. Int. J. on Adv. Computer Theory and Engg. 2(2) (2012)
9. Vempaty, P.C., Nagalla, S.C.P.: Automatic sandhi spliting method for telugu, an indian language. Procedia-Social and Behavioral Sciences 27, 218–225 (2011)
10. Natarajan, A., Charniak, E.: S3-statistical sam. dhi splitting (2011)
11. Devadath, V.V., Kurisinkel, L.J., Sharma, D.M., Varma, V.: A sandhi splitter for malayalam (accepted but yet to be published in proceedings of ICON 2014)
12. Krishnamurti, B.: A grammar of modern Telugu. Oxford University Press, New York (1985)

Chunking in Turkish with Conditional Random Fields

Olcay Taner Yıldız[1], Ercan Solak[1], Razieh Ehsani[1], and Onur Görgün[1,2]

[1] Işık University, Istanbul, Turkey
[2] Alcatel Lucent Teletaş Telekomünikasyon A.Ş., Istanbul, Turkey

Abstract. In this paper, we report our work on chunking in Turkish. We used the data that we generated by manually translating a subset of the Penn Treebank. We exploited the already available tags in the trees to automatically identify and label chunks in their Turkish translations. We used conditional random fields (CRF) to train a model over the annotated data. We report our results on different levels of chunk resolution.

1 Introduction

Chunking is one of the earlier steps of parsing and defined as dividing a sentence into syntactically disjoint sets of meaningful word-groups or chunks. The concept of phrase chunking was proposed by [1]. [1] argued that words could be brought together into disjoint 'chunks' and therefore on the whole simpler phrases in general. An example of a sentence split up into chunks is shown below:

(1) [NP John] [VP guesses] [NP the current deficit] [VP will decrease] [PP to only $1.3 billion] [PP in October]

The chunks are represented as groups of words between square brackets, and the tags denote the type of the chunk.

Extracted chunks can be used as input in more complex natural language processing tasks, such as information retrieval, document summarization, question answering, statistical machine translation, etc. Compared to syntactic and/or dependency parsing, chunking is an easier problem. For this reason, in the earlier years of statistical natural language processing, many researchers put emphasis on the chunking problem [2].

[3] was one of the earliest works. Their transformation-based learning approach achieved over 90% accuracy for NP chunks on the data derived from Penn-Treebank. [4] applied support vector machines (SVM) to identify NP chunks. Using an ensemble of 8 SVM-based systems, they got 93% in terms of F-measure. [5] applied generalized winnow (a classifier much simpler than the ensembles of SVM's) on the overall chunking problem, and get an average of 94% in terms of F-measure on CoNLL shared task data [6]. [6] give an overview of the CoNLL shared task of chunking. 10 different chunk types covered in CoNLL shared task are ADJP, ADVP, CONJP, INTJ, LST, NP, PP, PRT, SBAR, and VP. Training material consists of the 9,000 Wall Street Journal sentences augmented with POS tags. Although there is quite a bit of work in English chunking done in the last two decades, the work in other languages, especially on less resourced and/or morphologically rich languages, is scarce. There are limited works on Korean [7,8], Hindi [9], and Chinese [10,11].

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 173–184, 2015.
DOI: 10.1007/978-3-319-18111-0_14

[12] implemented the first Turkish NP chunker which uses dependency parser with handcrafted rules. NP's are divided into two sub-classes as main NPs (base NPs) and all NPs (including sub-NPs). Noun phrases with relative clauses are omitted in his work. [13] introduced a new chunk definition by replacing phrase chunks with constituent chunks. They used METU-Sabancı Turkish dependency treebank [14] for chunk annotation and only labeled verb chunks. The remaining chunks are left as general chunks and only their boundaries are detected. The algorithm is based on conditional random fields (CRF) enhanced with morphological and contextual features. According to the experiment results, their CRF-based chunker achieves a best accuracy (in terms of F-measure) of 91.95 for verb chunks and 87.50 for general chunks.

In this paper, we propose a general CRF-based Turkish chunker. Our contributions are two-fold; (i) we automatically construct a chunking corpus using the parallel Turkish treebank of about 5K sentences translated out of Penn-Treebank [15] (ii) we improve upon the work of [13] by learning all common chunk types in Penn-Treebank. We define three learning problems in increasing levels of difficulty. In the first level, we only identify the boundaries of chunks. In the second level, we detect chunk types. In the third level, we try to discriminate chunk sub-types (NP-SBJ, NP-OBJ, ADVP-TMP, etc.) of the chunks in the second level.

The paper is organized as follows: In Section 2, we give a very brief overview on Turkish syntax. We give the details of our data preparation steps in Section 3 and describe the CRF for chunking in Section 5. Our CRF features for chunking are detailed in Section 6 and we give the experimental results using those features in Section 7. We conclude in Section 8.

2 Turkish

Turkish is an agglutinative language with rich derivational and inflectional morphology. Morphemes attach to the stems as suffixes. Word forms usually have a complex yet fairly regular morphotactics. Most suffix morphemes have several allophones which are selected by morphophonemic constraints, [16].

Turkish sentences have an unmarked SOV constituent order. However, depending on the discourse, constituents are often scrambled to emphasize, topicalize, focus and background certain elements. Writing and formal speech tend towards the unmarked order.

Turkish is a head final language. Adjectives qualifying a head in a noun phrase are usually scrambled for emphasis. Even then, intonation in speech is used to add a further layer of emphasis.

Below is a Turkish sentence with the chunks identified in brackets and subscript labels.

(2) [Dün]$_{\mathrm{ADJP}}$ [büyük bir araba]$_{\mathrm{NP}}$ [beyaz binanın önüne]$_{\mathrm{PP}}$ [geldi]$_{\mathrm{VP}}$.
[White building.GEN front.DAT]$_{\mathrm{PP}}$ [yesterday]$_{\mathrm{ADJP}}$ [big a car]$_{\mathrm{NP}}$ [come.PAST.3SG]$_{\mathrm{VP}}$.
Yesterday, a big car came to the front of the white building.

In Turkish syntax, case markings of the heads identify the syntactic functions of their constituents. For example, accusative marking identifies the direct object of a transitive

verb. Similarly, dative marker sometimes denotes the directional aspect of its phrase and sometimes marks the oblique object of a verb. Although this aspect of Turkish syntax makes it a bit peculiar, in this work, we refrained from inventing our own set of Turkish chunk labels and used the same set of chunk labels that are generally used in English chunking.

3 Data Preparation

Constructing a chunking corpus from scratch by manually tagging and chunking linguistic data is very expensive. Instead, one can use already tagged corpora to automatically generate chunking corpus. Penn Treebank is a fairly large collection of English sentences annotated with their detailed constituent parses. In this work, we used the Penn Treebank and generated chunked sentences automatically. Below we explain the detailed corpus generation process.

For our previous SMT work, we generated a parallel corpus by manually translating a subset of English sentences in Penn Treebank to Turkish [17]. In the chunking task, we used this corpus to automatically generate chunks in Turkish sentences.

Throughout the manual translation, we had used the following constraints. We kept the same set of English tags. We did not introduce new tags either for POS labels or phrase categories. Furthermore, we constrained our translated Turkish sentences so that trees for Turkish and English sentences are permutations of each other. A human translator starts with an English tree and permutes the subtrees of the nodes recursively as necessary until he arrives at an acceptable word order for a Turkish sentence. Finally, he replaces the English word tokens at the leaves with Turkish tokens or the special *NONE* token that denotes a deletion. The Figure 1 illustrates the process between the parallel sentences in (2)

Next, we describe the steps involved in processing a Turkish parse tree to generate chunking data. We used the 8 labels in Table 1 to identify the basic categories of chunks.

Table 1. Chunk labels

Chunk label	Description
ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	General clause
CC	Coordinating conj
PP	Prepositional phrase
VG	Verb group
PUP	Punctuation

Given the parse tree of a Turkish sentence, we traverse it breadth first. For each node except the root encountered in the traversal, if the node tag is in the Table 1, we do not

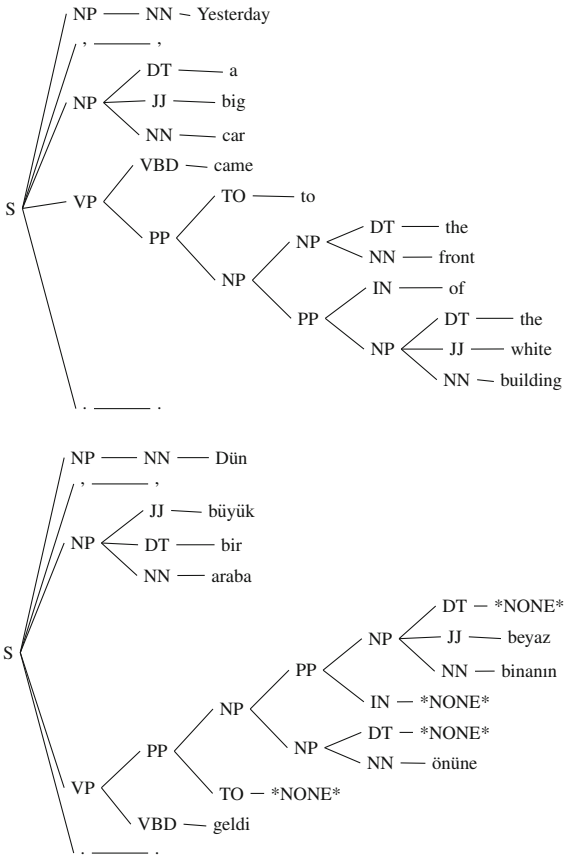


Fig. 1. The permutation of the nodes and the replacement of the leaves by the glosses or *NONE*

traverse its children further. We chunk the tokens in its leaves and label them with the node tag.

The last label VG in Table 1 is generated using the verb phrase VP with the some modifications to accommodate Turkish verb structure. In the Penn Treebank, VP often groups together the object NP and several PP and ADVP phrases under the same tree tagged as VP. Thus, all of these constituents would be put together as a VP chunk. However, for chunking, such a grouping is too coarse. We did not use VP as a chunk in our data. Instead, we first automatically identified and extracted the verb under VP and chunked it as a verb group, VG. Then, we identified the remaining subtrees of VP tree as chunks with their own labels by traversing it breadth first as above. For example, for the tree in Figure 1, we extract "geldi" as the verb group and extract out of VP subtree the phrase "beyaz binanın önüne" as PP.

In the Penn Treebank, syntactic tag set includes several distinct tags, SBAR, SBARQ, SINV, SQ, for subordinate clauses, question and inversion constituents etc. For chunking, we lumped these categories under a single category S.

We had about 5K sentences to begin with. Many of those are not full sentences but fragments. In generating the chunking corpus, we used only the full sentences. This reduced our corpus to about 4K sentences. We used this reduced set of Turkish parse trees to generate chunking data for training and testing. We confined our selection so that our training set is derived from the training subset of the Penn Treebank. For our test set, we combined the development and the test subsets derived from the Penn Treebank. As a result, we generated 3681 training sentences and 723 test sentences for our chunking task.

4 Chunking Levels

We treat the chunking problem as one of sequence labeling. We divide the labels into three major levels of complexity.

In the first level, we have only B and I labels. Thus, every chunk has a beginning token labeled B and the rest of its tokens are labeled I. We do not need to use O label at this level as all the tokens in the sentence belong to at least one chunk.

In the second level, we used the specific types of chunks that each token belongs to. The set of labels are given in Table 1. So, for example, for a NP chunk, we label its initial token as B-NP and the rest as I-NP. We also used PUP to label punctuation marks.

In the third level, the labels of the second category are augmented with the semantic roles. For example, NP-SBJ marks a noun phrase which functions as the subject of a predicate. There are 16 such role labels.

In the treebank, one basic phrase label such as ADVP might cary several simultaneous semantic tags. We used only the first of those and discarded the rest. Also, we discarded the numeric tags identifying the relations among phrases. The set of augments are given in Table 2.

5 CRFs for Chunking

To model the statistical relations among the sentence tokens and the sequence of output labels, we used conditional random fields, (CRF), [18]. CRFs have proven to be powerful tools to model the conditional probability of output label sequence given a sequence of input word tokens in a sentence.

Let $x = (x_1, x_2, \ldots, x_n)$ be an input sequence of tokens in a Turkish sentence and $y = (y_1, y_2, \ldots, y_n)$ be a candidate sequence of output labels. For example, for the first level of labels explained in the previous section, we have $y_i \in \{B, I\}$.

The probability of Y given X is expressed as

$$P(y|x) = \frac{1}{Z(x)} \exp(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y, x, i)),$$

Table 2. Semantic roles and functions

Role identifier	Description
-SBJ	Surface subject
-TMP	Temporal phrases
-LOC	Location
-DIR	Direction and trajectory
-PRD	Non VP predicates
-CLR	Closely related
-MNR	Manner
-TPC	Topicalized and fronted constituents
-EXT	Spatial extent of an activity
-NOM	Non NPs that function as NPs
-PUT	Marks the locativecomplement of put
-LGS	Logical subjects in passives
-TTL	Titles
-VOC	Vocatives
-DTV	Dative object
-PRP	Purpose and reason

where $f_j(y, x, i)$ is the j^{th} feature function corresponding to the token i in the sentence and $Z(x)$ is a normalization factor.

We used `wapiti` for the implementation of CRF, [19]. `wapiti` provides a fast training and labeling and uses the standard feature templates used by popular implementations like `CRF++`. The default options of `wapiti` uses l_1 regularization for fast convergence.

6 Chunking Features

An analysis at the syntactic level of a Turkish sentence needs to use the morphemes of the words as the morphological structure of Turkish words are closely related their syntactic functions in the sentence. For agglutinative languages like Turkish, morphological analysis is an essential early step in chunking as well as any other type of analysis. Turkish words may be quite long and contain a mixture of derivational and inflectional morphemes. In chunking, we use the inflectional morphemes and the word root.

There are a few available tools that perform automatic morpheme decomposition with a high level of accuracy, [20]. In our work, we used our own FST based morphological analyzer together with manual disambiguation. Thus, using the gold morphology, our chunking performance is isolated from the errors in morphological analysis.

In Turkish, cases indicate the syntactic function of a noun in the sentence. The case markings are suffixed to the heads of phrases. Thus, the presence of a case marking on a noun indicates that the phrase ends at that noun. Exception to this heuristic is the presence of genitive marking. In Turkish, genitive marking identifies the possessor in the noun compound. Thus, genitive is usually found when its noun is not the end of its phrase.

The basic features are the tokens themselves. This basic choice defines as many features as there are distinct word tokens. Contextual features for the token are inferred from the tokens around it.

We also tried the POS tags of the tokens as features and include contextual POS tags as well.

A summary of the features that we used in our training and test is given in Table 3.

Most of the features in the table are self explanatory. The feature F18 includes a binary feature A_i indicating whether the next root is an auxiliary Turkish verb. In Turkish, it is very common to construct two-word verbs by combining a noun or adjective with verbs "et" (do), "yap" (do) and "ol" (be). The variable A_i identifies the presence of one these auxiliary roots in the i^{th} word.

Table 3. Features

Identifier	Feature	Definition
F0	p_i	Current POS
F1	p_{i+1}	Next POS
F2	p_{i-1}	Previous POS
F3	c_i	Current case
F4	c_{i+1}	Next case
F5	c_{i-1}	Previous case
F6	g_i	Current has genitive marking, binary
F7	g_{i-1}	Previous has genitive marking, binary
F8	s_i	Current has possessive marking, binary
F9	s_{i+1}	Next has possessive marking, binary
F10	s_{i-1}	Previous has possessive marking, binary
F11	r_i	Current root
F12	r_{i+1}	Next root
F13	r_{i-1}	Previous root
F14	$U_i p_i p_{i-1}$	Current initial case, current and previous POS's
F15	$c_{i-1} s_i p_{i-1}$	Previous case, current has possessive, previous POS
F16	$c_i s_{i+1}$	Current case, next has possessiv
F17	$r_{i+1} p_{i+1}$	Next root , next POS
F18	$A_{i+1} p_{i-1}$	Next has auxiliary verb, previous POS

7 Experiments

For CRF based tagging, we use the features in Table 3. In listing the scores, P denotes the precision, R, recall and F, the F-measure.

For the first level of granularity, the possible output labels are B and I. The baseline is when we use only the tokens of the sentence as features. This choice of feature set gives the token level baseline accuracy of 0.68.

When we use the full set of features in Table 3, we obtain the results given in Table 4, broken down into performances for each label. The token level accuracy is 0.88.

Table 4. Results when only the boundaries of the chunks are identified

Tags	P	R	F
B	0.88	0.89	0.88
I	0.88	0.88	0.88

In the second level, we try to identify the type of each chunk. For this, we use the output labels such as B-NP, I-NP etc. Considering all of the 9 tags in Table 1, we have 17 possible output classes for the CRF tagger. As the number of classes increase, the small data size starts to become a problem in training. In order to mitigate the effects of data sparsity, we first analyzed our training data and kept only the most frequent labels, grouping the rest under a common label, O. The distribution of the labels in this new setup is given in Table 5. There are a total of 33,101 tokens in the training set. Note that the most common 5 labels make up the 94% of all the labels.

Table 5. Distribution of labels

Chunk label	**Percentage**
NP	35.9
VG	15.5
PUP	15.3
S	14.3
PP	12.4
ADVP	3.9
ADJP	1.5
CC	1.2

We drop the last three labels and classify them as O. The results under this new setup are given in Table 6. The token level accuracy is 0.66.

Table 6. Performance of the tagger broken down into chunk types.

Tags	P	R	F
B-NP	0.68	0.77	0.72
I-NP	0.56	0.78	0.65
B-VG	0.78	0.80	0.79
I-VG	0.64	0.38	0.47
PUP	0.96	0.99	0.98
B-S	0.43	0.21	0.28
I-S	0.49	0.38	0.43
B-PP	0.44	0.32	0.37
I-PP	0.56	0.45	0.50
O	0.62	0.58	0.60

As expected, the token level accuracy is drastically lower in this level.

Starts of the NPs are identified with fairly high accuracy. Inside the NP the performance somewhat declines. For VG, again the starts are handled better. This is somewhat expected as most verb groups in Turkish contain a single word. The low accuracy of S label is somewhat expected as S actually is a crude chunk whose boundaries are delineated by other chunks such as VG and NP.

To further analyze the source of errors, we constructed the confusion matrix for the output labels. The matrix is given in Table 7.

Table 7. Confusion matrix for the second level of granularity

	B-NP	I-NP	B-VG	I-VG	PUP	B-S	I-S	B-PP	I-PP	O
B-NP	0	**79**	8	3	3	32	22	27	1	14
I-NP	**32**	0	12	36	**11**	3	**86**	6	72	32
B-VG	**26**	**28**	0	15	1	1	24	14	2	17
I-VG	8	**95**	59	0	5	0	45	3	22	15
PUP	0	6	0	0	0	0	1	0	0	0
B-S	90	30	2	1	3	0	21	7	1	14
I-S	28	**279**	34	21	**15**	8	0	14	50	41
B-PP	76	32	4	0	1	9	26	0	13	15
I-PP	11	**184**	7	4	0	1	55	17	0	12
O	28	59	13	7	0	4	36	18	25	0

Looking at the confusion matrix, PUP column indicates that, in some error cases, other labels are predicted as punctuation when they are not. PUP is a label that is easy to learn by the CRF. However, the surrounding context seems to confuse the learner. This usually happens when a punctuation occurs within a noun phrase or subordinate sentence.

We also see that noun phrases are the source of many errors. In particular, S and NP are confused with each other. This is due the presence of NP's in the subordinate sentence boundaries. Similarly, NP frequently occurs within PP's which confuses the identification PP boundaries.

Another interesting source of errors is related to the verb group, VG. In many cases, Turkish verb group is a two word compound formed by a noun and verb such as "yardım ettim", "I helped". The first noun confuses the verb group with noun phrase. Similarly, VG-S confusion is a source of error. Again, NP's at the subordinate sentence boundaries confuse the VG identification.

In the last set of experiments, we used the augmented labels. Obviously, this increases the number of labels and as a result decreases the accuracy for each label. The sparsity becomes more prominent. As in the previous granularity level, we analyze the label counts in the training set and keep the most frequent labels as distinct classes and lump the rest together as O class. Instead of giving a large table of percentages, here we shortly describe the distribution. There are 53 distinct labels (not counting the B- and I- distinction). 17 of those make up 95% of all the labels. So we collect under O, less frequent 36 labels making up the remaining 5% of all the labels.

Table 8. Performance when the semantic roles and functions are identified

Tags	P	R	F
B-NP-SBJ	0.73	0.93	0.82
I-NP-SBJ	0.51	0.71	0.59
B-VG	0.70	0.82	0.76
I-VG	0.52	0.39	0.45
PUP	0.94	1.00	0.97
B-NP	0.20	0.17	0.18
I-NP	0.30	0.29	0.29
B-S	0.35	0.20	0.25
I-S	0.37	0.40	0.39
B-NP-PRD	0.11	0.05	0.07
I-NP-PRD	0.27	0.16	0.20
B-PP-CLR	0.24	0.13	0.17
I-PP-CLR	0.32	0.18	0.23
B-PP	0.47	0.13	0.20
I-PP	0.46	0.39	0.42
B-PP-DIR	0.68	0.61	0.64
I-PP-DIR	0.81	0.56	0.66
B-ADVP	0.55	0.59	0.57
I-ADVP	0.55	0.32	0.41
B-PP-LOC	0.28	0.23	0.25
I-PP-LOC	0.53	0.30	0.38
B-S-TPC	0.19	0.17	0.18
I-S-TPC	0.38	0.23	0.29
B-PP-TMP	0.61	0.25	0.35
I-PP-TMP	0.46	0.40	0.43
B-ADJP-PRD	0.08	0.03	0.05
I-ADJP-PRD	0.36	0.19	0.25
B-ADVP-TMP	0.68	0.45	0.54
I-ADVP-TMP	0.60	0.41	0.49
O	0.48	0.45	0.46

The results under this setup are given in Table 8. We have a token level accuracy of 0.59. We did not include the scores for labels that never occur in the test set.

Again, the highest performance is observed for the starts of subject noun phrases and verb groups. Interestingly, B-NP-SBJ is even higher than the accuracy of B-NP in the previous experiment. This might be due the bias created by the unmarked SOV order of Turkish sentences which places the subjects at the start of the sentences.

Again the most common source of errors is the confusion of NP's with other chunks. In particular, NP-SBJ is often confused with the generic class NP. Another interesting point to note is the difference between the performances of PP-DIR and PP. The learner detects PP-DIR better using the ablative and dative case markings.

The full confusion matrix for the third level of granularity is given in the supplementary materials of this paper.

8 Conclusion

In this work, we attempted a CRF based approach to Turkish chunking. Using the morpheme level features, we reported performances for different levels of chunk identification.

We used a novel approach to generate training and test data by leveraging the translated trees of the Penn Treebank. In previous works on Turkish chunking, dependency treebank was used [14]. Although we used a moderate sized data set, our approach in data generation is general enough to increase the size of the chunking corpus as more translation data becomes available.

Compared to previous works on Turkish chunking, our present work is the first attempt to solve the general chunking problem. In [12], only the NP chunks are detected using hand-crafted rules. [13] considers verb chunks only. All the other tokens in a sentence are considered out of chunk. Moreover, [13] uses dependency labels of word tokens as a feature. This requires an accurate dependency parser as a preliminary stage of the chunking. In our work, no such assumptions are made. Of course, such a general approach reflects negatively on the performance scores.

Many of our features in Table 3 have previously been used in similar chunking tasks for other languages the literature. However, we believe their application to Turkish is novel. In particular, our use of genitive and possessive markings as features is new.

An obvious direction for improving the results is designing better features to reflect the syntactic dependencies of the words in a Turkish chunk. Rather than using the given POS and case tags, one can define custom categories to reflect the coherence of nearby words. We actually used such features using the presence or absence of genitive and possessive markings. Another approach would be to design features to discriminate most frequently confused elements such as NP-PP and VG-PP. Such features would have to reflect the peculiar structure of Turkish syntactic constituents.

References

1. Abney, S.: Parsing by chunks. In: Principle-Based Parsing, pp. 257–278. Kluwer Academic Publishers (1991)
2. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall Series in Artificial Intelligence, 2 edn. Prentice Hall (2009)
3. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Third ACL Workshop on Very Large Corpora, pp. 82–94 (1995)
4. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, NAACL 2001, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2001)
5. Zhang, T., Damerau, F., Johnson, D.: Text chunking based on a generalization of winnow. J. Mach. Learn. Res. 2, 615–637 (2002)
6. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the CoNLL-2000 shared task: Chunking. In: Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning, ConLL 2000, pp. 127–132. Association for Computational Linguistics, Stroudsburg (2000)

7. Park, S.B., Zhang, B.T.: Text chunking by combining hand-crafted rules and memory-based learning. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, ACL 2003, vol. 1, pp. 497–504. Association for Computational Linguistics, Stroudsburg (2003)
8. Lee, Y.-H., Kim, M.-Y., Lee, J.-H.: Chunking using conditional random fields in korean texts. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 155–164. Springer, Heidelberg (2005)
9. Gune, H., Bapat, M., Khapra, M.M., Bhattacharyya, P.: Verbs are where all the action lies: Experiences of shallow parsing of a morphologically rich language. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010, pp. 347–355. Association for Computational Linguistics, Stroudsburg (2010)
10. Chen, W., Zhang, Y., Isahara, H.: An empirical study of chinese chunking. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL 2006, pp. 97–104. Association for Computational Linguistics, Stroudsburg (2006)
11. Sun, G.L., Huang, C.N., Wang, X.L., Xu, Z.M.: Chinese chunking based on maximum entropy markov models. International Journal of Computational Linguistics & Chinese Language Processing 11, 115–136 (2006)
12. Kutlu, M.: Noun phrase chunker for Turkish using dependency parser. Master's thesis, Sabancı University (2010)
13. El-Kahlout, İ.D., Akın, A.A.: Turkish constituent chunking with morphological and contextual features. In: Gelbukh, A. (ed.) CICLing 2013, Part I. LNCS, vol. 7816, pp. 270–281. Springer, Heidelberg (2013)
14. Atalay, N.B., Oflazer, K., Say, B.: The annotation process in the Turkish treebank. In: 4th International Workshop on Linguistically Interpreted Corpora (2003)
15. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics 19, 313–330 (1993)
16. Kornfilt, J.: Turkish. Routledge (1997)
17. Yıldız, O.T., Solak, E., Görgün, O., Ehsani, R.: Constructing a Turkish-English parallel treebank. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 112–117. Association for Computational Linguistics, Baltimore (2014)
18. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, NAACL 2003, pp. 134–141. Association for Computational Linguistics, Stroudsburg (2003)
19. Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 504–513. Association for Computational Linguistics (2010)
20. Hakkani-Tur, D., Oflazer, K., Tür, G.: Statistical morphological disambiguation for agglutinative languages. Computers and the Humanities (2002)

Syntax and Parsing

Statistical Arabic Grammar Analyzer

Michael Nawar Ibrahim

Department of Computer Engineering, Cairo University
Cairo, Egypt
michael.nawar@eng.cu.edu.eg

Abstract. The grammar analysis is considered one of the complex tasks in the Natural Language Processing (NLP) field, since it determines the relation between the words in the sentence. This paper proposes a system to automate the grammar analysis of Arabic language sentences (Sentence Grammar Analysis, إعراب الجمل, <ErAb Aljml). The task of Arabic grammar analysis has been divide into three sub-tasks, of determining the grammatical tag, the case, and the sign of each token in the level of the sentence. For the task of Arabic grammar analysis, a dataset has been compiled and a statistical system that assigns an appropriate tag, case and sign has been implemented. The proposed system has been tested and the experiments show that it achieves a 89.74% token accuracy and a 63.56% overall sentence accuracy and it has the potential to be further improved.

1 Introduction

Generally, the grammar analysis is the process of determining the grammatical role of each word in a sentence, but in Arabic, the grammar analysis includes an additional task which is the determination of the case ending diacratization of each word too. Therefore, the Arabic grammar analysis could not be implemented using simple parsing techniques. Another property in the Arabic grammar analysis, that it is flatter than regular parsing tree structures because they lack a finite verb phrase forms. Once the Arabic grammar analysis of a sentence is completed many problems can be simply solved. Nawar and Ragheb [16] used grammar analysis in the task of Arabic text correction. They assigned a simple grammatical tag to some words in the text, and based on the tag they determine whether the word contains a grammatical error or not, and finally they correct it. Other tasks also could be simply solved, such as automatic diacritics, question answering systems and accurate translation.

As example for the task of grammar analysis, lets consider the following sentence to be grammatically analyzed: (الأولاد يلعبون في حديقة المدرسة مع زملائهم., Al>wlAd ylEbwn fy Hdyqp Almdrsp mE zmlA}hm, The boys are playing with their colleagues in the school garden.). The complete grammatical analysis for the sentence is shown in table 1.

The proposed system covers the basic grammar tags for verbal and nominal sentence. However, it has the following limitations:

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 187–200, 2015.
DOI: 10.1007/978-3-319-18111-0_15

Table 1. Example of Sentence Grammatical Analysis

word	Grammatical Analysis
الأولاد	مبتدأ مرفوع وعلامة الرفع الضمة
Al>wlAd	mbtd> mrfwE wElAmp AlrfE AlDmp
The boys	subject, nominative, sign: damah
يلعبون	فعل مضارع مرفوع وعلامة الرفع ثبوت النون
ylEbwn	fEl mDArE mrfwE wElAmp AlrfE vbwt Alnwn
are playing	present verb, nominative, sign: waw and noon
في	حرف جر مبني لا محل له من الاعراب
fy	Hrf jr mbny lA mHl lh mn AlAErAb
in	preposition, uninflected, sign: no sign
حديقة	اسم مجرور وعلامة الجر الكسرة
Hdyqp	Asm mjrwr wElAmp Aljr Alksrp
garden	genitive noun, genitive, sign: kasra
المدرسة	مضاف إليه مجرور وعلامة الجر الكسرة
Almdrsp	mDAf <lyh mjrwr wElAmp Aljr Alksrp
the school	possessive, genitive, sign: kasra
مع	ظرف مبني في محل نصب
mE	Zrf mbny fy mHl nSb
with	circumstance, uninflected accusative, sign: no sign
زملاء	مضاف إليه مجرور وعلامة الجر الكسرة
zmlA'	mDAf <lyh mjrwr wElAmp Aljr Alksrp
colleagues	possessive, genitive, sign: kasra
+هم	ضمير مبني في محل جر بالإضافة
+hm	Dmyr mbny fy mHl jr bAl<DAfp
theirs	possessive, uninflected genitive, sign: no sign
.	علامة ترميز لا محل لها من الاعراب
.	ElAmp trmyz lA mHl lhA mn AlAErAb
.	punctuation, no case, no sign

- As any statistical system, it is limited by the grammatical tags, cases, and signs in the annotated data
- The system analyze the grammar of a complete and correct sentence whether morphologically or grammatically; and error correction is not included right now.
- As a nature of Arabic, a word could have multiple meanings based on its diacratization, for example the verb could be in passive or active voice e.g., (ضرب, drb) could be read as (ضُرِبَ, doreb, beaten) or (ضَرَبَ, darab, beat), however the system provides one single correct grammatical analysis for each word in the sentence.
- The system provides a correct grammatical analysis for a sentence independently of its semantic meaning. In other words, the semantic analysis is not verified.

For the training and the evaluation of the system, the Arabic Treebank part 1 [14] that consists of 140k words corresponding to 168k tokens is used. Also, an

additional datatest from 12340 sentences that consist of 82430 words corresponding to 88501 tokens, have been annotated for the training and the evaluation of the system.

This paper is organized as follow, in section 2, an overview of the related work in the field of Arabic NLP and the Arabic grammar analysis is discussed. In section 3, the system architecture and its main components are explained. The annotated data and the evaluation process are presented in section 4. Finally, concluding remarks and future work are presented in section 5.

2 Related Work

For the last two decades, most of the work on the Arabic natural language processing focused on simple tasks like morphological analysis, and part of speech tagging. Multiple systems were implemented like ([2], [3], and [5]). Frameworks that provides multiple tasks for Arabic NLP were implemented.

One framework is MADA+TOKAN ([11] and [17]) is one of the most famous Arabic NLP systems. It provides a framework for morphological disambiguation, POS tagging, diacritization, lexicalization, lemmatization stemming and other tasks. It consists of two main parts: MADA, a system for Morphological Analysis and Disambiguation for Arabic, and TOKAN, a general tokenizer for MADA-disambigauted text. In simple words, the MADA system along with TOKAN provide one solution to different Arabic NLP problems.

Another framework for different Arabic NLP problems is the AMIRA system [6]. It is a framework for Arabic tokenization, POS tagging, Base Phrase Chunking, and Named Entities Recognition. The AMIRA toolkit includes a clitic tokenizer (TOK), part of speech tagger (POS) and base phrase chunker (BPC) - shallow syntactic parser. The technology of AMIRA is based on supervised learning with no explicit dependence on knowledge of deep morphology; hence, in contrast to systems such as MADA, it relies on surface data to learn generalizations.

A final framework is the Arabic NLP tools developed by Stanford natural language processing group. The developed Arabic NLP products are a word segmenter [9], part-of-speech tagger [17] and a probabilistic parser [10] the data set used is the Penn Arabic Treebank [14].

Although of the importance of Arabic grammar analysis, few researchers attempted to extract grammar analysis for Arabic text. However, the research interested in extracting the grammar analysis follows two main approaches. The first depends on the deep knowledge of Arabic morphology and grammatical rules and usually apply rule-based techniques; while the second make use of annotated data and try to assign an appropriate grammatical tag to each word using parsing techniques.

As an example of rule-based frameworks, Al-Daoud and Basata [1] proposed a system to automate the grammar analysis of Arabic language sentences in general. Their system focuses on verbal sentences, and they claimed that it could be extended to any Arabic language sentence. Moreover, this system has

the limitations that the entered sentences are correct lexically and grammatically, and the verbs are always in the active voice.

Ibrahim et al. [13] proposed a hybrid system between learning-based approaches and rule-based approaches for Arabic grammar analysis. The proposed system provides an acceptable accuracy and could be simply implemented. Although the use of learning parts in the proposed system, it requires deep knowledge of Arabic as any rule-based system.

Attia ([2], [3]) used parsing-based technique to disambiguate Arabic text. He built an Arabic parser using Xerox linguistics environment to write grammar rules and notations that follow the LFG formalisms. Attia tested his approach on short sentences randomly selected from a corpus of news articles, and he claimed an accuracy of 92%.

Habash and Roth [12] construct The Columbia Arabic Treebank (CATiB). Columbia Treebank is a database of syntactic analyses of Arabic sentences. CATiB contrasts with previous approaches to Arabic Treebanking in its emphasis on speed with some constraints on linguistic richness. Two basic ideas inspire the CATiB approach: no annotation of redundant information and using representations and terminology inspired by traditional Arabic syntax. So the task of grammar analysis can be done by applying a simple parsing approach.

Few people work on grammar analysis of classical Arabic, Duke and Buckwalter [8] constructed the Quranic Arabic Dependency Treebank (QADT), which is an annotated grammar resource consisting of 77,430 words of Quran. This project differs from other Arabic treebanks by providing a language model based on traditional Arabic grammar.

After exploring and analyzing the research in the area of Arabic grammar analysis, it appears that most of it concentrated on short sentences and used hand-crafted grammars, which are time-consuming to produce and difficult to scale to unrestricted data. Also, these approaches used traditional parsing techniques like top-down and bottom-up parsers demonstrated on simple verbal sentences or nominal sentences with short lengths. To simplify the task of Arabic grammar analysis, in this paper the task of grammar analysis is divided into three classification sub-tasks, and a dataset for these classifications is presented.

3 The Proposed System

The proposed system is divided into five main components: Morphological analyzer, stemmer, part of speech tagger (POS tagger), base phrase chunker, and finally the grammar analyzer. The Arabic text is first processed by the stemmer. The stemmer separates proclitics and enclitics of each word in the text. Then the POS tagger assigns an adequate POS tag to each token. Then, the base phrase chunker groups words belonging to the same phrases. Additional morphological information is extracted for each word using the morphological analyzer. Finally, the Arabic grammar analyzer uses the extract information to assign a grammatical tag, case and sign for each token in the text.

3.1 Morphological Analyzer

The morphological analyzer used in the system is based on BAMA-v2.0 (Buckwalter Arabic morphological analyzer version 2.0) [4]. The extended morphological analyzer provides additional features like the extraction of the pattern of the word. For example, the pattern of (كاتب,kAtb, writer) is (فاعل,fAEl) and the pattern of (مكتب,mktb, office) is (مفعل,mfEl). For the extraction of the word pattern, an appropriate pattern is assigned for each stem in the stem table of the morphological analyzer, then the word pattern is determined based on its stem pattern, prefix and suffix. For example, if the pattern (ستفعل, stfEl) is assigned for the stem (ستخدم, stxdm) , then the word (مستخدمين, mstxdmyn, users) that have a prefix (م,m) and a suffix (ين, yn) will have the pattern (مستفعلين, mstfElyn), and the word (يستخدمون, ystxdmwn, they use) that have a prefix (ي,y) and a suffix (ون, wn) will have the pattern (يستفعلون, ystfElwn). A word will not have a pattern assigned to it if the word is Arabized (nouns borrowed from foreign languages) like (كمبيوتر, kmbywtr, computer) or (أمريكا, >mrykA, America), or if the word is fixed (words used by Arabs, and do not obey the Arabic derivation rules) like (هذا, h*A, this) or (كل, kl, every).

Also, the morphological analyzer could be used to extract the root of the word. For example, the root of (كاتب,kAtb, writer)) is (كتب,ktb) and the root of (مكتب,mktb,office) is (كتب,ktb). For the extraction of the word root, an appropriate root is assigned for each stem in the stem table of the morphological analyzer, then the word root is determined based on its stem. For example, if the root (خدم, xdm) is assigned to the stem (ستخدم, stxdm), then the root of the word (مستخدمين, mstxdmyn, users) will be (خدم, xdm), and the root of the word (يستخدمون, ystxdmwn, they use) will be (خدم, xdm). No root is assigned to a word if the word is Arabized, or if the word is fixed.

And finally, the morphological analyzer is developed to determine if a word is definite or not. To determine the definiteness of a word, the morphological analyzer extract all possible analyses for a word, then check if all of them are definite then return definite, if all of them are indefinite then return indefinite, and if the result is a mix between definite and indefinite then return undetermined. The morphological analyzer uses the same process to determine if a word is masculine or feminine, and if it is plural or dual or singular.

3.2 Stemmer

In this task, the stemmer takes an input of raw text, without any processing, and assigns each character the appropriate tag from the following tag set B-PRE1, B-PRE2, B-WRD, I-WRD, B-SUFF, I-SUFF. Where I denotes inside a segment, B denotes beginning of a segment, PRE1 and PRE2 are proclitic tags, SUFF is an enclitic, and WRD is the stem plus any affixes and/or the determiner Al. These tags are similar to the tags used by Diab et al. [7].

A feature used in the stemmer is the binary feature introduced by Nawar [15]. The binary feature has a length of 6 bits where each bit in the feature is

mapped to one of the 6 tags in the tokenization tag set. A bit is set if at least one analysis in the morphological analyses of the word, the character is assigned the tag corresponding to the bit. For example the word (وحيد, wHyd) has two possible tokenization schemes: (و حيد, w Hyd, and move away) or (وحيد, wHyd, Wahid (proper noun)); then (و, w) could be (B-PRE1 or B-WRD) then in the binary feature of the character there will be 2 bits set which map to B-PRE1 and B-WRD, (ح, H) could be (B-WRD or I-WRD) then in the binary feature of the character there will be 2 bits set which map to B-WRD and I-WRD, (ي, y) and (د, d) could be only (I-WRD), then in the binary feature of the characters there will be only one bit set which map to I-WRD. Table 2 shows the binary feature of each character of the word (وحيد, wHyd).

Table 2. Example of Stemming Binary Feature

Arabic Letter	Transliterated Letter	Binary Feature					
		B-PRE1	B-PRE2	B-WRD	I-WRD	B-SUFF	I-SUFF
و	w	1	0	1	0	0	0
ح	H	0	0	1	1	0	0
ي	y	0	0	0	1	0	0
د	d	0	0	0	1	0	0

The classifier training and testing data could be characterized as follow:

- **Input:** A sequence transliterated Arabic characters processed from left-to-right with break markers for word boundaries.
- **Context:** A fixed-size window of -5/+5 characters centered at the character in focus.
- **Features:** All characters and previous tag decisions within the context, and the binary feature of each character with the context.
- **Classifier:** CRF suite classifier.
- **Data:** Arabic Treebank part 1.

3.3 Part of Speech Tagger

In this task, the POS tagger takes an input of tokenized text, and it assigns each token an appropriate POS tag from the Arabic Treebank collapsed POS tags, which comprises 24 tags as follows: {ABBREV, CC, CD, CONJ+NEG PART, DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO FUNC, NUMERIC_COMMA, PRP, PRP$, PUNC, RB, UH, VBD, VBN, VBP, WP, WRB }.

A feature used in the POS tagger is the binary feature also introduced by Nawar [15]. The binary feature has a length of 24 bits where each bit in the feature is mapped to one of the 24 tags in the collapsed POS tag set. A bit is set when its corresponding tag exists in the morphological analysis of a token. For example the word (كتب, ktb) has 3 different reduced POS tags: VBD then it will

mean (write), VBN then it will mean (be written), and NN then it will mean (book); so there will be 3 bits set to one in the binary feature of the (كتب, ktb) word corresponding to VBD, VBN and NN. While you can find a word like (الولد, Alwld) has only one reduce POS tag which is NN and it have only one meaning the boy. And if the word is not analyzed by the morphological analyzer (out of vocabulary) like the word (الفالوجة, AlfAlwjp) which is a village in Palestine, then there will be 5 bits set in the binary feature which map to JJ, NN, NNS, NNP, and NNPS. In table 3, you can find the binary feature for the words of the sentence (كتب الولد الدرس, ktb Alwld Aldrs, The boy wrote the lesson).

Table 3. Example of POS Tagging Binary Feature

Arabic Word	Transliterated Word	Binary Feature					
		VBD	VBN	NN	JJ	NNS	...
كتب	ktb	1	1	1	0	0	0
الولد	Alwld	0	0	1	0	0	0
الدرس	Aldrs	0	0	1	0	0	0

The classifier training and testing data could be characterized as follow:

- **Input:** A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- **Context:** A window of -2/+2 tokens centered at the focus token.
- **Features:** Every character N-gram, N<=4 that occurs in the focus token, the 5 tokens themselves, POS tag decisions for previous tokens within context, and the binary feature of the words within the context.
- **Classifier:** CRF suite classifier.
- **Data:** Arabic Treebank part 1.

3.4 Base Phrase Chunker

In this task, the base phrase chunker takes an input of tokenized text, and it assigns each token an appropriate Base Phrase Chunk tag from the Arabic Treebank collapsed BPC tags. Nine types of chunked phrases are recognized using a phrase BIO tagging scheme, Inside (I) a phrase, Outside (O) a phrase, and Beginning (B) of a phrase. The 9 chunk phrases identified for Arabic are PP, PRT, NP, SBAR, INTJ, and VP. Thus the task is a one of 12 classification task (since there are I and B tags for each chunk phrase type except PRT, and a single O tag). The classifier training and testing data could be characterized as follow:

- **Input:** A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- **Context:** A window of -2/+2 tokens centered at the focus token.

- **Features:** Every character N-gram, N<=4 that occurs in the focus token, the 5 tokens themselves, POS tag decisions for previous tokens within context and the previous Base phrase tag.
- **Classifier:** CRF suite classifier.
- **Data:** Arabic Treebank part 1.

3.5 Grammatical Analyzer

The task of grammatical assign a complete analysis to a word is a really complex task. This task is reduced to assign a grammatical tag, a case and a sign to each word. Then after this tags are assigned to a word, a sentence is assigned to each word that represent the full Arabic grammatical analysis.

Arabic grammatical tags: present verb (فعل مضارع, fEl mDArE), imperative verb (فعل أمر, fEl >mr), past verb (فعل ماضي, fEl mADy), doer (فاعل, fAEl), direct object (مفعول به, mfEwl bh), second direct object (مفعول به ثان, mfEwl bh vAn), subject (مبتدأ, mbtd>), predicate (خبر, xbr), ena subject (أسم إن, >sm <n), ena predicate(خبر إن, xbr <n), kan subject (أسم كان, >sm kAn), kan predicate (خبر كان, xbr kAn), apposition (بدل, bdl), adjective (نعت, nEt), conjunction (معطوف, mETwf), possessive (مضاف اليه, mDAf Alyh), genitive noun (أسم مجرور, >sm mjrwr), circumstance (ظرف, Zrf), particle ena(حرف ناسخ, Hrf nAsx), particle kan (فعل ناسخ, fEl nAsx), accusative particle (حرف نصب, Hrf nSb), jussive particle (حرف جزم, Hrf jzm), preposition (حرف جر, Hrf jr), coordinating conjunction (حرف عطف, Hrf ETf), realization particle (حرف تحقيق, Hrf tHqyq), diminishing particle (حرف تقليل, Hrf tqlyl), particle (حرف, Hrf), punctuation (علامة ترميز, ElAmp trmyz) .

Arabic cases: nominative (مرفوع, mrfwE), accusative (منصوب, mnSwb), genitive (مجرور, mjrwr), jussive (مجزوم, mjzwm), uninflected (مبني, mbny), uninflected nominative (مبني في محل رفع, mbny fy mHl rfE), uninflected accusative (مبني في محل نصب, mbny fy mHl nSb), and uninflected genitive (مبني في محل جزم, mbny fy mHl jzm), no case (بدون اعراب, bdwn AErAb).

Arabic signs: fatha (الفتحة, AlftHp), kasra (الكسرة, Alksrp), damah (الضمة, AlDmp), sukun (السكون, Alskwn), writing noon (ثبوت النون, H*f Alnwn), waw (الواو, AlwAw), ya' (الياء, AlyA'), alef (الألف, Al>lf), no sign (بدون علامة اعراب, bdwn ElAmp AErAb).

To assign an appropriate grammatical tag to the tokens, the classifier training and testing could be characterized as follow:

- **Input:** A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- **Context:** A window of -3/+3 tokens centered at the focus token.
- **Features:** Every character N-gram, N<=4 that occurs in the focus token, the 7 tokens themselves, POS tag decisions for tokens within context, the base phrase chunk for tokens within the context, the root of the words within the context,, the pattern of the words within the context, whether the word is definite or not, whether the word is feminine or not, and whether the word is plural or dual or singular.

- **Classifier:** CRF suite classifier.
- **Data:** The annotated data.

To assign an appropriate case to the tokens, the classifier training and testing could be characterized as follow:

- **Input:** A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- **Context:** A window of 4 tokens that include the token and its previous 3 tokens.
- **Features:** Every character N-gram, N<=4 that occurs in the focus token, the 4 tokens themselves, POS tag decisions for tokens within context, the base phrase chunk for tokens within the context, the root of the words within the context, the pattern of the words within the context, whether the word is definite or not, whether the word is feminine or not, and whether the word is plural or dual or singular.
- **Classifier:** CRF suite classifier.
- **Data:** The annotated data.

To assign an appropriate sign to the tokens, the classifier training and testing could be characterized as follow:

- **Input:** A sequence of transliterated Arabic tokens processed from left-to-right with break markers for word boundaries.
- **Context:** The token itself.
- **Features:** Every character N-gram, N<=4 that occurs in the focus token, the token, whether the word is feminine or not, and whether the word is plural or dual or singular.
- **Classifier:** CRF suite classifier.
- **Data:** The annotated data.

To assign an appropriate complete analysis for each word a simple rule based system that merge the word tag, case, and sign is implemented. For example, if the word grammatical tag is subject, its case is nominative, and its sign is damah, the analysis (مبتدأ مرفوع وعلامة الرفع الضمة, mbtd> mrfwE wElAmp AlrfE AlDmp) is assigned to the sentence.

4 Evaluation of the System

4.1 The Annotated Data

The annotated data contains 12340 sentences that consist of 82430 words corresponding to 88501 tokens, and it is available on http://www.CICLing.org/2015/data/168. The sentence of the data were collected from newspapers. Table 4 contains some important facts about the dataset.

The average tokens per sentence is 7.17 tokens with the median being 7. The average sentence contains 6.68 words with the median being 7. Figure 1 shows

Table 4. Important Data Statistics

Number of Sentences	12340
Number of words	82430
Average number of words per sentence	6.68
Median number of words per sentence	7
Minimum number of words per sentence	3
Maximum number of words per sentence	11
Number of tokens	88501
Average number tokens per sentence	7.17
Median number of tokens per sentence	7
Minimum number of tokens per sentence	3
Maximum number of tokens per sentence	15

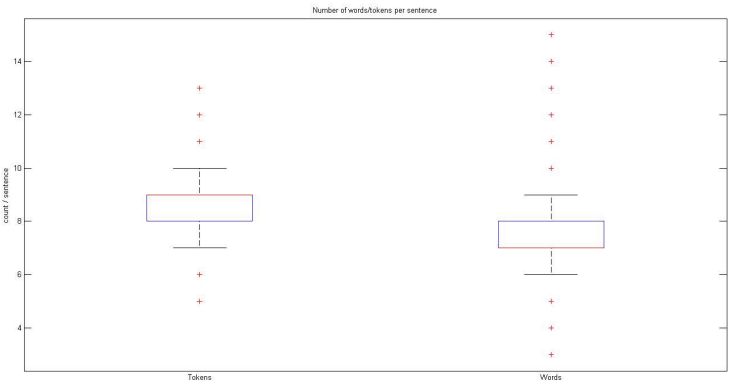


Fig. 1. Number of words/tokens per sentence

a box plot of the number of tokens per sentence and the number of words per sentence.

In table 5, the grammatical tags, and their frequencies in the dataset is presented. Also, table 6 shows the frequencies of Arabic cases of tokens in the dataset. Finally, the case ending signs and their frequencies are presented in table 7.

4.2 The Evaluation Results

For the evaluation of these experiments, k-fold algorithm was used by setting the parameter k to five so the Penn Arabic tree bank part1 and the annotated data are randomly partitioned into five portions of equal size. In each iteration of the k- fold algorithm four portions were used for training the model and one portion was used for testing the model. The cross-validation process is then repeated five times (the folds), with each of the k subsamples used exactly once

Table 5. Grammatical Tags Statistics

Tag	Count
present verb	7501
past verb	94
imperative verb	8
doer	64
direct object	4970
second direct object	12
subject	12250
predicate	335
ena subject	11
ena predicate	4
kan subject	2
kan predicate	7
apposition	78
adjective	8400
conjunction	235
possessive	14992
genitive noun	13288
circumstance	327
particle ena	11
particle kan	9
accusative particle	18
jussive particle	3
coordinating conjunction	233
preposition	13290
realization particle	4
diminishing particle	5
particle	18
punctuation	12340
Total	88501

as the testing data. The five results from the folds were averaged to produce the model evaluation. Then the following performance measures are calculated for each component and finally, the overall system performance is calculated

$$macro\ average\ precision = \frac{1}{n}\sum_{i=1}^{n} precision(tag(i))$$

$$macro\ average\ recall = \frac{1}{n}\sum_{i=1}^{n} recall(tag(i))$$

$$macro\ average\ F_{(\beta=1)} = \frac{1}{n}\sum_{i=1}^{n} F_{(\beta=1)}(tag(i))$$

Table 6. Arabic Cases Statistics

Case	Count
nominative	23227
accusative	6119
genitive	32547
jussive	3
uninflected nominative	36
uninflected accusative	400
uninflected genitive	247
uninflected	13582
no case	12340
Total	88501

Table 7. Arabic Signs Statistics

Sign	Count
no sign	26605
fatha	5803
kasra	32440
damah	22808
sukun	3
waw	170
ya'	423
alef	20
writing noon	229
Total	88501

Where n is the total number of tags.

$$Accuracy = \frac{number\ of\ true\ results}{number\ of\ true\ and\ false\ results}$$

Table 8 shows the evaluation results of the Stemmer, POS tagger, Base Phrase chunker, and the 3 parts of the Arabic analyzer. The results of the stemmer and the POS tagger show significant accuracy improvement compared to the state of the art stemming and POS tagging systems. Further experiment are made on the system as a whole, complete test sentences were used as input to the system and the output of the system is observed, and it is found that the overall accuracy of tokens that have correct tag, case and sign is 89.74%, and that the overall sentence accuracy (i.e. sentence that all of its tokens are correctly analyzed) is 63.56%.

Table 8. System Components Evaluation Results

	Precision	Recall	$F_{\beta=1}$	Accuracy
Stemmer	0.9999	0.9990	0.9995	99.99%
POS tagger	0.8487	0.8123	0.8269	98.05%
BPC	0.8753	0.7448	0.7691	96.09%
Tag	0.8634	0.8513	0.8501	93.42%
Case	0.9638	0.9514	0.9616	96.32%
Sign	0.9878	0.6735	0.7420	97.58%

5 Conclusion and Future Work

In this paper a statistical system for grammatically analyze Arabic text has been proposed. The system architecture has been discussed, and its components from morphological analyzer, stemmer, POS tagger, and base phrase chunker are

described and evaluated. The proposed achieves a token accuracy of 89.74% and a complete sentence accuracy of 63.56%. Finally, the dataset constructed for the task of Arabic grammar analysis is analyzed, and its properties and statistics are explored. The current results are promising, and the system could be further improved by annotating more data, that covers grammatical tags not existing in the dataset or exist with low frequencies.

References

1. Al-Daoud, E., Basata, A.: A framework to automate the parsing of arabic language sentences. Int. Arab J. Inf. Technol. 6(2), 191–195 (2009)
2. Attia, M.: An ambiguity-controlled morphological analyzer for modern standard arabic modelling finite state networks. In: Challenges of Arabic for NLP/MT Conference, vol. 200610. The British Computer Society, London (2006)
3. Attia, M.A.: Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation. Ph.D. thesis, University of Manchester (2008)
4. Buckwalter, T.: Buckwalter arabic morphological analyzer version 2.0. linguistic data consortium, university of pennsylvania, 2002. ldc cat alog no.: Ldc2004l02. Tech. rep., ISBN 1-58563-324-0 (2004)
5. Daoud, A.M.: Morphological analysis and diacritical arabic text compression. Computer Journal of the International Journal of ACM Jordan 1(1), 41–47 (2010)
6. Diab, M.: Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. In: 2nd International Conference on Arabic Language Resources and Tools. Citeseer (2009)
7. Diab, M., Hacioglu, K., Jurafsky, D.: Automated methods for processing arabic text: from tokenization to base phrase chunking. Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer (2007)
8. Dukes, K., Buckwalter, T.: A dependency treebank of the quran using traditional arabic grammar. In: 2010 the 7th International Conference on Informatics and Systems (INFOS), pp. 1–7. IEEE (2010)
9. Green, S., DeNero, J.: A class-based agreement model for generating accurately inflected translations. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 146–155. Association for Computational Linguistics (2012)
10. Green, S., Manning, C.D.: Better arabic parsing: Baselines, evaluations, and analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 394–402. Association for Computational Linguistics (2010)
11. Habash, N., Rambow, O., Roth, R.: Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, pp. 102–109 (2009)
12. Habash, N., Roth, R.M.: Catib: The columbia arabic treebank. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 221–224. Association for Computational Linguistics (2009)
13. Ibrahim, M.N., Mahmoud, M.N., El-Reedy, D.A.: Bel-arabi: Advanced arabic grammar analyzer
14. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The penn arabic treebank: Building a large-scale annotated arabic corpus. In: NEMLAR Conference on Arabic Language Resources and Tools, pp. 102–109 (2004)

15. Nawar, M.N.: Improving arabic tokenization and pos tagging using morphological analyzer. In: Hassanien, A.E., Tolba, M.F., Taher Azar, A. (eds.) AMLTA 2014. CCIS, vol. 488, pp. 46–53. Springer, Heidelberg (2014)
16. Nawar, M.N., Ragheb, M.M.: Fast and robust arabic error correction system. In: ANLP 2014, p. 143 (2014)
17. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: 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, vol. 1, pp. 173–180. Association for Computational Linguistics (2003)

Bayesian Finite Mixture Models for Probabilistic Context-Free Grammars

Philip L.H. Yu and Yaohua Tang

The University of Hong Kong
Department of Statistics & Actuarial Science
Hong Kong, China
{plhyu,tangyh}@hku.hk

Abstract. Instead of using a common PCFG to parse all texts, we present an efficient generative probabilistic model for the probabilistic context-free grammars(PCFGs) based on the Bayesian finite mixture model, where we assume that there are several PCFGs and each of these PCFGs share the same CFG but with different rule probabilities. Sentences of the same article in the corpus are generated from a common multinomial distribution over these PCFGs. We derive a Markov chain Monte Carlo algorithm for this model. In the experiments, our multi-grammar model outperforms both single grammar model and Inside-Outside algorithm.

Keywords: Bayesian Finite Mixture Model, Phrase Parsing, MCMC.

1 Introduction

Since hand annotated corpus are available more than ever before, and easier to use for various data tasks, supervised methods for NLP tasks have become a hot topic. Many supervised methods have been introduced and achieved great progress in the NLP literature.

Although supervised methods often significantly outperform current unsupervised induction algorithms, there are still compelling motivations to continue the work on unsupervised methods. First of all, preparing training data for supervised problems requires considerable resources, including time and linguistic expertise, which are time consuming, hard, and expensive. Furthermore, the resulting hand-crafted treebank may be only applicable to a particular domain, application, or genre[1] and hence the supervised methods may be very difficult to adapt for new tasks, languages, and domains. This problem is even more complicated when we deal with languages other than English as we usually lack of the necessary resources. Consequently, it is the corpus availability that directs the research in this area. However, unsupervised methods do not need such training data, and they can also be used in many applications, for example, in primary phases of constructing large treebanks, in language modeling, and in some NLP research areas that do not require an exact grammar of sentences.

In this paper, we will present an efficient unsupervised learning algorithm for probabilistic context-free grammars(PCFGs) based on a Bayesian inference approach. Over the past few years there has been considerable interest in Bayesian inference

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 201–212, 2015.
DOI: 10.1007/978-3-319-18111-0_16

for computational linguistics, including part-of-speech tagging[2][3], phrase-structure parsing[4][5] and combinations of models[6]. Recently Bayesian inference algorithms for PCFG have also been discussed. Kurihara Kenichi et al.[7] introduced a variational Bayes algorithm for inferring PCFG using a mean field approximation. Johnson et al.[8] introduced a Markov Chain Monte Carlo algorithm. Tomoharu Iwata[9] attempted to extract hidden common syntax across languages from non-parallel multilingual corpora using variational Bayes method.

The main idea of this paper is that we assume different authors may use different grammars. It is intuitive that authors have their distinctive usage of words, phrases and sentence structures, which means different written habits. For this purpose, we propose a generative model for multi-grammar that is learned in an unsupervised fashion. A related well known model is Latent Dirichlet allocation(LDA)[10]. Johnson et al.[11] established a connection between LDA and PCFG by showing that LDA topic models can be viewed as a special kind of PCFG.

LDA is a generative probabilistic model of a corpus, which is motivated from the idea that articles have different mixed portions of topics. By developing a LDA model for grammars, we can also build a connection between PCFGs and LDA. We assume that there are several PCFGs and each of these PCFGs share the same CFG but with different rule probabilities. Sentences of the same article in the corpus are generated from a common multinomial distribution over the PCFGs, which represents the written habits of the author. We assume these grammars are generated from a prior grammar that is common across articles. The inference of the model can be performed by using Markov chain Monte Carlo(MCMC) algorithm.

The rest of this paper is structured as follows. Section 2 reviews PCFG and LDA. Section 3 introduces the proposed model and Section 4 discusses inferencing the model by Markov chain Monte Carlo algorithm. Experimental results showing the competitiveness of our model for PCFGs are presented in Section 5.

2 Background

2.1 Probabilistic Context-Free Grammars

Let $G = (\boldsymbol{N}, \sum, ROOT, \boldsymbol{R})$ be a Context-Free Grammar(CFG), where $\boldsymbol{N}$ is a finite set of nonterminal symbols, $\sum$ is a finite set of terminal symbols(disjoint from $\boldsymbol{N}$), $ROOT \in \boldsymbol{N}$ is the start symbol, and $\boldsymbol{R}$ is a finite set of rules. We assume that the grammar is in Chomsky normal form. Thus, each rule is either of the form $A \to B\,C$ or $A \to w$, where $A, B, C \in \boldsymbol{N}$ and $w \in \sum$. We use δ as a variable defined over $(\boldsymbol{N} \times \boldsymbol{N}) \cup \sum$.

A Probabilistic Context-Free Grammar $(G, \boldsymbol{\theta})$ is a pair consisting of a CFG G and a probability vector $\boldsymbol{\theta}$. $\theta_{A\to\delta}$ is the probability of rule $A \to \delta \in \boldsymbol{R}$. Basically, it's required that $\theta_{A\to\delta} \geq 0$ and that for all nonterminals $A \in \boldsymbol{N}$, $\sum_{A\to\delta\in\boldsymbol{R}} \theta_{A\to\delta} = 1$.

Let H be a prior distribution of $\boldsymbol{\theta}$. In this paper, we consider H be a product of Dirichlet distributions, with one distribution for each non-terminal $A \in \boldsymbol{N}$. H is parameterized by a positive real valued vector $\boldsymbol{\mu}$ indexed by rules in $\boldsymbol{R}$, so each rule probability $\theta_{A\to\delta}$ has a corresponding Dirichlet parameter $\mu_{A\to\delta}$. For each nonterminal $A \in \boldsymbol{N}$, $\boldsymbol{R}_A$ denote the rules with A on their left side, and let $\boldsymbol{\theta}_A$ and $\boldsymbol{\mu}_A$ refer to the component subvectors

of $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ respectively indexed by rules in $\boldsymbol{R_A}$. For each rule r, $\boldsymbol{R}(r)$ denote the rules that have the same left side as r has.

The prior distribution $H(\boldsymbol{\theta}|\boldsymbol{\mu})$ is:

$$H(\boldsymbol{\theta}|\boldsymbol{\mu}) = \prod_{A \in \boldsymbol{N}} H(\boldsymbol{\theta}_A|\boldsymbol{\mu}_A) \text{ with } H(\boldsymbol{\theta}_A|\boldsymbol{\mu}_A) = \frac{1}{B(\boldsymbol{\mu}_A)} \prod_{r \in \boldsymbol{R_A}} \theta_r^{\mu_r - 1} \tag{1}$$

where we denoting the normalizing constant, $B(\boldsymbol{\mu}_A) = \frac{\prod_{r \in \boldsymbol{R_A}} \Gamma(\mu_r)}{\Gamma(\sum_{r \in \boldsymbol{R_A}} \mu_r)}$. It is easy to see that $B(\boldsymbol{\mu}_A)$ can be re-expressed as

$$B(\boldsymbol{\mu}_A) = \int \prod_{r \in \boldsymbol{R_A}} \theta_r^{\mu_r - 1} d\boldsymbol{\theta}. \tag{2}$$

A $PCFG(G, \boldsymbol{\theta})$ defines a probability distribution over trees generated by itself as follows:

$$P_G(t|\boldsymbol{\theta}) = \prod_{r \in \boldsymbol{R}} \theta_r^{c_r(t)} \tag{3}$$

where $c_r(t)$ is the frequency of the rule $r = A \rightarrow \delta \in \boldsymbol{R}$ in t. If t cannot be generated by G, we set $P_G(t|\boldsymbol{\theta}) = 0$. The yield $y(t)$ of a parse tree t is the sequence of terminals labeling its leaves.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) was introduced in 2003 by Blei and his colleagues as an explicit probabilistic counterpart to Latent Semantic Indexing (LSI). Like LSI, LDA is intended to produce a low-dimensional representation of an article in a collection of articles for information retrieval purposes. LDA is commonly used as a generative probabilistic topic model of a corpus. Its basic idea is that articles in a corpus share the same set of K topics, and each article is represented as a random mixture of latent topics, where each topic is characterized by a distribution over words in the vocabulary. In LDA, the topic proportions for an article is drawn from a Dirichlet distribution. The words in the article are obtained by repeatedly choosing a topic from these proportions, and then drawing a word from the corresponding topic. Blei et al.[10] described a Variational Bayes algorithm for LDA models based on a mean-field approximation; Griffiths and Steyvers[12] described an Markov Chain Monte Carlo algorithm.

3 Bayesian Finite Mixture Model

Let $\boldsymbol{S}$ be a training corpus with total D articles and $\boldsymbol{T}$ be a set of trees, and put

$$\boldsymbol{S} = (s_{11}, s_{12}, \ldots, s_{1n_1}, s_{21}, s_{22}, \ldots), \boldsymbol{T} = (\boldsymbol{t}_{11}, \boldsymbol{t}_{12}, \ldots, \boldsymbol{t}_{1n_1}, \boldsymbol{t}_{21}, \boldsymbol{t}_{22}, \ldots)$$

where $\boldsymbol{t}_{di}$ is a tree of a sentence s_{di}.

Suppose there are K PCFGs. Each of them share the same CFG $G = (\boldsymbol{N}, \sum, ROOT, \boldsymbol{R})$, but possesses K possibly distinct probabilities $\boldsymbol{\theta}$ over R. As mentioned above, each of

these K $\boldsymbol{\theta}$ follows the prior distribution H. Assume the probability distribution $\boldsymbol{\pi}$ over the K grammars, is a Dirichlet distribution with a dirichlet parameter $\boldsymbol{\alpha}$.

Using this new model, we can generate sentences as follows. For each article d in the corpus $\boldsymbol{S}$, first generate a distribution over grammars $\boldsymbol{\pi}_d \sim Dir(\gamma)$, and then for each sentence s_{di} in the article, generate a grammar z_{di} from $\boldsymbol{\pi}_d$, and then the tree and raw text of the sentence are generated from the grammar $\boldsymbol{\theta}_{z_{di}} \sim H$. The following summarizes the steps of generating sentences via the Bayesian finite mixture model.

- For each of the K grammars, draw grammar $\boldsymbol{\theta}_k \sim H(\boldsymbol{\theta}|\boldsymbol{\mu})$
- For each article d in a corpus $\boldsymbol{S}$:
  1. Draw $n_d \sim Poisson(\xi)$, the number of sentences in d.
  2. Draw $\boldsymbol{\pi}_d \sim Dir(\boldsymbol{\alpha})$.
  3. For each of the n_d sentences s_{di}:
     (a) Draw a grammar $z_{di} \sim Multinomial(\boldsymbol{\pi}_d)$.
     (b) Draw a sentence from $p(s_{di}, t_{di}|\boldsymbol{\theta}, z_{di})$.

Several simplifying assumptions are made in this model. First, the total number K of the grammars is assumed known and fixed. Second, the Poisson assumption for the sentence number is not critical and other distributions can be used. Furthermore, n_i is independent of $\boldsymbol{\theta}$ and $\boldsymbol{z}$ and thus can be ignored.

The model looks straightforward while the posterior is intractable to compute and we must appeal to approximate posterior inference. Modern approximate posterior inference algorithms fall into two categories: sampling approaches and optimization approaches. The sampling approaches usually use Markov Chain Monte Carlo (MCMC) sampling, with the objective to simulate draws from the posterior distribution. Optimization approaches are usually based on variational inference. A typical one is the so-called variational Bayes (VB) method in the context of a Bayesian hierarchical model. Variational Bayes methods aim to optimize the Kullback-Leibler divergence of a simplified parametric distribution to the posterior.

Generally speaking, variational Bayes methods are computationally more efficient than MCMC methods, but their performances are usually poorer. However, in our context, we found that the computational complexity of variational Bayes methods is too high to bare due to the embedded Inside-Outside algorithm. To update the parameters upon each sentence in the corpus, we need to calculate the inside table and the outside table in the Inside-Outside algorithm, together with the expected counts for each rule in CFG. In this paper, we thus consider a Markov chain Monte Carlo method which requires only the computation of the inside tables. Our objective is to compare the parsing accuracy(F1 score) of standard CKY parsers using the grammars trained by our multi-grammar model, single-grammar model[8] and standard Inside-Outside algorithm[13].

4 Inference by Markov Chain Monte Carlo (MCMC) Method

In Bayesian inference, the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ used in the Dirichlet priors of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ are assumed to be known (or chosen by the model designer).

Given the corpus $\boldsymbol{S}$, the objective of this paper is to sample $\boldsymbol{Z}$ and $\boldsymbol{\Theta}$ from the joint posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{Z}|\boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu})$ which is not straightforward to obtain. That is because the evaluation of the normalizing constant for this joint posterior distribution

requires summing over all set of rule probabilities and all set of possible grammar assignments. Therefore, we adopt the MCMC approximation method in this paper. During derivation, We notice that $\boldsymbol{\Theta}$ can be marginalized out and updated after the sampling in a straightforward way. Besides, this can save us a lot of computation time.

We start from deriving the joint distribution of $\boldsymbol{S}$, $\boldsymbol{T}$ and $\boldsymbol{Z}$:

$$p(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\mu}) = p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\mu})p(\boldsymbol{Z}|\boldsymbol{\alpha}). \tag{4}$$

which forms the basis of the derivation of the MCMC updating rules and the parameters estimation rules. Since $p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\mu})$ and $p(\boldsymbol{Z}|\boldsymbol{\alpha})$ depend on $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ respectively, we will derive them separately.

According to the definition of the Bayesian mixture model, we have

$$p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\mu}) = \int p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\mu})d\boldsymbol{\Theta}. \tag{5}$$

where $p(\boldsymbol{\Theta}|\boldsymbol{\mu})$ is a product of the prior distributions:

$$p(\boldsymbol{\Theta}|\boldsymbol{\mu}) = \prod_{k=1}^{K} H(\boldsymbol{\theta}_k|\boldsymbol{\mu}). \tag{6}$$

and $p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\Theta})$ is a multinomial distribution:

$$\begin{aligned} p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\Theta}) &= \prod_{d=1}^{D} \prod_{i=1}^{n_d} p(t_{di}|z_{di}, \theta)p(s_{di}|t_{di}) \\ &= \prod_{k=1}^{K} \prod_{A \in N} \prod_{r \in R_A} \theta_{k,r}^{\sum_{d=1}^{D} \sum_{i=1}^{n_d} c_r(t_{di}) \mathbb{1}(y(t_{di})=s_{di}) \mathbb{1}(z_{di}=k)}. \end{aligned} \tag{7}$$

To simplify the representation, we denote $\boldsymbol{\Psi}$ as a $K \times R$ count matrix and $\Psi_{k,r}$ is the number of times that rule r in k-th grammar occurred:

$$\Psi_{k,r} = \sum_{d=1}^{D} \sum_{i=1}^{n_d} c_r(t_{di}) \mathbb{1}(y(t_{di}) = s_{di}) \mathbb{1}(z_{di} = k).$$

and we use $\boldsymbol{\Psi}_k$ to denote the k-th row of the matrix $\boldsymbol{\Psi}$. $\boldsymbol{\Psi}_{k,A}$ is a sub-vector of $\boldsymbol{\Psi}_k$, with elements belonging to the rules in R_A. Given Equ. (6) and Equ. (7), Equ. (5) becomes

$$\begin{aligned} p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\mu}) &= \int \prod_{k=1}^{K} \prod_{A \in \boldsymbol{N}} \frac{1}{B(\boldsymbol{\mu}_A)} \prod_{r \in \boldsymbol{R_A}} \theta_{k,r}^{\Psi_{k,r}+\mu_{k,r}-1} d\boldsymbol{\Theta} \\ &= \prod_{k=1}^{K} \prod_{A \in \boldsymbol{N}} \frac{B(\boldsymbol{\Psi}_{k,A} + \boldsymbol{\mu}_{k,A})}{B(\boldsymbol{\mu}_{k,A})}. \end{aligned} \tag{8}$$

Now we derive $p(\boldsymbol{Z}|\boldsymbol{\alpha})$ analogous to $p(\boldsymbol{S}, \boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{\mu})$. $p(\boldsymbol{Z}|\boldsymbol{\Pi})$ is a multinomial distribution and $p(\boldsymbol{\Pi}|\boldsymbol{\alpha})$ is a product of Dirichlet distributions. For the same reason, we denote $\boldsymbol{\Omega}$ as

a $D \times K$ count matrix and $\Omega_{d,k}$ is the number of times that k-th grammar is assigned to sentences in article d :

$$\Omega_{d,k} = \sum_{i=1}^{n_d} \mathbb{1}(z_{di} = k).$$

we use $\boldsymbol{\Omega}_d$ to denote the d-th row of the matrix $\boldsymbol{\Omega}$. Similar with Equ. (5), we have

$$\begin{aligned} p(\boldsymbol{Z}|\boldsymbol{\alpha}) &= \int p(\boldsymbol{Z}|\boldsymbol{\Pi}) p(\boldsymbol{\Pi}|\boldsymbol{\alpha}) d_{\boldsymbol{\Pi}} \\ &= \prod_{d=1}^{D} \frac{B(\boldsymbol{\Omega}_d + \boldsymbol{\alpha})}{B(\boldsymbol{\alpha})}. \end{aligned} \tag{9}$$

Based on Equ. (8) and Equ. (9), the joint distribution Equ. (4) is:

$$p(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{Z}|\boldsymbol{\alpha}, \boldsymbol{\mu}) = \prod_{k=1}^{K} \prod_{A \in \boldsymbol{N}} \frac{B(\Psi_{k,A} + \mu_{k,A})}{B(\boldsymbol{\mu}_{k,A})} \times \prod_{d=1}^{D} \frac{B(\Omega_d + \boldsymbol{\alpha})}{B(\boldsymbol{\alpha})}. \tag{10}$$

The theory of MCMC shows that MCMC algorithms construct a Markov chain that has the desired distribution as its equilibrium distribution. That is, after a number of iterations the states of the Markov chain can be viewed as a sample draw from the desired distribution. In this paper, the states are $\boldsymbol{T}$: all possible trees of the entire corpus $\boldsymbol{S}$; and $\boldsymbol{Z}$: all possible combinations of grammar assignments for the sentences in the corpus. The transition probabilities are well-designed to guaranteed to converge to our joint posterior distribution. As $\boldsymbol{S}$ is given, we need to sample $\boldsymbol{Z}$ and $\boldsymbol{T}$ alternatively from the two distributions, $p(\boldsymbol{Z}|\boldsymbol{T}, \boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu})$ and $p(\boldsymbol{T}|\boldsymbol{Z}, \boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu})$.

4.1 MCMC Updating Scheme for $\boldsymbol{Z}$

With Equ. (10), we can derive the MCMC updating scheme for $\boldsymbol{Z}$:

$$\begin{aligned} p\left(z_{di} = k|\boldsymbol{Z}^{-di}, \boldsymbol{T}, \boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right) &= \frac{p\left(z_{di} = k, \boldsymbol{Z}^{-di}, \boldsymbol{T}, \boldsymbol{S}|\boldsymbol{\alpha}, \boldsymbol{\mu}\right)}{p\left(\boldsymbol{Z}^{-di}, \boldsymbol{T}, \boldsymbol{S}|\boldsymbol{\alpha}, \boldsymbol{\mu}\right)} \\ &\propto \frac{p\left(\boldsymbol{Z}, \boldsymbol{T}, \boldsymbol{S}|\boldsymbol{\alpha}, \boldsymbol{\mu}\right)}{p\left(\boldsymbol{Z}^{-di}, \boldsymbol{T}^{-di}, \boldsymbol{S}^{-di} \mid \boldsymbol{\alpha}, \boldsymbol{\mu}\right)} \quad \text{where } z_{di} = k. \end{aligned} \tag{11}$$

$$\begin{aligned} p\left(\boldsymbol{Z}^{-di}, \boldsymbol{T}^{-di}, \boldsymbol{S}^{-di} \mid \boldsymbol{\alpha}, \boldsymbol{\mu}\right) &= p\left(\boldsymbol{S}^{-di}, \boldsymbol{T}^{-di}|\boldsymbol{Z}^{-di}, \boldsymbol{\mu}\right) p\left(\boldsymbol{Z}^{-di} \mid \boldsymbol{\alpha}\right) \\ &= \prod_{\bar{k}=1}^{K} \prod_{A \in \boldsymbol{N}} \frac{B(\Psi_{\bar{k},A}^{-di} + \mu_{\bar{k},A})}{B(\boldsymbol{\mu}_{\bar{k},A})} \times \prod_{d=1}^{D} \frac{B(\Omega_d^{-di} + \boldsymbol{\alpha})}{B(\boldsymbol{\alpha})} \end{aligned} \tag{12}$$

where $\Psi_{k,r}^{-di}$ is the number of times that rule r in the k-th grammar occurred, but with the i-th sentence of d-th article and its grammar assignment excluded. Similarly, $\Omega_{d,k}^{-di}$ is the

number of sentences in article d that are assigned grammar k, but with the i-th sentence of d-th article and its grammar assignment excluded. Some properties can be derived as

$$\Psi_{k,r} = \Psi_{k,r}^{-di} \text{ if } z_{di} \neq k \tag{13}$$

$$\Omega_{d,k} = \begin{cases} \Omega_{d,k}^{-di} + 1 & \text{if } z_{di} = k \\ \Omega_{d,k}^{-di} & \text{otherwise} \end{cases} \tag{14}$$

$$\sum_{k=1}^{K} \Omega_{d,k} = \sum_{k=1}^{K} \Omega_{d,k}^{-di} + 1 \tag{15}$$

Combined together:

$$p\left(z_{di} = k | \boldsymbol{Z}^{-di}, \boldsymbol{T}, \boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right) \propto \frac{\prod_{A \in \boldsymbol{N}} B(\Psi_{k,A} + \mu_{k,A})}{\prod_{A \in \boldsymbol{N}} B(\Psi_{k,A}^{-di} + \mu_{k,A})} \times (\Gamma(\Omega_{d,k} + \alpha_k) - 1) \tag{16}$$

4.2 MCMC Updating Scheme for $\boldsymbol{T}$

We will use the following formula to update $\boldsymbol{T}$.

$$p(t_{di} | \boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{p\left(s_{di} | t_{di}\right) p\left(t_{di} | \boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)}{p\left(s_{di} | \boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)} \tag{17}$$

As described above, integrating out $\boldsymbol{\Theta}$, we will obtain,

$$\begin{aligned} p\left(\boldsymbol{T} | \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right) &= \prod_{d=1}^{D} \prod_{i=1}^{n_d} \int p(t_{di} | \boldsymbol{\Theta}, z_{di}) p(\boldsymbol{\Theta} | \boldsymbol{\mu}) d\boldsymbol{\Theta} \\ &= \prod_{k=1}^{K} \prod_{A \in \boldsymbol{N}} \frac{B(\hat{\Psi}_{k,A} + \mu_{k,A})}{B(\boldsymbol{\mu}_{k,A})} \end{aligned} \tag{18}$$

where $\hat{\boldsymbol{\Psi}}$ is a $K \times R$ count matrix and the (k, r) cell $\hat{\Psi}_{k,r}$ is the number of times that rule r in k-th grammar occurred within T :

$$\hat{\Psi}_{k,r} = \sum_{d=1}^{D} \sum_{i=1}^{D_i} c_r(t_{di}) \mathbb{1}(z_{di} = k),$$

and $\hat{\boldsymbol{\Psi}}_{k,A}$ is a sub-vector of the k-th row of the matrix $\hat{\boldsymbol{\Psi}}$, with elements belonging to the rules in R_A. With Equ. (18), we can get the formula for one of the probabilities in Equ. (17).

$$\begin{aligned} p\left(t_{di} | \boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right) &= \frac{p\left(\boldsymbol{T} | \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)}{p\left(\boldsymbol{T}^{-di} | \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)} \\ &= \prod_{A \in \boldsymbol{N}} \frac{B(\hat{\boldsymbol{\Psi}}_{k,A} + \boldsymbol{\mu}_{k,A})}{B(\hat{\boldsymbol{\Psi}}_{k,A}^{-di} + \boldsymbol{\mu}_{k,A})} \end{aligned} \tag{19}$$

where $k = z_{di}$ and the tree for sentence i in article d is a newly sampled tree t_{di}. We notice that the probability $p\left(s_{di}|\boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)$ in Equ. (17) is hard to calculate. Therefore, here we use Metropolis-Hastings algorithm as suggested by Johnson et al.[8].

The Metropolis-Hastings (MH) algorithm is a Markov Chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult. The algorithm simulates samples from a probability distribution by making use of a proposal distribution, which is easy to sample. Let $Q(x^t|x^{t-1})$ be a proposal distribution that is used to sample from for the desired probability distribution $P(x)$. For each iteration, a candidate $\hat{x}$ is sampled from $Q(x^t|x^{t-1})$ and it will be accepted with probability

$$p(x^t = \hat{x}) = min\left\{1, \frac{P(\hat{x})Q(x^{t-1}|\hat{x})}{P(x^{t-1})Q(\hat{x}|x^{t-1})}\right\} \tag{20}$$

and with probability $1 - p(x^t = \hat{x})$ the candidate is rejected, then we set $x^t = x^{t-1}$. We choose $p(t_{di}|s_{di}, z_{di}, \hat{\Theta})$ as the proposal distribution for Equ. (17), where $\hat{\Theta} = (\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \ldots, \hat{\boldsymbol{\theta}}_K)$ is the expected value $E[\Theta|\boldsymbol{T}^{-di}, \boldsymbol{Z}^{-di}, \mu]$ as follows:

$$\hat{\theta}_{k,r} = \frac{\Psi_{k,r}^{-di} + \mu_{k,r}}{\sum_{r \in R_A} \Psi_{k,r}^{-di} + \mu_{k,r}} \tag{21}$$

For sentence i in article d, the current tree is t_{di}. We first calculate the expected value of $\hat{\boldsymbol{\theta}}_{z_{di}}$ based on the other trees $\boldsymbol{T}^{-di}$. Then we sample a new tree $\hat{t}_{di}$ from $p(t_{di}|s_{di}, \hat{\boldsymbol{\theta}}_{z_{di}})$ using the algorithm described in next section. Finally, we choose the next tree for s_{di} as

$$t_{di}^{new} = \begin{cases} \hat{t}_{di} & \text{with probaility } q = \min\left\{1, \frac{p(\hat{t}_{di}|\boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}) p(t_{di}|\hat{\boldsymbol{\theta}}_{z_{di}})}{p(t_{di}|\boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}) p(\hat{t}_{di}|\hat{\boldsymbol{\theta}}_{z_{di}})}\right\} \\ t_{di} & \text{with probaility } 1 - q \end{cases} \tag{22}$$

Probabilities like $p(s|\boldsymbol{\theta})$ in $p(t|s, \boldsymbol{\theta}) = \frac{p(t|\boldsymbol{\theta})}{p(s|\boldsymbol{\theta})}$, and $p\left(s_{di}|\boldsymbol{T}^{-di}, \boldsymbol{Z}, \boldsymbol{\alpha}, \boldsymbol{\mu}\right)$ in Equ. (17) are common factors of both the numerator and denominator in the formula of q, and hence will be eliminated. The item $p(s_{di}|t_{di})$ in Equ. (17) will always be 1 since the proposed distribution can only generate trees that s_{di} is the yield.

4.3 Sampling from the Proposal Distribution $p(t|s, \theta)$

To sample trees from $p(t|s, \boldsymbol{\theta})$, we utilize an efficient sampling algorithm described in previous works[6][8][14][15]. There are two parts in this algorithm. The first part constructs a standard inside table as in the Inside-Outside algorithm for PCFG[13]. The second part selects the tree by recursively sampling trees from top to bottom.

Let $s = (w_1, w_2, \ldots, w_l)$ and $w_i^j = (w_{i+1}, \ldots, w_j)$. We define the inside probability $\mathcal{I}$ as follows. Given initial values for $\boldsymbol{\theta}$, for all $A \in N$, for all (i, j) such that $0 \le i < j \le l$, we calculate the following quantities in a dynamic fashion.

$$\begin{aligned} \mathcal{I}(A, i-1, i) &= \theta(A \to w_i) \\ \mathcal{I}(A, i, j) &= \sum_{A \to B\ C \in R} \sum_{m=i+1}^{j-1} (\theta(A \to B\ C) \times \mathcal{I}(B, i, m) \times \mathcal{I}(C, m, j)) \end{aligned} \tag{23}$$

The resulting inside probabilities are then used to sample trees for s. The tree is generated recursively from larger to smaller spans in a top-down fashion. For each span w_i^j and each nonterminal A, the sampling algorithm randomly chooses one of the rules $r \in R_A$ and one of the mid-position $i < m < k$ from a multinomial distribution:

$$p(m, r(A \to B\ C)) = \frac{\theta(A \to B\ C) \times \mathcal{I}(B, i, m) \times \mathcal{I}(C, m, j)}{\mathcal{I}(A, i, j)}. \tag{24}$$

4.4 Summarize the Algorithm

The whole algorithm can be summarized as:

Table 1. A Markov chain Monte Carlo algorithm for inferencing PCFGs

(1)	initialize $K, \boldsymbol{\mu}, \boldsymbol{\alpha}$
(2)	initialize trees based on initial values of $\boldsymbol{\mu}$
(3)	initialize $\boldsymbol{Z}$ randomly
(3)	**repeat**
(4)	update $\boldsymbol{Z}$ by Equ. (16) for fixed times
(5)	for each sentence,
(6)	update $\boldsymbol{\Theta}$ by Equ. (21)
(7)	sample a tree by Equ. (24)
(8)	determine the next tree by Equ. (22)
(9)	**until** convergence

The objects $\boldsymbol{Z}$ and $\boldsymbol{\Theta}$ are collected during the repeat.

5 Experimental Results

We conducted experiments to compare our multi-grammar($K > 1$) model with both the Inside-Outside(IO) algorithm and the single grammar($K = 1$) model. The training corpus used in our experiments was Wall Street Journal corpus of the U. Penn Treebank[16]. In the corpus, articles with no less than 30 sentences and each sentence having 5 to 50 words were collected from WSJ06-09, resulting in a collection of 62 articles with 1950 sentences and gold trees. The initial CFG was extracted from these gold trees and then converted to Chomsky normal form. The converted CFG has 19071 rules, 4252 non-terminals and 7945 terminals. The training data has no labels or brackets.

Three sets of experiments were implemented with initial values for $\boldsymbol{\mu}$ setting equally. First, we ran the standard Inside-Outside algorithm to produce a MLE of $\hat{\boldsymbol{\theta}}_{MLE}$, and then made use of CKY algorithm to get the Viterbi parses $\boldsymbol{T}$ for the sentences in the corpus under PCFG $(G, \hat{\boldsymbol{\theta}})$. Second, we set $K = 1$, and ran our MCMC algorithm, which is nearly equivalent to the Hastings sampler used in Johnson et al.[8]. The estimate $\hat{\boldsymbol{\theta}}_{K=1}$ were again used to parse the corpus. Finally, we considered the cases of $K > 1$. For each value of K, the MCMC algorithm was used to estimate the K rule probabilities

$\hat{\boldsymbol{\theta}}_{k\in 0:K-1}$ and the grammar assignments $\boldsymbol{Z}$. After this, the individual sentence was parsed by CKY algorithm using the corresponding grammar $\hat{\boldsymbol{\theta}}_k$. In all these experiments, the resulting parses were evaluated by comparing against the corresponding gold trees of the same corpus. We set μ to be 0.01 for all rules in $\boldsymbol{R}$ as suggested in [8].

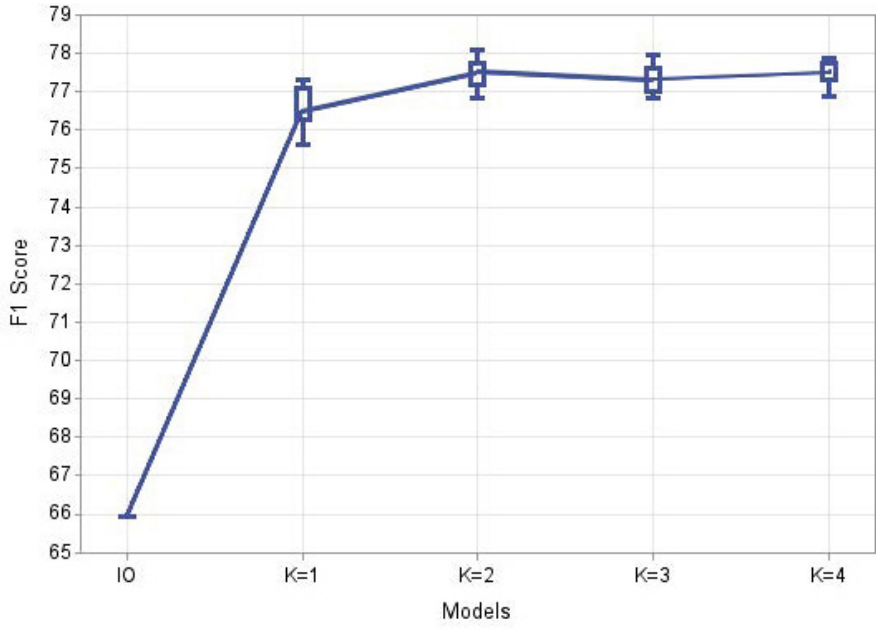


Fig. 1. Box Plot of the experimental results for different models based on a simulation of 11 trails

The experimental results are summarized in Figure 1. IO means Inside-Outside algorithm and $K = 1$ represents the single grammar model. The average F1 score of Inside-Outside algorithm is about 65.95%. All the other models outperform this baseline. The average F1 score for single grammar algorithm is 76.572%, consistent with the results reported by Johnson et al.[8]. Limited by the used corpus size, we only considered three different multi-grammar $K = 2, 3$ and 4, with corresponding average F1 scores 77.477%, 77.387% and 77.479%. Hence all the three groups of multi-grammar model have improved upon the single grammar model. Besides F1 score, we also observed consistent improvements measured by LP, LR and exact accuracy from single grammar model.

We also considered setting the elements of the prior $\boldsymbol{\mu}$ to be distinct values, for example, random numbers sampled from gamma distribution. In our experiments, we noticed that under specified gamma distribution, the F1 score of our model can be as large as 84%, quite close to the performance of pure supervised MLE PCFG.

After the computation of case $K = 2$, we analyzed the resulting 2 PCFGs. The top 10 rules in R_{NP} from these 2 PCFGs were collected and sorted by probabilities in a descending order, which were presented in Table 2. The results show that the distribution for R_{NP} in grammar 1 is quite different from that in grammar 2. Some of the top rules

appear in both grammars, but with different ranks and probabilities. For example, "NP → NP PP-LOC" ranks the second in PCFG 1, while it falls to the 6th place in PCFG 2. This also applies to R_{VP}, which were reported in Table 3.

Table 2. Top 10 rules for NP

PCFG 1		PCFG 2	
NP → DT NN	(0.164)	NP → DT NN	(0.121)
NP → NP PP-LOC	(0.051)	NP → DT JJ NN	(0.052)
NP → DT NX	(0.047)	NP → JJ NNS	(0.044)
NP → DT JJ NN	(0.043)	NP → NP NN	(0.044)
NP → NP NN	(0.043)	NP → NP PP-TMP	(0.042)
NP → NNP NNP	(0.040)	NP → NP PP-LOC	(0.037)
NP → JJ NNS	(0.035)	NP → NP VP	(0.031)
NP → DT NNS	(0.031)	NP → NNP NNP	(0.029)
NP → PRP$ NN	(0.030)	NP → NN NNS	(0.027)
NP → NP SBAR	(0.021)	NP → JJ NN	(0.023)

Table 3. Top 10 rules for VP

PCFG 1		PCFG 2	
VP → TO VP	(0.101)	VP → TO VP	(0.106)
VP → MD VP	(0.062)	VP → MD VP	(0.065)
VP → VB NP	(0.037)	VP → VB NP	(0.051)
VP → VBZ VP	(0.036)	VP → VBP VP	(0.037)
VP → VBP VP	(0.035)	VP → VBZ VP	(0.036)
VP → VBZ NP-PRD	(0.033)	VP → VBG NP	(0.025)
VP → VB NP-PRD	(0.028)	VP → ADVP VP	(0.025)
VP → ADVP VP	(0.027)	VP → VP CC VP	(0.024)
VP → VB SBAR	(0.022)	VP → VBD VP	(0.022)
VP → VP CC VP	(0.020)	VP → VB PP	(0.020)

Furthermore, we also analyzed the vocabulary used by these two PCFGs. After removing a list of stop words and trivial words, the top 20 words occurred in these two PCFGs were presented in Table 4. These two lists of words seem to capture some of the underlying genres in the corpus. The list from PCFG 1 focuses on financial words while the list from PCFG 2 concentrates on political words, which may represent two different categories of contents. The above analysis demonstrate the efficiency of our multi-grammar model. Given a initial grammar, the inference algorithm learns the similarity among the sampled trees of the sentences in the corpus. Through effectively updating, these similar sentences (and their sampled trees) gather together and form their own distinctive distributions over rules and words.

Table 4. Top 20 words for the two PCFGs

PCFG 1	market, stock, U.S., companies, company, markets, prices, government, funds, futures bonds, junk, financial, rate, fund, system, business, share, exchange, contract
PCFG 2	Bush, U.S., people, House, President, Congress, White, government, market, drug budget, country, president, Congress, Engelken, power, world, Senate, program, law

6 Conclusion

We have considered a Bayesian finite mixture model for PCFGs. This model is inferred by the proposed Markov chain Monte Carlo Method. Our experimental results demonstrate that the proposed multi-grammar model outperforms single grammar model. In fact, the articles in corpus WSJ come from related fields and have similar contents and vocabulary. However, in practice, the raw texts may be generated from many different domains. The contents, the vocabulary and the common written practice may be quite diverse which make a single-grammar inadequate. Therefore, we believe our model can function even better in practice than single grammar model.

References

1. Kehler, A., Stolcke, A.: Preface. In: Kehler, A., Stolcke, A. (eds.) Proceedings of the Workshop Unsupervised Learning in Natural Language Processing. Association for Computational Linguistics (1999)
2. Goldwater, S., Griffiths, T.L.: A fully Bayesian approach to unsupervised part-of-speech tagging. In: Proc. of ACL (2007)
3. Toutanova, K., Johnson, M.: A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In: Proc. of NIPS (2007)
4. Eisner, J.: Transformational priors over grammars. In: Proc. of EMNLP (2002)
5. Liang, P., Petrov, S., Jordan, M., Klein, D.: The infinite PCFG using hierarchical Dirichlet processes. In: Proc. of EMNLP (2007)
6. Finkel, J.R., Manning, C.D., Ng, A.Y.: Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 618–626. Association for Computational Linguistics (2006)
7. Kenichi, K., Sato, T.: An application of the variational Bayesian approach to probabilistic context-free grammars. In: International Joint Conference on Natural Language Processing Workshop Beyond Shallow Analyses (2004)
8. Johnson, M., Griffiths, T.L., Goldwater, S.: Bayesian inference for PCFGs via Markov chain Monte Carlo. In: Proc. of NAACL (2007)
9. Iwata, T., Mochihashi, D., Sawada, H.: Learning common grammar from multilingual corpus. In: Proceedings of the ACL 2010 Conference Short Papers. Association for Computational Linguistics (2010)
10. Blei, D.M., Andrew Y.N., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
11. Johnson, M.: PCFGs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2010)
12. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences, 101:52285235 (2004)
13. Lary, K., Young, S.J.: The estimation of stochastic context-free grammars using the inside-outside algrithm. Computer, Speech and Language, 4:3556 (1990)
14. Goodman, J.T.: Parsing Inside-Out. Ph.D. thesis, Harvard University Cambridge, Massachusetts (1998)
15. Sun, L., Mielens, J., Baldridge, J.: Parsing low-resource languages using Gibbs sampling for PCFGs with latent annotations. To appear in Proceedings of EMNLP 2014 (2014)
16. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)

Employing Oracle Confusion for Parse Quality Estimation

Sambhav Jain, Naman Jain, Bhasha Agrawal, and Rajeev Sangal

International Institute of Information Technology, Gachibowli,
Hyderabad 500032, Telangana, India
{sambhav.jain,naman.jain,bhasha.agrawal}@research.iiit.ac.in,
sangal@iiit.ac.in
http://www.iiit.ac.in

Abstract. We propose an approach for *Parse Quality Estimation* based on the dynamic computation of an entropy-based confusion score for directed arcs and for joint prediction of directed arcs and their dependency labels, in a typed dependency parsing framework. This score accompanies a parsed output and aims to present an exhaustive picture of the *parse quality*, detailed down to each arc of the parse tree. The methodology explores the confusion encountered by the oracle of a transition-based data-driven dependency parser. We support our hypothesis by analytically illustrating, for 18 languages, that the arcs with high confusion scores are notably the predominant parsing errors.

Keywords: Dependency Parsing, Parse Quality Estimation, Confusion Score.

1 Introduction

A major goal of syntactic parsing research is to develop quality parsers, which can provide reliable syntactic analysis to various NLP applications, such as statistical machine transition [32], natural language generation [31], text summarization evaluation [23] etc. In spite of extensive advancements in parsing research, it is observed that even state of the art parsers, with high accuracies, often fail to meet quality expectations of an application. The reason behind this expectation-gap is manifold.

In context of the utility of a parser by an application, one expects a certain parsing accuracy for a given input sentence. Customary evaluation matrices for parsing compute accuracies averaged over total nodes of a test set (sometimes an n-fold cross validation). The assumption that this average accuracy approximates the accuracy of a sentence, is inaccurate, since the errors are not equally distributed over the sentences. For instance, parsers are known to perform poorly for longer sentences in comparison to short ones [16]. Users are often lured by the high averaged accuracies and expect them to hold for all kind of sentences (short or long).

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 213–226, 2015.
DOI: 10.1007/978-3-319-18111-0_17

Domain confinement of parsers is another well known concern. Statistical parsers are trained on data compiled from finite domains, but NLP applications are often domain unbound and do freely accept data from any common domain. Thus, the reported performance is an overestimate of actual parsing performance of these parsers and is destined to degrade, when employed practically in NLP applications [10].

Much of the efforts in the past have been on improving parser performance or accustoming parsers to new domains. Now, it is the time to think on: *"how to reliably deliver a reliable parse (or parts of a parse)"*. If we have a mechanism of identifying accurate parses, they can be utilized in an application without hesitation. Similarly for an incorrect parse, there may exist fragments which are correctly parsed. Again, if we have a mechanism to identify these correct fragments, they too could be selectively utilized. Thus, in order to give a more informed picture about *per-edge confusion (or confidence)* of a parsed tree, we propose an approach for Parse Quality Estimation (henceforth $\mathcal{PQE}$) by dynamically computing an *entropy-based* confusion score for predicted dependency arcs, based on the methods proposed by [8]. To state briefly, we present the following contributions of this paper:

- Integrating our approach with the current functionality of MaltParser [21], a popular transition-based data-driven parser to accredit the parse quality corresponding to each arc in the output dependency structure.
- $\mathcal{PQE}$ based automatic error detection for 18 languages.

The rest of the paper is organized as follows: In the next Section we will discuss about the related research in the area of quality estimation. In Section 3 we will discuss the methodology to be adopted to calculate $\mathcal{PQE}$. In Section 4 we will discuss our efforts to calculate the oracle confusion for $\mathcal{PQE}$ and in the Section 5 we will discuss how $\mathcal{PQE}$ captures the Oracle confusion. In Section 6 we will discuss the error detection task using parser's output accredited with $\mathcal{PQE}$ score. Finally we conclude with some future direction in Section 7.

2 Related Research

The need for upfront estimation of parse quality has been acknowledged in various works, yet there exist very few efforts in the direction of explicitly addressing $\mathcal{PQE}$.

Parse Re-ranking Scores. $\mathcal{PQE}$ bears a resemblance with the parse re-ranking problem where the k-best probable parses are compared on lexicalized and syntax based linguistic features such as POS bi-grams, lexical bi-grams, head modifiers [3,11,2]. However, the objective here is to scrutinize and weight distinct parses of the same sentence. Thus the designed scores, unlike $\mathcal{PQE}$, are not

directly commensurable to compare the quality of parse from two distinct sentences[1].

Uncertainty Measure in Active Learning. Another area with coinciding concern is Active Learning [26] where new sentences are ranked based on the highest uncertainty, an existing parser exerts during parsing [30,7]. [7] in their work on sample selection for statistical constituency parsing, proposed an uncertainty score, measured in terms of entropy of a parse tree, that a hypothesis grammar generate for a given sentence.

Supervised Reliability Prediction. [25] worked on predicting the parser accuracy for constituency parsing by training a SVM regression model over text based features such as sentence length, unknown words etc. [9] used similar features to train a binary SVM classification model for judging reliable dependency parse while addressing parser domain adaptation. However, such a supervised approach is highly sensitive to the selection of the training data, which again in turn requires adequate exploration.

Per Edge Correctness Estimation. [4] in their attempt on precision-biased parsing, defined a riskiness function over dependency arc, which is calculated based on ensembling agreement between two parsers. [17] proposed four methods to estimate the confidence in predicting the dependency tree edges. His approach renders a confidence score for each parse edge produced by MST Parser [15]. [8] proposed an approach to compute a confusion score for dependency arc-label predicted by MaltParser [21].

3 Methodology

The suggestion of $\mathcal{PQE}$ looks promising and alluring, but to be put into practice, it requires a computational approach to materialize the idea.

We found three works, whose approaches stand out from others and are an exception to the related research in this direction. Firstly, Hwa's [7] entropy based uncertainty measure, which captures the uncertainty in generation of a tree by a probabilistic grammar. Secondly, [17], targets confidence calculation for each edge in a parse tree. Lastly, [8], targets confusion score calculation for each arc-label in a parser tree by calculating entropy with class membership probabilities of the parser actions in MaltParser.

Taking insights from these works, we suggest $\mathcal{PQE}$, computed for each edge of a parse tree, based on the prediction-uncertainty in the model of a parser.

[1] We conducted an experiment on English to confirm the proposition. Parse generation probabilities and re-ranking scores, from Collins parser and Charnaik & Jhonson's parser, were individually plotted against parse accuracy (of the best parse) for a sentence. We did not find any correlation between individual parse's correctness and scores generated by the parsers.

To actualize our efforts, we chose MaltParser [21], a popular transition-based parser, to work with.

A dependency structure explicitly represents head-dependent relations (directed arcs) and functional categories (arc labels). MST parser [15], a graph based parser, has already been explored by [17] in their work for predicting confidence for dependency tree arcs. MaltParser [21] has also been explored by [8] to capture confusion encountered by the oracle in predicting arc labels but it lacks to capture the complete picture of the parsed output as it does not capture the confusion encountered by the oracle while predicting arcs in a dependency parsed structure. Given the fact that each paradigm (transition or graph based) has its own strengths and weaknesses [33], motivated us to explore a transition-based parser for employing oracle confusion during either prediction of arc-formation or joint prediction of arc-formations and arc-labels for $\mathcal{PQE}$, which has not been explored to the best of our knowledge.

4 Oracle Confusion for $\mathcal{PQE}$

MaltParser outputs a single best parse by greedily choosing parsing actions advocated by an oracle trained on the training data. In a typed dependency framework, the parser performs two distinct kinds of actions to form a dependency tree: formation of directed arcs (or edges) between two words (or vertices) (henceforth *attachment*) and assignment of arc labels (or dependency labels) (henceforth *label*) to previously formed directed arcs. MaltParser provides a choice[2] to train separate oracles for the above two kinds of actions or a single oracle that jointly predicts both attachment and label for a pair of words.

In case of separate oracles for attachment and label prediction, the parser first starts with querying the attachment oracle for an appropriate parser action. In Nivre's algorithm, there are four possible parsing actions namely, *Shift* (S), *Reduce* (R), *Left Arc* (LA) and *Right Arc* (RA). Here LA and RA are arc-forming[3] parser actions, while S and R are non-arc-forming[4] parsing actions. The attachment oracle is queried until an arc-forming action is returned, which signifies an arc formation. Next, the label oracle is queried for an arc label, as per the given context, which is then associated with the recently formed arc. In joint oracle, the arc-forming actions are concatenated with possible labels. Now, if an arc-forming parser action is returned it also has a label concatenated with it (for eg. $LA \sim nsubj$).

[2] `http://www.maltparser.org/userguide.html#predstrate`

[3] Parsing action results in arc formation either in right to left or left to right direction in separate oracle. In case of a joint oracle, in addition to arc formation also delivers the label corresponding to the resultant arc

[4] Parsing action does not result in forming an arc neither does it predicts a label, instead governs a state change in the arrangement of tokens or switches in data structures like stack of partially processed tokens and queue of remaining input tokens.

4.1 Calculating Entropy for Parser Actions

The confusion score of each attachment in a parse tree can be derived from the entropies of the parser actions that resulted in arc formation. A parser action can results either in arc-forming or non-arc-forming category and accordingly corresponding entropy is denoted as H_{arc} or $H_{non-arc}$. Based on the lines of Jain and Agrawal [8], uncertainty or confusion score of each attachment is quantitatively determined by *entropy(H)* using the following formula:

$$H_{PA} = -\sum_{i=1}^{n} p_i \log(p_i) \tag{1}$$

where n denotes the total number of possible candidates for parser action (henceforth *PA*) and p_i denotes the membership posterior probability corresponding to i^{th} candidate. The higher the entropy, the more uncertain the oracle is about the prediction.

4.2 Confusion Score for Tree Edges

Transforming uncertainty into confusion score for tree edges is not straightforward. The restricting factor being the presence of non-arc-forming parser actions i.e. S and R. Since these transitions are not decomposed over the tree edges, the oracle confusion associated with them can not be delineated to any specific edge. For example, at a given state during parsing, oracle will need to decide if it should perform a LA, RA, R or S action. This decision will not only influence the single edge immediately added by the LA or RA action, but also influence other future edges. It should also be noticed that, though S or R action will not add any edge, yet will have a complex effect on the set of edges that could or could not be added in future. Thus, the entropy of parser actions, resulting in non-arc-forming decisions should be considered together with the entropy of an arc-forming action while computing the uncertainty of an arc (edge) formation. However, it must be noted that label prediction in case of separate oracles does not have a dependency on any of the previous parser actions, and thus entropy of the labeling parser action is directly attributed as the confusion score for that label.

$$Confusion\ Score_{Attachment_i} = f(H_{arc_i}, H_{non-arc_{i-1}}, H_{non-arc_{i-2}}....) \tag{2}$$

For confusion score of attachments, we resort to following approaches to derive a function for combining the non-arc decisions with the arc-decisions :

Assuming Independence from Vicinity. Before moving to complex mechanisms of combining H_{arc} and $H_{non-arc}$ entropies, it is worth to consider the approximation that the confusion of an i^{th} attachment action only depends on H_{arc_i}.

$$Confusion\ Score_{Attachment_i} = H_{arc_i} \tag{3}$$

Decay Factor. Adhering to the fact that transition-based parser makes local and greedy decisions, we applied *principal of locality* and propose that parser actions in immediate vicinity have larger contribution in the confusion score as to a bit previous ones. Figure 1 describes the derivation of the algorithm to compute the confusion score of i^{th} attachment action in totality which takes a fraction of the remaining uncertainty at every step.

Total Confusion Score (uncertainty) due to all *PA*s can be **1.0**

The contribution from the *arc* forming *PA* $(arc_i) = \beta_0 \times H_{arc_i}$

where $\beta_0 = 1$

Remaining Uncertainty $= 1 - H_{arc_i}$

The contribution from previous *non-arc* forming *PA* $((i-1)^{\text{th}}$ *PA*) (if present)

$= (1 - H_{arc_i}) \times \beta_1 \times H_{non-arc_{i-1}}$

where β_1 is an additional decay parameter.

Similarly, contribution from 2^{nd} previous *non-arc* forming *PA* or *(i-2)*th *PA*

$= \text{New Remaining Uncertainty} \times \beta_2 \times H_{non-arc_{i-2}}$

$= (((1 - H_{arc_i}) - (1 - H_{arc_i}) \times \beta_1 \times H_{non-arc_{i-1}}) \times \beta_2 \times H_{non-arc_{i-2}}$

$= (1 - H_{arc_i})(1 - \beta_1 . H_{non-arc_{i-1}}) \times \beta_2 \times H_{non-arc_{i-2}}$

from j^{th} previous *non-arc* forming *PA* or *(i-j)*th PA

$\text{Contribution}_j = (1 - H_{arc_i}).(1 - \beta_1 . H_{non-arc_{i-1}}) \ldots (1 - \beta_{j-1} . H_{non-arc_{i-(j-1)}}) \times \beta_j \times H_{non-arc_{i-j}}$

$$\textbf{Contribution}_j = (1 - H_{arc_i}) \times \left(\prod_{j=1}^{j-1} (1 - \beta_j H_{non-arc_{i-j}}) \right) \times \beta_j \times H_{non-arc_{i-j}}, \text{ for } j \geq 1$$

Consolidating the contribution as,

$$\textbf{Confusion Score}_{Attachment_i} = \sum_{j=0}^{k} \text{Contribution}_j$$

where,

$\text{Contribution}_0 = (1 - H_{arc_i})$,

k is total number of previous *non-arc* forming *PA*s corresponding to i^{th} *arc* forming *PA*

Fig. 1. Algorithm to Compute the Confusion Score in totality

Now, confusion score corresponding to an arc, depends on the entropy of a parser action contributing to the arc-formation and previous actions, but multiplied by a factor known as Decay factor β_i corresponding to i^{th} previous parser action. Since β_i is unknown in the calculation of confusion score, we establish different functional behaviors to calculate it:

1. **Linear Decay.** The parameter adhere to a linear decay and subsequently can be calculated as:

$$\begin{aligned} \beta_x &= -tx + c \\ &= 1 - tx \quad (\ c = 1 \ \because \ \beta_0 = 1); \end{aligned} \tag{4}$$

2. **Polynomial Decay.** The parameter follows a polynomial decay as :

$$\begin{aligned} \beta_x &= -\alpha x^t + c \\ &= 1 - \alpha x^t \quad (\ c = 1 \ \because \ \beta_0 = 1) \end{aligned} \tag{5}$$

3. **Exponential Decay.** The parameter follows a exponential decay as :

$$\begin{aligned} \beta_x &= \alpha^{mx} + c \\ &= \alpha^{mx} \quad (\ c = 0 \ \because \ \beta_0 = 1) \\ let\ \alpha^m &= e^t, \\ \beta_x &= e^{tx} \end{aligned} \tag{6}$$

where, $0 \leq t$, $\alpha \leq 1$, to accommodate decaying characteristics and $x = \{0, 1, 2, ..., k\}$ denotes the previous non-arc-forming *PA*s. The parameters α and t are tuned on development set. Confusion scores are calculated for each possible values of parameters iteratively and in each iteration, edges are sorted in decreasing order of their confusion scores followed by book-keeping the number of incorrectly parsed edges in top *'K'* entries. *'K'* is total number of incorrectly parsed edges in the development set. The corresponding values which prioritize maximum errors are chosen as parameters. As illustrated in Figure 2, α is chosen to be *0.72* as it prioritize maximum errors (156/161).

α=.01	α=.02	... ... α=0.72	α=0.98	α=0.99
e#134* (0.998)	e#045* (0.984)	... ...	e#091* (0.983)	e#063 (0.991)
.	.	. . . . .	.	.
.	.	. . . . .	.	.
e#111 (0.785)	e#063 (0.712)	... ...	e#134* (0.804)	e#045* (0.821)
#Error= 42/161	46/161	. . 156/161 . .	32/161	32/161

Fig. 2. Choosing parameter α based on the ability to prioritize errors. 'e#N' (Confusion Score) : Edge Index 'N' with Confusion Score in brackets; * denotes an actual incorrect edge. '#Error' : Total number of incorrect edges correctly prioritized out of total incorrect edges actually present

4.3 Complex Association: Regression Analysis

We work with an intuitive assumption of a diminishing contribution from previous *PA*. However, the actual relation may be more complex. So, we do away with the assumption of decaying contribution of previous *PA*s.

In order to establish a function that best fits the scenario to furnish confusion scores, we utilize regression analysis. We used a development set to train a *SVM* regression model with entropy of an arc-forming action and all prior actions' entropies as features. If an arc has been parsed incorrectly we keep the

corresponding confusion score value as 1.0 otherwise it is kept 0.0 in the data for training the regression model.

The above five configurations give us five distinct systems to compute *confusion scores* for attachments. We denote them respectively as $\mathcal{S}_{independent}$, $\mathcal{S}_{linear}$, $\mathcal{S}_{polynomial}$, $\mathcal{S}_{exponential}$ and $\mathcal{S}_{regression}$. All the systems give confusion scores between 0.0 and 1.0. $\mathcal{S}_{regression}$ utilizes a curve fitting approach, therefore by default may give values slightly less than 0.0 or marginally greater than 1.0. We incorporated an intermediate normalization step for adjusting the values between 0.0 to 1.0.

5 $\mathcal{PQE}$ Score Capturing Oracle Confusion

The confusion scores computed from aforementioned approaches are indicators of parse quality. To estimate the quality of an arc (or confidence score) on the scale of 0 to 1, the confusion score can be subtracted from 1.

$$\mathcal{PQE}_{system} = 1.0 - \mathcal{S}_{system} \tag{7}$$

5.1 Correlation between $\mathcal{PQE}$ Scores and Actual Accuracy

To validate the authenticity of our proposed $\mathcal{PQE}$ scores, we plot them against the actual accuracy of the arcs which depict a positive correlation between the quantities as shown in Figure 3. Best performance will be obtained when a line corresponding to a method is close to the line $y = x$ in Figure 3.

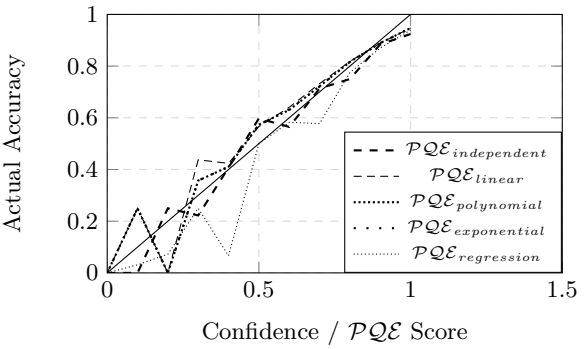


Fig. 3. Predicted vs Actual Accuracy comparison between different PQE systems for English attachments

5.2 Comparison with Mejer and Crammer (2012) [17]

We compared our approach with [17], for English attachments. MSTParser along with confidence score is trained for the CoNLL 2007 English data, so that a fair comparison can be performed. We found that $\mathcal{PQE}_{exponential}$ closely match [17] for scores ≥ 0.3, while for the rest of the range a bit better.

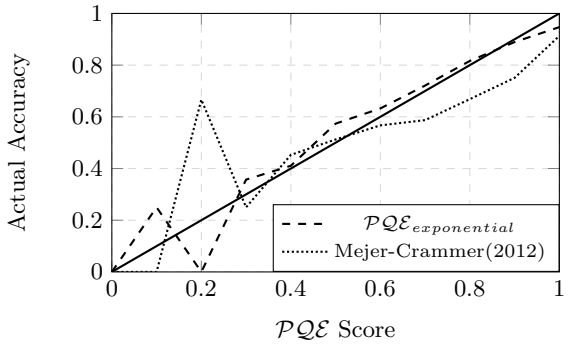


Fig. 4. Predicted vs Actual Accuracy comparison between our PQE system and Mejer and Crammer (2012) for English attachments

6 Error Detection in Parser Output

In this section, we empirically illustrate the efficacy of our proposed measure in automatic error detection.

6.1 Automatic Error Detection

Automatic error detection aims to efficiently determine and flag incorrectly predicted edges. The edges exhibiting high confusion scores are also highly probable to be incorrect, as the oracle is uncertain in its decision. Using this insight, an attachment or label is flagged as potential error if its confusion score is above a pre-calculated threshold(θ). In this task we have focused on identifying errors in two possible configurations i.e. attachment incorrectness and combined attachment & label incorrectness (either label or attachment is incorrect).

6.2 Data and Experimental Setup

We conducted experiments on 18 languages[3], using data from CoNLL-X [1], CoNLL 2007 [18] and MTPIL COLING 2012 [27] shared tasks on dependency

[3] Include all the languages in CoNLL-X and CoNLL 2007 shared task, except German and Czech due to unavailability of resources.

Table 1. Average results over 18 languages and results for English for automatic error detection task. EDI x% edges= Error detected on inspecting x% of total edges.

		Average						English					
	Measure	**F**	**P**	**R**	**EDI-1**	**EDI-5**	**EDI-10**	**F**	**P**	**R**	**EDI-1**	**EDI-5**	**EDI-10**
Attachment	**Baseline-I**	14.79	12.28	19.06	0.77	3.84	7.68	14.15	12.39	16.5	0.89	4.45	8.9
	Baseline-II	33.58	23.75	62.74	2.49	10.82	20.99	32.55	23.02	55.52	3.08	13.49	22.55
	Independent	39.69	33.63	52.64	2.15	10.40	24.56	46.68	47.02	46.34	6.16	24.53	38.88
	Linear	47.01	40.48	59.39	4.67	21.08	36.60	47.73	49.54	46.05	6.61	26.41	40.2
	Polynomial	45.82	38.31	61.32	4.66	19.77	34.69	47.65	48.73	46.63	6.61	**26.56**	**40.34**
	Exponential	**47.62**	40.54	60.33	**4.76**	**21.12**	**37.76**	**48.73**	45.67	52.22	6.61	**26.56**	**40.34**
	Regression	32.36	31.78	56.51	4.07	16.83	29.44	46.41	46.47	46.34	**7.17**	24.53	39.74
Attachment+Label	**Baseline-I**	21.47	18.00	28.80	0.78	3.92	7.84	14.6	12.8	16.99	0.83	4.14	8.28
	Baseline-II	34.06	21.38	99.96	1.11	5.51	10.74	26.53	15.29	100	2.9	14.18	23.49
	Independent	54.98	49.22	64.03	**3.97**	17.56	32.12	51.42	60.28	44.84	6.03	**27.19**	**42.35**
	Linear	55.99	49.39	65.49	3.88	17.76	32.67	**52.78**	52.24	53.33	6.03	26.34	40.94
	Polynomial	54.16	47.01	65.80	3.83	17.37	31.51	50.95	49.45	52.55	6.03	25.11	39.21
	Exponential	**56.20**	49.44	66.00	3.95	**17.79**	**32.80**	52.49	50.06	55.16	6.03	26.68	41.22
	Regression	42.48	41.19	62.66	3.81	16.18	28.96	36.02	44.36	30.33	**6.27**	20.52	29.93

parsing. We employ MaltParser version-1.7[5] [21]. We carried out experiments on the systems proposed in [22], [5] and [28], which are individually, the best performing MaltParser based systems, in their respective shared tasks. Best performing MaltParser based systems for all the languages use Arc-Eager mode of Nivre's algorithm[6] [19,20] except Chinese which uses Arc-Standard mode. All the results reported here are on the official test sets.

6.3 Identifying Optimum Threshold(θ)

Threshold(θ) is a crucial parameter in our experimental setup. An optimum θ is chosen by making use of the development set. Corresponding to each of the iteratively increasing candidate values for θ from minimum to maximum, the incorrect edges are flagged and *precision*, *recall* & *F-score* [13] are calculated. The value asserting the maximum *F-score* is chosen as threshold θ. Here for simplicity, we have used balanced *F-score*, i.e. F_1-*score*. However, as per the application and available resources, a relevant F_β can be chosen to maximize the yield on the input effort.

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{8}$$

Figure 5 depicts *precision*, *recall* and F_1-*score* corresponding to each candidate value of θ for Hungarian development data. The maximum F_1-*score* is attained at 0.36 which is thus taken as θ for Hungarian.

Since, CoNLL-X and CoNLL 2007 datasets did not provide development sets, we holdout 10% sentences of each training set as development data, using random sampling stratified on sentence length. The remaining training data is utilized to

[5] `http://www.maltparser.org/download.html`

[6] Nivre's Algorithm has two different modes namely, Arc-Eager and Arc-Standard.

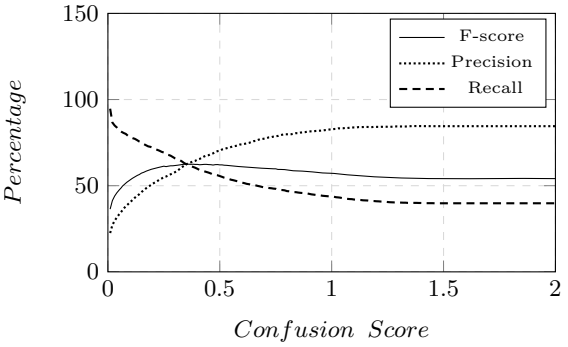


Fig. 5. Precision, recall and f-score for various values of confusion score on 'Hungarian' development set

train a parser model. The development set is again partitioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively utilized for parameter tuning as mentioned in section 4.2 (or training regression model) and threshold selection as explained earlier. However, the final training is performed on the entire training data and evaluation on the test set.

6.4 System and Baseline

We constructed two baselines for each configuration and language in our experiments. The first, Baseline-I, adopts a naive methodology of randomly selecting and marking errors. The number of errors to be marked is derived from the evaluation[7] on $\mathcal{D}_1$. The second, Baseline-II, assigns a confusion score to arc, equal to percentage of error of respective coarse POS tag, as per the evaluation on $\mathcal{D}_1$. Five systems to predict the confusion score of an arc are created as discussed in section 4.2 and 4.3 for 18 languages.

6.5 Results and Discussion

Table 1 exhibits the results obtained for automatic error detection. We present the F_1-*score*, *precision* and *recall* obtained over 18 languages in the task, with detailed results for English.

EDI: To efficiently capture the efficacy of our approach, another metric *EDI* (Error Detected on Inspecting x% of total edges) is presented which corresponds to the percentage of errors detected by inspecting 1%, 5% and 10% of total edges. The metric portrays a more precise view of the effort required to correctly identify parsing errors (often through manual validation).

Our experiments indicate that confusion score has a dependency on previous parser actions, as the $\mathcal{PQE}_{independent}$ (section 4.2) assumption is not found to perform well in comparison to other strategies of combining entropies. $\mathcal{PQE}_{exponential}$

[7] `http://nextens.uvt.nl/depparse-wiki/SoftwarePage/#eval07.pl`

is found to perform best, but marginally better than $\mathcal{PQE}_{linear}$ and $\mathcal{PQE}_{polynomial}$, for attachment error prediction. For joint prediction of attachments and labels, $\mathcal{PQE}_{independent}$, $\mathcal{PQE}_{linear}$, $\mathcal{PQE}_{polynomial}$ and $\mathcal{PQE}_{exponential}$ show comparable results. The regression based system $\mathcal{PQE}_{regression}$, however is not found to match up with the accuracies of these systems except the baselines. Best results are obtained for Portuguese while for Hindi results are not much distant from Baseline-II.

A comparison with [17] over English indicates our results ($\mathcal{PQE}_{polynomial}$ for Attachment) at par with them; 6.6% vs 7.17% (in ours) for 1% edge inspection, 27% vs 26.56% (in ours) for 5% and 46% vs 40.34% (in ours) for 10% edge inspection. This is only a tentative comparison since they reported results on Penn Treebank (55K words) while we use English data from CoNLL 2007 shared task (5K).

7 Conclusion and Future Work

This paper presents our effort towards computing a confusion score that can beforehand estimate, the correctness of the dependency parsed tree. The confusion score, accredited with each edge of the output, is targeted to give an informed picture of the parsed tree quality. We supported our hypothesis by experimentally illustrating that the edges with relatively higher confusion scores are the predominant parsing errors.

While much attention of parsing community is on improving parsers, our work stands out in identifying the potential in forecasting correctness at edge level. This may benefit the applications taking advantage of partial but precise parses like , self training [14], uptraining [24], parsing with partial trees [12], active learning [29] etc. by selectively dispensing only the quality segments of the parse. Not only parsed output, manual treebank validation too can benefit from such a score. An n-fold cross validation scheme can be adopted, in this case, to compute and assign confusion scores and detect annotation errors.

Currently we have not utilized any lexical features or parse's features which we think will be worth investigating in future. We have reason to believe, [17] have indirectly benefited from such feature since they use K-best parse from [6], which utilizes such features.

References

1. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 149–164. Association for Computational Linguistics (2006)
2. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and maxent discriminative reranking. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 173–180. ACL (2005)
3. Collins, M., Koo, T.: Discriminative reranking for natural language parsing. In: Machine Learning-International Workshop then Conference, pp. 175–182. Citeseer (2000)

4. Goldberg, Y., Elhadad, M.: Precision-biased parsing and high-quality parse selection. arXiv preprint arXiv:1205.4387 (2012)
5. Hall, J., Nilsson, J., Nivre, J., Eryiğit, G., Megyesi, B., Nilsson, M., Saers, M.: Single malt or blended? A study in multilingual parser optimization. In: Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, pp. 933–939 (2007)
6. Hall, K.: K-best spanning tree parsing. In: Annual Meeting-Association for Computational Linguistics, vol. 45, p. 392 (2007)
7. Hwa, R.: Sample selection for statistical parsing. Computational Linguistics 30(3), 253–276 (2004)
8. Jain, S., Agrawal, B.: A dynamic confusion score for dependency arc labels. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 1237–1242. Asian Federation of Natural Language Processing, Nagoya (2013), http://www.aclweb.org/anthology/I13-1176
9. Kawahara, D., Uchimoto, K.: Learning reliability of parses for domain adaptation of dependency parsing. IJCNLP 2008 (2008)
10. Kolachina, S., Kolachina, P.: Parsing any domain english text to conll dependencies. In: Calzolari N., Choukri, K., Declerck, T., DoÄ§an, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC 2012. European Language Resources Association (ELRA), Istanbul (May 2012)
11. Koo, T., Collins, M.: Hidden-variable models for discriminative reranking. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 507–514. ACL (2005)
12. Mannem, P., Dara, A.: Partial parsing from bitext projections. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1597–1606. Association for Computational Linguistics (2011)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
14. McClosky, D., Charniak, E., Johnson, M.: 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, pp. 152–159. Association for Computational Linguistics (2006)
15. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 523–530. Association for Computational Linguistics (2005)
16. McDonald, R.T., Nivre, J.: Characterizing the errors of data-driven dependency parsing models. In: EMNLP-CoNLL, pp. 122–131 (2007)
17. Mejer, A., Crammer, K.: Are you sure?: confidence in prediction of dependency tree edges. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 573–576. Association for Computational Linguistics (2012)
18. Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL, pp. 915–932. sn (2007)
19. Nivre, J.: An efficient algorithm for projective dependency parsing. In: Proceedings of the 8th International Workshop on Parsing Technologies, IWPT. Citeseer (2003)
20. Nivre, J., Hall, J., Nilsson, J.: Memory-based dependency parsing. In: Proceedings of CoNLL, pp. 49–56 (2004)

21. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kubler, S., Marinov, S., Marsi, E.: MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95 (2007)
22. Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., Marinov, S.: Labeled pseudo-projective dependency parsing with support vector machines. In: Proceedings of the Tenth Conference on Computational Natural Language Learning, pp. 221–225. Association for Computational Linguistics (2006)
23. Owczarzak, K.: Depeval (summ): dependency-based evaluation for automatic summaries. 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, vol. 1, pp. 190–198. Association for Computational Linguistics (2009)
24. Petrov, S., Chang, P.C., Ringgaard, M., Alshawi, H.: Uptraining for accurate deterministic question parsing. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 705–713. Association for Computational Linguistics (2010)
25. Ravi, S., Knight, K., Soricut, R.: Automatic prediction of parser accuracy. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 887–896. Association for Computational Linguistics (2008)
26. Settles, B.: Active learning literature survey. University of Wisconsin, Madison (2010)
27. Sharma, D.M., Mannem, P., van Genabith, J., Devi, S.L., Mamidi, R., Parthasarathi, R. (eds.) Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages. The COLING 2012 Organizing Committee, Mumbai, India (December 2012), http://www.aclweb.org/anthology/W12-56
28. Singla, K., Tammewar, A., Jain, N., Jain, S.: Two-stage Approach for Hindi Dependency Parsing Using MaltParser. Training 12041(268,093), 22–27 (2012)
29. Steedman, M., Hwa, R., Clark, S., Osborne, M., Sarkar, A., Hockenmaier, J., Ruhlen, P., Baker, S., Crim, J.: Example selection for bootstrapping statistical parsers. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 157–164. Association for Computational Linguistics (2003)
30. Tang, M., Luo, X., Roukos, S.: Active learning for statistical natural language parsing. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 120–127. Association for Computational Linguistics (2002)
31. Wann, S., Dras, M., Dale, R., Paris, C.: Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 852–860. Association for Computational Linguistics (2009)
32. Xu, P., Kang, J., Ringgaard, M., Och, F.: Using a dependency parser to improve smt for subject-object-verb languages. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 245–253. Association for Computational Linguistics (2009)
33. Zhang, Y., Clark, S.: A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 562–571. Association for Computational Linguistics (2008)

Experiments on Sentence Boundary Detection in User-Generated Web Content

Roque López and Thiago A.S. Pardo

Interinstitutional Center for Computational Linguistics (NILC), São Paulo, Brazil
Institute of Mathematical and Computer Sciences, University of São Paulo,
São Paulo, Brazil
{rlopez,taspardo}@icmc.usp.br

Abstract. Sentence Boundary Detection (SBD) is a very important prerequisite for proper sentence analysis in different Natural Language Processing tasks. During the last years, many SBD methods have been used in the transcriptions produced by Automatic Speech Recognition systems and in well-structured texts (e.g. news, scientific texts). However, there are few researches about SBD in informal user-generated content such as web reviews, comments, and posts, which are not necessarily well written and structured. In this paper, we adapt and extend a well-known SBD method to the domain of the opinionated texts in the web. Particularly, we evaluate our proposal in a set of online product reviews and compare it with other traditional SBD methods. The experimental results show that we outperform these other methods.

Keywords: Sentence Boundary Detection, Noisy Text Processing, User Generated Content.

1 Introduction

In the last decade, many websites have appeared where users may freely generate content and with few restrictions. Websites such as forums, wikis and product review sites have become big repositories of information about different topics. Unfortunately, in these websites, the vast majority of this information is usually written in an informal and, sometimes, ill-formed way, not following orthography and grammar rules. For instance, in product reviews, it is very common to find a lot of noise, such as spelling mistakes, non-standard abbreviations and missing or inadequate sentence boundary marks [9].

Sentence Boundary Detection (SBD) is the focus of this paper. This task consists in identifying the sentences within a text [26]. In Automatic Speech Recognition (ASR), it is very popular due to the necessity of finding sentential segments in the stream of words (the transcripts) that are automatically recognized. In text processing, it is essential to produce the input – the sentences – to other tools (as POS tagger and parser) and applications (as information extraction and summarization).

In the majority of the languages, the period (".") is usually employed as sentence boundary marker, while it may also be used in abbreviations, acronyms,

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 227–237, 2015.
DOI: 10.1007/978-3-319-18111-0_18

ordinal numbers, e-mails and URLs. The variety of applications of the period mark represents a challenge in the SBD task. In online user-generated content, this challenge is even greater because this marker is usually omitted or not properly used. Figure 1 shows an example of a (real) product review written in Brazilian Portuguese and translated into English. As we may see, users generally do not use the period mark to delimit sentences as well as do not respect the use of other punctuation marks (as commas and semicolons) or capital letters, making more challenging the SBD process and, consequently, the other tasks that depend on it.

```
Prós: O telefone deve ser ótimo

Contras: Cuidado com a Empresa_X...tem preço bom
mas péssima entrega (Empresa_Y é palhaçada)

Opinião: Não recomendo a ninguém comprar na
Empresa_X;nop e-commerce eles são piores que o
Empresa_Z


[Possible translation]

Pros: The phone must be great

Cons: Beware the Company_X...it has good price
but bad delivery (Company_Y is a joke)

Opinion: I do not recommend anyone to buy in
Company_X;in the e-commerce they are worse than
the Company_Z
```

Fig. 1. Example of online product review[1]

To demonstrate the relevance of tackling such issues, [9] recently presented an analysis of different kinds of noise in online product reviews written in Brazilian Portuguese. In that study, the manual correction of punctuation marks led to an improvement of 4.34% in the precision of the POS tagger.

In this paper, we explore SBD methods in user-generated web content. We start by adapting and extending the supervised machine learning method proposed in [25]. This is one of the most classical methods and, unlike other ones, does not use prosodic information (e.g., rhythm, stress, or intonation) – as it usually happens in the ASR context – and thus it is suitable for written texts. We also evaluate two other SBD systems, MxTerminator [22] and Punkt [11], which are considered state of the art systems. In particular, for training the machine learning method, we use well-written news texts, expecting that patterns for good usage of period mark may be learned and used for SBD in user-generated content.

We opted to run our experiments on texts of the same corpus used in [9], which is composed of product reviews written in Brazilian Portuguese, retrieved

[1] Company names were omitted in this figure due to ethical concerns.

from a product evaluation webpage. We agree with [9] that such texts are good representatives of the writing phenomena that occurs in user-generated web content. Finally, as our database is in Portuguese, we use some corpora of news texts in this language for training the machine learning solution, adopting, in the end, the publicly available CSTNews corpus [7].

We show that our results outperformed the other state of the art methods and, interestingly, that a large training corpus is not necessary for achieving good results. The remaining of the paper is organized as follows: in Section 2, we introduce the main related work; in Section 3, we describe the proposed method to identify sentence boundaries; the experiments and results are presented in Section 4; finally, in Section 5, we conclude this paper.

2 Related Work

There are many approaches used to detect sentence boundaries in different languages. According to [24], SBD systems are grouped in two classes: methods that use fixed rules and methods that use machine learning techniques. These methods have been well studied in the ASR area and are widely applied in news texts [5][11][21][25]. In this section, we comment some of these methods.

For Brazilian Portuguese language, [24] is one of the first works in SBD, with very interesting results. The authors compare the performance of two systems that use machine learning methods (MxTerminator [22] and Satz [16]) and one system based on fixed rules (RE SYSTEM [23]). These systems were evaluated in a corpus of news texts in two scenarios: (i) when the domain of the texts is known in advance, the results of these systems were similar, and (ii) when the domain is unknown, the best results were obtained by the machine learning methods. The main reason for these results is that rules are dependent on the domain.

MxTerminator [22], tested in the work above mentioned, and Punkt [11] are language-independent SBD systems and have been used in many languages, including Portuguese. MxTerminator uses a statistical approach based on Maximum Entropy to identify the sentences of a document. From a corpus with the sentences already identified, this method learns the contextual information where sentence boundaries occur. For this, MxTerminator uses some features, such as the preceding token, the following token, and capitalization information. For Brazilian Portuguese, MxTerminator showed a robust performance (96.46 of F-measure) in the Lacio-Web Corpus [3] using 10-fold cross-validation. Punkt is an unsupervised SBD system based on the assumption that, once abbreviations have been identified, it is more feasible to identify sentence boundaries. For this, Punkt uses properties of abbreviations to identify them and considers that all periods not attached to an abbreviation are sentence boundaries. Additionally, Punkt uses some heuristics (e.g., the presence of digits followed by a period mark) to identify name initials and ordinal numbers. For Brazilian Portuguese, Punkt outperformed the results of MxTerminator with 97.22 of F-measure in the same corpus (Lacio-Web).

[17] presents SENTER, a rule-based system to sentence segmentation of well written texts. This system is very simple and uses some general heuristics to detect sentence boundaries (such as the presence of newline characters and the different possibilities where the period mark is not a sentence boundary symbol). In that work, the authors do not present an evaluation about the performance of SENTER.

In the ASR area, [5] made experiments concerning punctuation and capitalization recovery for spoken texts about news in European Portuguese. In order to recover the period mark, the authors use maximum entropy models with some features like n-grams, POS tags and prosodic information. In the experiments, the authors show that lexical features had less impact than prosodic features, but the combination of all features produced better results.

For informal user-generated content, there are few researches on SBD. [21] evaluated several SBD systems in news texts and user-generated content written in English. As expected, the lowest results were obtained in informal texts, because, according to the authors, in these texts there is a decline in linguistic formality. For Brazilian Portuguese, as far as we know, there is still no SBD works for informal texts. For others languages, like Arabic and Chinese, there are some efforts. [1] uses common words as sentence delimiting symbols in Arabic texts, and [27] presents a maximum entropy model-based approach to predict and correct punctuation marks to segment sentences written in Chinese.

In this paper, we test MxTerminator and Punkt in the intended scenario and compare their results with the main method that we explore in this paper, which we introduce in what follows.

3 Our Approach

The proposed approach in this study is an adaptation of the supervised machine learning method proposed in [25]. In that work, the authors introduced the problem of SBD on the text produced by ASR systems and used written texts to evaluate their proposal. The authors used the Timbl memory-based learning algorithm [8] with a set of twelve features derived from the analysis of the preceding and following words in relation to the point where the punctuation should be included. To train and test their method, they used news of the Wall Street Journal.

Before detailing the features and our approach, it is important to clarify how to model the task as a machine learning solution. According to [12], the SBD problem may be represented as a classification task in the following way: for each word in the text, we determine whether it is or not a sentence boundary, i.e., each word (the learning instance) might be classified as belonging to either the *boundary* class or the *no_boundary* class. Words of the *boundary* class are those that should be followed by a period mark. This is the general learning schema used in [25] and is adopted in this work.

In our proposal, in addition to consider the twelve features used in [25], we experiment an extended version with two more features: (i) a flag indicating whether the following token is a newline mark and (ii) a flag indicating whether the following token is a period mark. In Table 1, we show the fourteen features used in this paper: the first twelve features are those used in [25] and the last two features are the ones proposed above. While some of these features are computed in traditional ways, some deserve explanations.

Table 1. Features used in the proposed approach

Id	Feature
F1	The preceding word
F2	Probability that the preceding word ends a sentence
F3	Part of speech tag assigned to the preceding word
F4	Probability that the above part of speech tag (feature F3) is assigned to the last word in a sentence
F5	Flag indicating whether the preceding word is a stopword
F6	Flag indicating whether the preceding word is capitalized
F7	The following word
F8	Probability that the following word begins a sentence
F9	Part of speech tag assigned to the following word
F10	Probability that the above part of speech tag (feature F9) is assigned to the first word in a sentence
F11	Flag indicating whether the following word is a stopword
F12	Flag indicating whether the following word is capitalized
F13	Flag indicating whether the following token is a period mark
F14	Flag indicating whether the following token is a newline mark

We propose the newline mark as a feature because, in product reviews, users usually use this symbol as a sentence boundary. It is very common in online informal texts. In the case of the period mark, we consider this feature because, although they are rarely used, when users use this symbol, it is very likely that it is a sentence delimiter. For this feature, using regular expressions, we previously filter out occurrences of period marks that are decimal points or parts of e-mails and URLs.

For capitalized words (feature F6 and F12), we perform a simple analysis. We verify if the first letter is the only capitalized letter, because, in product reviews, users do not respect the correct use of capitalized words. Cases like *PRODUTO RUIM* (BAD PRODUCT, in English) or *BoM SeRvIÇo* (GoOd SeRvIcE, in English) are very common. These examples, with mixed letter cases, we consider as lowercase words. We believe that when users employ these types of words they want to highlight an expression and not to start a new sentence.

For features F3 and F9, [25] used the POS tags manually annotated in the news of the Wall Street Journal. In our case, the product reviews do not present previously annotated POS tags. For this reason, we followed a probabilistic approach. This approach uses as data source the Mac-Morpho corpus [2], in which

each word shows the corresponding correct POS tag. To tag a word in product reviews, our approach searches the most likely tag for that word, i.e., the tag that is the most used one in the above corpus. For words not present in MacMorpho, we used the first listed tag in the DELAF dictionary [15], in which each word is associated to all its possible tags. In the case a word is not present in both sources, we consider it as a noun. As an alternative, a traditional POS tagger might be used, but, in product reviews, there are many noises that affect the performance of POS taggers. This motivated us to use the probabilistic approach.

Once we have the features, we used Naïve Bayes as the machine learning method, specifically the version implement in scikit-learn library [18]. We also conducted some experiments with others machine learning methods (SVM, k-Nearest Neighbors and Stochastic Gradient Descent), but Naïve Bayes got the best results. For this reason, we only report its results. As said before, we train our method with well-written news texts, expecting that we may learn patterns of good usage of period marks to detect (in the test phase) where user-generated texts need segmentation. We describe the corpora we tested in the next section. It is also important to say that it was not possible to use user-generated texts for training our method because there is no such manually annotated data available, to the best of our knowledge.

As input, our method receives a product review in plain text format. After that, we eliminate all punctuations marks and, for each word in the text, we extract the features showed in Table 1. With these features, our proposal determines whether the word evaluated is at a sentence boundary position (and should have a period mark inserted after it) or not. Finally, an output is generated with the detected sentence boundaries. In Figure 2, we show an example of input and output of our method. The input and output are in the top and bottom of the figure, respectively.

```
Prós: O telefone deve ser ótimo

Contras: Cuidado com a Empresa_X...tem preço bom
mas péssima entrega (Empresa_Y é palhaçada)

Opinião: Não recomendo a ninguém comprar na
Empresa_X;nop e-commerce eles são piores que o
Empresa_Z
```

```
Prós: O telefone deve ser ótimo.

Contras: Cuidado com a Empresa_X... tem preço
bom mas péssima entrega (Empresa_Y é palhaçada).

Opinião: Não recomendo a ninguém comprar na
Empresa_X; nop e-commerce eles são piores que o
Empresa_Z.
```

Fig. 2. Examples of input and output data in our proposed approach

4 Experiments

4.1 Datasets

To train the proposed method, we used the CSTNews corpus [7], a collection of news texts written in Brazilian Portuguese. As CSTNews is a small corpus, we initially did experiments with much larger corpora, using the Corpus NILC [19] and PLN-Br GOLD corpus [6] in the training phase, but, surprisingly, the results were not better. More than this, in Corpus NILC there were some sentences (tittles of the news) without period marks, and this affected the learning process. In the case of PLN-Br GOLD corpus, the results were similar but it took a long time to process all the documents.

As the results were not better with the larger corpora, we believe that our proposal have a good learning process with few data. For these reasons, we only used the CSTNews corpus in the training phase and the results we report here are based on this corpus.

The CSTNews is a corpus composed of 140 news texts grouped in 50 clusters. Each cluster contains from 2 to 3 news texts on the same topic compiled from some of the main online newspapers in Brazil. These texts are news about sports, politics, science and others. In total, this corpus has 2067 sentences. Additionally, CSTNews has other types of manual annotations, like CST (Cross-document Structure Theory) [20], RST (Rhetorical Structure Theory) [14], multi-document summaries and their alignment with the corresponding source texts, among other annotation layers. In this study, we only use the full texts of this corpus.

To test our methods, we used the corpus of product reviews described in [9], which were collected from Buscapé[2], a website where users comment about different products (e.g., smartphones, digital cameras, notebooks, etc.). These comments are written in a free format within a template with three sections: Pros, Cons, and Opinion.

To conduct the experiments, we used a sample of 35 product reviews annotated by a computational linguist. The annotation consisted in deletions, insertions or substitutions of punctuations marks to correct the texts. This data was all the data that we had available for testing the methods.

4.2 Results

In the experiments, we evaluated our proposal, the original method proposed by [25], and two state-of-the-art SBD systems: MxTerminator [22] and Punkt [11], which have the highest results reported in the literature [26]. For the experiments, we use the implementations of OpenNLP [4] and NLTK [13] libraries for MxTerminator and Punkt, respectively. We also tried to use the sentence separator proposed by [26], but the online version was not working for Portuguese texts.

Table 2 shows the results of our experiments with Precision, Recall and F-measure metrics. Precision is computed as *tp / (tp + fp)*, where *tp* is the number

[2] `http://www.buscape.com.br/`

of true positives and *fp* the number of false positives. Recall is the ratio *tp / (tp + fn)*, where *tp* is the number of true positives and *fn* the number of false negatives. F-measure is the harmonic mean of precision and recall, being a unique indicator of the quality of the method. The overall results shown in Table 2 are the averages over the *boundary* and *no_boundary* classes. One may see that our proposal obtained the best results.

Table 2. Overall results

Method	Precision	Recall	F-Measure
MxTerminator [22]	0.939	0.847	0.886
Punkt [11]	0.943	0.843	0.885
Original Approach [25]	0.801	0.834	0.817
Proposed Approach	**0.953**	**0.895**	**0.921**

With the use of news texts in the training process, the results were good, showing that good patterns could be learned, as we had hypothesized before. We may also see that the results obtained by our proposal are better than the original method proposed by [25] in Precision, Recall and F-measure, reflecting that our two additional features (F13 and F14) helped improving the performance of the method.

It is important to highlight that, using the Student's t-test with 95% of confidence, the differences between the F-measures obtained by our proposal and the other methods are statistically significant. In relation to the general accuracy, our proposal also got the best results, with 97.60%, while the original method achieved 93.70%, and MxTerminator and Punkt 96.70%.

We used the Relief algorithm [10] to evaluate the importance of each feature of our proposal. Of the fourteen features used, the probability that the POS tag is assigned to the first word in a sentence (F10) and the POS tag assigned to the following word (F9) do not contribute to the final performance. In other words, removing these features does not affect the results. However, if we remove any other feature, the performance decreases. The three best features were the presence of the newline mark (F14), the probability that the following word begins a sentence (F8) and the presence of the period mark (F13).

In order to analyze the performance of our proposal in more details, we show, in Table 3, the results obtained by the four SBD systems for the words that belong to the *boundary* class (the *yes* class, therefore). Clearly, the proposed approach in this paper got the best results.

We attribute these best results to the special analysis that we made of the product reviews characteristics, such as usage of capitalization, use of newline marks and POS tags in informal texts. On the other hand, a very common error made by our method in identifying boundaries occurred when words are unknown. These new words are not present in the training corpus and our proposal cannot learn patterns when these words are followed by the period mark. These new words may simply be unseen words in the training data as well as

Table 3. Results for the *boundary* class

Method	Precision	Recall	F-Measure
MxTerminator [22]	0.907	0.701	0.791
Punkt [11]	0.915	0.693	0.789
Original Approach [25]	0.632	0.708	0.668
Proposed Approach	**0.925**	**0.795**	**0.855**

spelling mistakes, foreign words or slangs, which are typical elements in product reviews.

We believe that, if we use an annotated corpus of reviews in the training phase, these types of errors would not frequently occur because the machine learning method would identify and learn, with more coverage, the features of this domain. In this context, Recall measure, that was low (see Table 3), would improve.

It was also evaluated the performance of the SBD systems for the *no_boundary* class (the *no* class). The obtained results are presented in Table 4. In comparison with Table 3, the performances are much better and there is little difference among the four methods. It is because the majority of words in texts are not sentence boundaries, and, thus, there are more instances of the *no_boundary* class in the training phase. We believe that this unbalance in the data influenced the results for this class.

Table 4. Results for the *no_boundary* class

Method	Precision	Recall	F-Measure
MxTerminator [22]	0.971	0.993	0.982
Punkt [11]	0.971	**0.994**	0.982
Original Approach [25]	0.971	0.960	0.965
Proposed Approach	**0.980**	**0.994**	**0.987**

In relation to other romance languages, such as Spanish or French, we believe that it is possible to use the fourteen features of our proposal and get satisfactory results, because these languages share some common linguistics characteristics like the basic *subject–verb–object* order. In addition, we believe that internet users of these languages have similar behavior when they generate web content (e.g., use of newline marker). However, for other languages, such as Chinese or Japanese, it is complicated to use our approach because their linguistic characteristics are different and some of our machine learning features are not present in these languages, such as the capitalization rule (features F6 and F12).

5 Conclusion and Future Work

In this work, we analyzed the SBD problem in user-generated web content. As it may be seen, we adapted and extended a classical approach to the problem and

outperformed other state of the art systems. This research has been motivated, mainly, by the importance of the SBD systems in the preprocessing of web texts for posterior processing by other NLP tools.

As a future work, we plan to study the use of the above methods for detecting other punctuation marks, as comma and semicolon, which must be bigger challenges to deal with, since their usage is more flexible in several different situations.

Acknowledgments. Part of the results presented in this paper were obtained through research on a project titled "Semantic Processing of Texts in Brazilian Portuguese", sponsored by Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91.

References

1. Al-Subaihin, A., Al-Khalifa, H., Al-Salman, A.: Sentence Boundary Detection in Colloquial Arabic Text: A Preliminary Result. In: Proceedings of the International Conference on Asian Language Processing, pp. 30–32 (2011)
2. Aluísio, S., Pelizzoni, J.M., Marchi, A.R., de Oliveira, L., Manenti, R., Marquiafável, V.: An Account of the Challenge of Tagging a Reference Corpus for Brazilian Portuguese. In: Mamede, N.J., Baptista, J., Trancoso, I., Nunes, M.d.G.V. (eds.) PROPOR 2003. LNCS, vol. 2721, pp. 110–117. Springer, Heidelberg (2003)
3. Aluísio, R.M., Pinheiro, G., Finger, M., Nunes, M.G., Tagnin, S.: The LacioWeb Project: Overview and Issues in Brazilian Portuguese Corpora Creation. In: Proceedings of Corpus Linguistics, pp. 14–21 (2003)
4. Baldridge, J.: The OpenNLP Project (2005), `http://opennlp.apache.org/index.html` (accessed January 15, 2015)
5. Batista, F., Caseiro, D., Mamede, N., Trancoso, I.: Recovering Capitalization and Punctuation Marks for Automatic Speech Recognition: Case Study for Portuguese Broadcast News. Speech Communication 50(10), 847–862 (2008)
6. Bruckschen, M., Muniz, F., Souza, J., Fuchs, J., Infante, K., Muniz, M., Gonçalves, P., Vieira, R., Aluísio, S.: Anotação Linguística em XML do Corpus PLN-BR. Série de Relatórios do NILC, NILC-TR-09-08 (2008)
7. Cardoso, P.C., Maziero, E.G., Jorge, M., Seno, E.M., Di Felippo, A., Rino, L.H., Nunes, M.D.G.V., Pardo, T.A.: CSTNews-A Discourse-Annotated Corpus for Single and Multi-Document Summarization of News Texts in Brazilian Portuguese. In: Proceedings of the 3rd RST Brazilian Meeting, pp. 88–105 (2011)
8. Daelemans, W., Jakub, Z., Van Der Sloot, K., Van Den Bosch, A.: TiMBL: Tilburg Memory Based Learner-Version 2.0 - Reference Guide (1999)
9. Duran, M., Avanço, L., Aluísio, S., Pardo, T., Nunes, M.: d.G.: Some Issues on the Normalization of a Corpus of Products Reviews in Portuguese. In: Proceedings of the 9th Web as Corpus Workshop (WaC-9), pp. 22–28 (2014)
10. Kira, K., Rendell, L.A.: The Feature Selection Problem: Traditional Methods and a New Algorithm. In: Proceedings of the 10th National Conference on Artificial Intelligence, pp. 129–134 (1992)
11. Kiss, T., Strunk, J.: Unsupervised Multilingual Sentence Boundary Detection. Computational Linguistics 32(4), 485–525 (2006)

12. Liu, Y., Chawla, N.V., Harper, M.P., Shriberg, E., Stolcke, A.: A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech. Computer Speech & Language 20(4), 468–494 (2006)
13. Loper, E., Bird, S.: NLTK: The Natural Language Toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, pp. 63–70 (2002)
14. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: A Theory of Text Organization. University of Southern California, Information Sciences Institute (1987)
15. Muniz, M.C., Nunes, M.D.G.V., Laporte, E.: UNITEX-PB, a set of Flexible Language Resources for Brazilian Portuguese. In: Workshop on Technology on Information and Human Language, pp. 2059–2068 (2005)
16. Palmer, D.D., Hearst, M.A.: Adaptive Multilingual Sentence Boundary Disambiguation. Computational Linguistics 23(2), 241–267 (1997)
17. Pardo, T.A.S.: SENTER: Um Segmentador Sentencial Automático para o Português do Brasil. Série de Relatórios do NILC, NILC-TR-06-01 (2006)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
19. Pinheiro, G.M., Aluísio, S.M.: Corpus Nilc: Descrição e Análise Crítica com Vistas ao Projeto Lacio-Web. Série de Relatórios do NILC, NILC-TR-06-09 (2003)
20. Radev, D.R.: A Common Theory of Information Fusion from Multiple Text Sources Step One: Cross-document Structure. In: Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue, pp. 74–83 (2000)
21. Read, J., Dridan, R., Oepen, S., Solberg, J.L.: Sentence Boundary Detection: A Long Solved Problem? In: Proceedings of 24th International Conference on Computational Linguistics, pp. 985–994 (2012)
22. Reynar, J.C., Ratnaparkhi, A.: A Maximum Entropy Approach to Identifying Sentence Boundaries. In: Proceedings of the 5th Conference on Applied Natural Language Processing, pp. 16–19 (1997)
23. Silla, C., Kaestner, C.: Automatic Sentence Detection Using Regular Expressions. In: Proceedings of the 3rd Brazilian Computer Science Congress, pp. 548–560 (2003) (in Portuguese)
24. Silla, C., Kaestner, C.: An Analysis of Sentence Boundary Detection Systems for English and Portuguese Documents. In: Computational Linguistics and Intelligent Text Processing, pp. 135–141 (2004)
25. Stevenson, M., Gaizauskas, R.: Experiments on Sentence Boundary Detection. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 84–89 (2000)
26. Wong, D.F., Chao, L.S., Zeng, X.: iSentenizer-μ: Multilingual Sentence Boundary Detection Model. The Scientific World Journal 2014 (2014)
27. Zhao, Y., Fu, G.: A MEMs-based Labeling Approach to Punctuation Correction in Chinese Opinionated Text. In: Proceedings of the 2013 International Conference on Intelligence Artificial, pp. 329–336 (2013)

Anaphora Resolution and Word Sense Disambiguation

An Investigation of Neural Embeddings for Coreference Resolution

Varun Godbole, Wei Liu, and Roberto Togneri

The University of Western Australia
20742098@student.uwa.edu.au, {wei.liu,roberto.togneri}@uwa.edu.au

Abstract. Coreference Resolution is an important task in Natural Language Processing (NLP) and involves finding all the phrases in a document that refer to the same entity in the real world, with applications in question answering and document summarisation.Work from deep learning has led to the training of neural embeddings of words and sentences from unlabelled text. Word embeddings have been shown to capture syntactic and semantic properties of the words and have been used in POS tagging and NER tagging to achieve state of the art performance. Therefore, the key contribution of this paper is to investigate whether neural embeddings can be leveraged to overcome challenges associated with the scarcity of coreference resolution labelled datasets for benchmarking. We show, as a preliminary result, that neural embeddings improve the performance of a coreference resolver when compared to a baseline.

Keywords: coreference resolution, neural embeddings, deep learning.

1 Introduction

Coreference Resolution is the Natural Language Processing (NLP) task of finding all the terms in a piece of text that refer to the same entity in the real world. For example, consider the sentences: "Bob went to the beach. He loved the water.". A coreference resolver should tell you that "Bob" and "He" refer to the same entity in the real world. A *mention* is a phrase that refers to an entity in the real world and two mentions are said to corefer if they both refer to the same entity in the real world.

Deep neural networks have recently been shown to provide impressive state of the art performance in a wide array of machine learning tasks from object recognition [1] to paraphrase detection [2]. A compelling result from the research in training deep neural networks is that they learn good representations, or features, of the data from the data, while being trained for some task. A trained network can then potentially be used as a feature extractor for tasks the network may not have been explicitly trained for [3].

Work in deep learning has led to neural word embeddings [4], which are distributed vector representations of words. Subsequent work has led to distributed vector representations for sentences [2,5]. Surprisingly, it has been shown that these neural word embeddings capture syntactic and semantic regularities in the

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 241–251, 2015.
DOI: 10.1007/978-3-319-18111-0_19

words [6]. For example, the resulting word vector for "King" minus vector for "man" plus the vector for "woman" leads to a vector closest to the vector for "Queen" [6]. Given the wealth of information these embeddings capture, one can leverage this information by using these embedding vectors as features in a task that may not have sufficient labelled training data.

In fact, neural embedding vectors have been used as features in existing baselines for other NLP tasks, such as Part-Of-Speech (POS) tagging [7] or Named Entity Recognition (NER) tagging [7].

The newest benchmark dataset for coreference resolution, the CoNLL-2012 Shared Task [8] dataset provides annotations for hundreds of thousands of words. However, the actual number of coreferent mentions is a very small fraction of this dataset, and this problem is exacerbated with the fact that a coreference resolver must account for proper (e.g. Isaac Newton), nominal (e.g. scientist) and pronominal (e.g. he) mentions. This problem is even further exacerbated when one considers that these mentions must first be predicted and tagged before a coreference resolver can resolve them, as was the case for the CoNLL-2012 Shared Task. Therefore, for this task, it is crucial to find good features and to use them effectively.

This challenge of a scarcity of large labelled datasets for coreference resolution motivates our work. Given that neural embeddings capture a lot of syntactic and semantic information about the words and phrases they represent, and given that they have led to state of the art performance in the other NLP tasks discussed above, we seek to investigate these embedding vectors in the context of coreference resolution, which, to our knowledge is the first such investigation.

The key contribution of this paper is to show, as a preliminary result, that neural embeddings improve the performance of a coreference resolution system when compared to a baseline, with discussions as to why this might be the case. That is, we demonstrate the validity of the concept that neural embeddings should be used as features for coreference resolution. Given that the emphasis of this investigation was on determining whether these neural embeddings are in fact good features for this task, care was taken to ensure that all changes in performance were due to the presence (or absence) of these neural embeddings.

In Section 2, we consider the related work in coreference resolution. In Section 3 we describe the methodology of our investigation. Section 4 contains the results of our investigation along with some discussion of the results. Finally, Section 5 contains the conclusions of this investigation and some promising avenues for future work.

2 Related Work

As discussed above, coreference resolution involves finding all the phrases in a document or a piece of text that refer to the same entity in the real world. These phrases are called *mentions*. When two mentions refer to the same entity in the real world, they are said to be *coreferent*. For a given mention, if there are coreferent mentions earlier in the text then they are called the mention's

antecedents. A *cluster* of mentions is simply a collection of mentions that are all coreferent with each other. This task can be broadly broken up into the two subtasks of mention identification (i.e. predicting all the mentions) and mention resolution (i.e linking all the mentions together).

There have been a variety of machine learning approaches proposed to tackle this problem, including pairwise models [9,10], graphical models [11,12] and ranking models [13]. However, many machine learning approaches use a pairwise classifier that considers a pair of mentions at a time to predict if they are coreferent [9,10,14].

A pairwise classifier can only predict whether a pair of mentions is coreferent at a time, but a document may contain many mentions. Therefore, another algorithm is necessary that visits all pairs of mentions in a document and uses the output of the pairwise classifier as necessary. Suppose an antecedent needs to be found for mention m_i, for the ith mention in a document. Then a common strategy [10] is to start with the pair of mentions m_{i-1} and m_i and walk backwards in the document until the classifier finds a mention that is coreferent with m_i. This can be called the *closest first strategy* [15]. Alternatively, if the classifier model produces a confidence score associated with the prediction, one could keep searching backwards and then choose the best antecedent that is predicted as coreferent for each mention m_i. This can be called the *best first strategy* [14].

Pairwise classifiers make a number of naïve assumptions, for example, they consider the coreference of each pair of mentions independently of other mentions. This can be a bad assumption given that the links between mentions are transitive. That is, if mention A and mention B are coreferent, and mention B and mention C are coreferent, then mention A and C are also coreferent. However, despite these challenges, they remain a popular approach due to their simplicity.

With respect to ranking based approaches, the work due to Durrett et. al. [13] is especially interesting because they demonstrate state of the art performance using simple surface level features and a log-linear model [16] to rank the best antecedent for each mention candidate. Some examples of syntactic surface level features are the first and last words of the mention, the full string of the mention, the head word of the mention, etc. Some examples of surface level semantic features might involve checking whether the number and gender match, the NER tags of the two mentions, etc. This work is interesting because they show that despite using simple surface level features, because of the complex interactions of these features, their system can yield good performance. Therefore, given that neural embeddings have been shown to capture syntactic and semantic relationships, one would expect that using neural embeddings as features instead of these surface level features may help. This line of thinking has been explored in our investigations below.

3 Methodology

Figure 1 provides a block diagram representation describing the process under which our investigations were carried out. In the figure, the preprocessing block

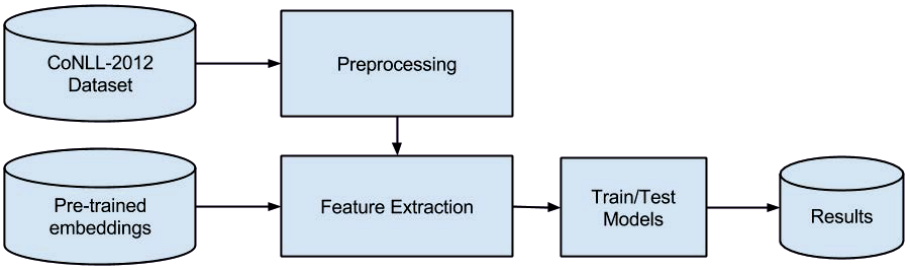


Fig. 1. A block diagram representing how the investigations were performed

simply refers to a conversion from the CoNLL format to an intermediate JSON format to make experimentation with feature extraction easier.

The CoNLL-2012 Shared Task dataset [8] is the latest standard benchmark dataset for coreference resolution and we used this dataset for all our experiments. This dataset contains corpora for English, Chinese and Arabic but all our experiments were restricted to English. The dataset also makes the distinction between gold mentions and predicted mentions, where for gold mentions, the mentions have already been identified and coreference links need to be predicted.

Given that our work has been motivated by a desire to find better features, in order to better achieve control over the results of our experiments, all experiments use gold mentions. This is also why all our experiments use a simple pairwise classifier and a best-first resolution strategy, consistently through all the experiments. A very simple model and resolution strategy has the effect of further ensuring that all changes in performance are due to the quality of the features of the model. A pairwise classifier, that is, a classifier that considers pairs of mentions at a time for coreference implies that this is a binary classification task with outcomes being either coreferent or not coreferent.

The CoNLL-2012 Shared Task uses the unweighted mean of three metrics, the MUC [17], the B^3 and CEAFE [18] to determine the final CoNLL Score. The MUC generates a score based on the quality of the links, whereas B^3 and CEAFE generate a score based on the quality of the entities (i.e clusters of mentions). BLANC [19] is a link based metric that has been proposed as a replacement metric to gauge the resolution of mentions, as opposed to a combination of the resolution and the identification of mentions.

To facilitate the advancement of the state of the art, some researchers make available online for download, embedding vectors they have trained on very large corpora (e.g. 100 billion words). In this paper, our investigations make use of two standard sets of embedding vectors: the word vectors generated by the **word2vec** [1] library released online by Mikolov et. al [20]; and we also generate phrase/sentence vectors from the code[2] released by Socher et. al. [2].

[1] `https://code.google.com/p/word2vec/`
[2] `http://nlp.stanford.edu/~socherr/classifyParaphrases.zip`

Specifically we used the **word2vec** that were trained using the *skip-gram* objective. This can be seen as a regression problem, where a window is moved across the text corpus and for a given word, we try to predict the real valued word vectors of the surrounding words. The code released by Socher et. al. [2] contains an implementation of an unfolding recursive autoencoder, which generates vector representations of phrases by first generating a parse tree and then using a recursive autoencoder to follow the parse tree. Unlike a traditional recursive autoencoder, an unfolding recursive autoencoder tries to unfold/reconstruct the entire subtree, which allows it to capture more information.

Concretely, all experiments conducted can be broadly classified into two categories: NGRAM-i experiments and SENTENCE-j experiments. Given a set of baseline features f_1 to f_n, i.e. $F_{baseline} = \{f_1, \ldots, f_n\}$ in each experiment, vector representations for either words, as in NGRAM-i, or sentences, as in SENTENCE-j, are chosen and then concatenated onto $F_{baseline}$. Both experiment sets share the same baseline features.

The NGRAM-i experiments involve choosing i words around the mention in a breadth-first manner and then using their embedding vectors from the pre-trained vectors released online. The SENTENCE-j involve choosing j sentences on either side of the mention, as well as the sentence the mention is in, and then generating their respective vectors using the code released online. That is, a SENTENCE-j experiment will involve $2j + 1$ sentences being chosen.

To provide a suitable comparison for the NGRAM and SENTENCE experiments, a baseline model was trained using syntactic, semantic and positional features as was used in recent literature. The features used in the baseline include distances (in words) between the mentions, whether the mention was demonstrative or a proper name, whether it was a pronoun, whether the mention was definite, the length of the longest common substring between the mentions, whether the mention was quoted, the NER tags in each mention and the number and gender of the mention based on the surface level properties and POS tags of the mentions.

In designing the experiments, the training sets were carefully constructed because it has been shown that having an overly skewed training set can be detrimental for performance [21]. That is, it is desirable to have the number of positive examples in the training set be roughly equal to the number of negative examples. Therefore, one possible strategy for generating positive examples in the training set, as described by Soon et. al. [10], for a given mention, is to choose the closest preceding coreferent mention as the pair of mentions from which features are extracted. Negative examples are then just the non-coreferent mentions in between this pair of mentions. This also ensures the data distribution of the feature vectors generated in the training set is close to what the classifier might see in the test set if something akin to a closest-first resolution strategy is used.

Table 1. Scores on the CoNLL-2012 Shared Task test set. (R: Recall, P:Precision)

Experiments	MUC			B^3			CEAFE			BLANC			CoNLL
	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	
Baseline	72.75	72.08	72.41	68.77	47.10	55.91	47.94	49.43	48.68	66.59	57.46	56.08	59.00
NGRAM-2	59.63	62.18	60.88	48.41	55.79	51.84	53.41	46.93	49.96	60.54	65.53	62.27	54.23
NGRAM-3	73.71	73.15	73.43	67.82	45.97	54.80	47.21	48.47	47.83	65.92	56.63	52.99	58.69
NGRAM-4	74.63	73.07	73.85	69.32	44.32	54.07	45.73	49.26	47.43	66.21	56.66	52.40	58.45
NGRAM-5	74.20	73.34	73.81	68.97	45.17	54.59	46.50	48.61	47.53	64.34	55.74	48.95	58.64
NGRAM-6	62.44	63.67	63.05	52.57	51.97	52.27	46.67	49.70	48.14	60.17	60.21	60.19	54.49
SENTENCE-0	75.26	74.41	74.83	69.76	46.12	55.52	47.24	49.13	48.17	66.61	56.85	52.83	59.51
SENTENCE-1	75.24	73.47	74.35	69.88	44.49	54.37	46.17	50.25	48.12	65.72	56.42	51.81	58.95
SENTENCE-2	73.81	74.80	74.3	68.18	48.81	56.89	49.96	47.84	48.87	66.90	57.11	53.98	**60.02**
SENTENCE-3	72.55	73.40	72.97	65.79	49.18	56.29	50.00	48.11	49.04	66.60	57.27	55.29	59.43
SENTENCE-4	67.13	69.38	68.24	58.84	51.76	55.08	52.09	46.97	49.40	64.27	57.78	58.35	57.57

4 Results and Discussion

Table 1 contains a concise list of our experiments. The same results are displayed in the line charts in Figure 2. As discussed in Section 3, a baseline was trained using a few standard features. In each of the other experiments, the vector representations were concatenated to the baseline feature vector. There are two broad categories of experiments, NGRAM style vectors that represent each word by a 300-dimensional vector of real numbers and SENTENCE style vectors that can instead represent a sentence by a 100-dimensional vector of real values.

All models were trained using AdaBoost with Decision Trees as implemented in sk-learn [22] with the parameter n_estimators = 15. For all the other hyperparameters, the default values provided by sk-learn [22] were used, that is, learning_rate = 1.0 and the base_esitmator was the DecisionTreeClassifier.

We adopt the same metrics used in the CoNLL-2012 Shared Task, a benchmark dataset for coreference resolution. All the results in Table 1 correspond to a classifier using gold mentions, that is, the start and end of each mention was provided, but not how mentions connect together. For all experiments, gold syntactic annotations provided by the dataset were used. Note that in the table, the CoNLL Average, or the CoNLL Score, is defined as the unweighted mean of F_1 scores of the MUC, B^3 and CEAFE metrics as discussed in [8]. The shared task provides an implementation of a reference scorer that provides a reference implementation of all the metrics and all the results were produced using this scorer. The results of all the experiments with respect to the BLANC metric have also been provided for completeness, especially given that BLANC has been proposed as a potential replacement for the three other metrics [19].

In Table 1 the MUC [17] score across experiments seems consistently higher than the other scores. As discussed earlier, unlike the other metrics, it is fairly simple and considers each coreference link independently and ignores singleton mentions. Given that the model we trained also attempts to predict each link independently, it is unsurprising that this metric would score highest.

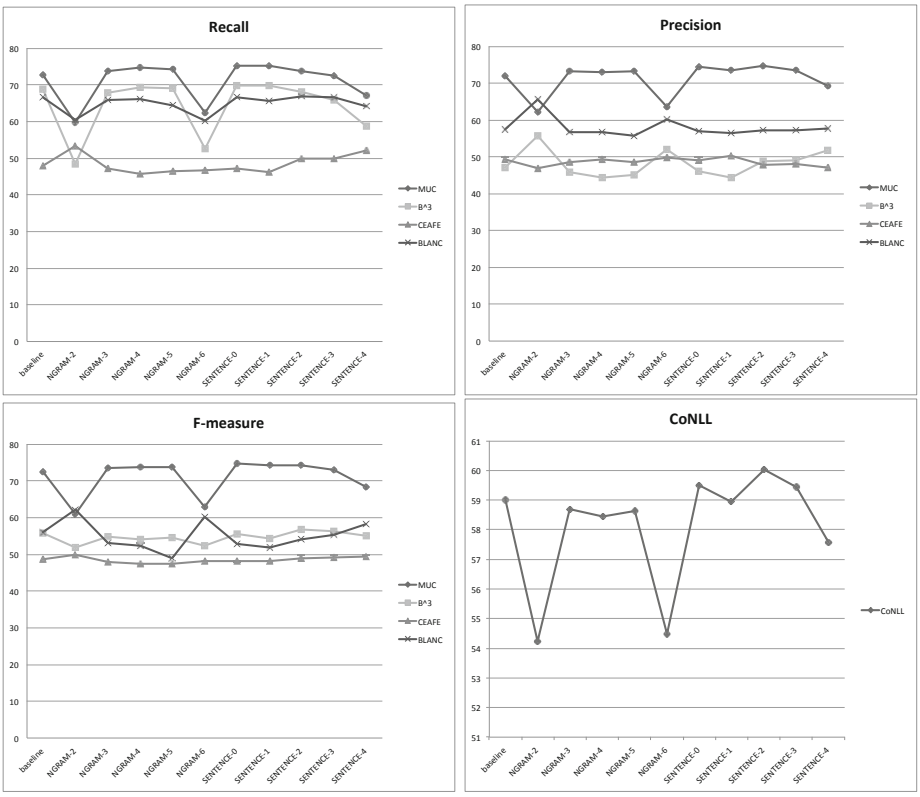


Fig. 2. A comparison of the performance of embedding features against the baseline

From Table 1, it can also be seen that the CoNLL score of the NGRAM experiments is consistently lower than the SENTENCE experiments. From Table 1, one should note that the baseline model had a CoNLL score of 59 on the test set, compared to 58.69, which was the highest score in the NGRAM experiments.

However, despite the NGRAM models being worse than the baseline, from the results in Table 1, Figure 3 shows the surprising trend that increasing the number of context words does not continue increasing the CoNLL score. This could be due to the fact that when these word embeddings are trained, they only consider the co-occurrence of the words nearby and so fail to consider the compositional meaning of the whole sentence relative to the mention. So it could be that making the window too wide introduces too much irrelevant information making it harder to discriminate between examples.

As discussed above, Durrett et. al. [13] have shown that it is possible to get good performance by using surface level syntactic features, such as a few words surrounding the mention. Intuitively, using word vectors from surrounding context words is similar to this and given that word embeddings have been shown to capture syntactic and semantic information [6], one would have expected that

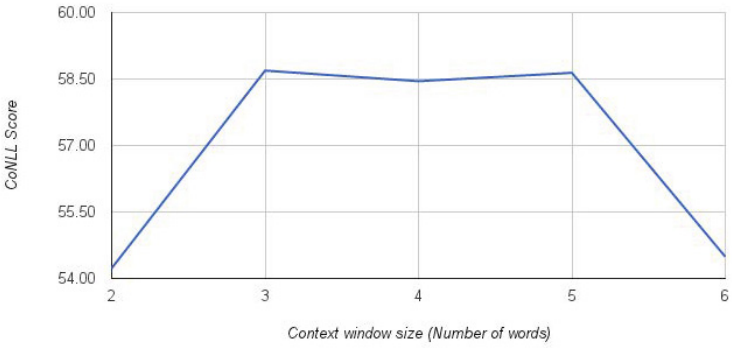


Fig. 3. NGRAM Experiments - CoNLL Score on the test set against the size of the number of context words around the mention

they would have led to good performance for small window sizes. From Table 1, a higher CoNLL score (which is the unweighted mean of MUC, B^4 and CEAFE) seems to correlate with a lower BLANC score. If we were to ignore the CoNLL score, and use BLANC instead, NGRAM-2 has a BLANC F_1 score of 62.27 compared to 56.08 on the baseline. This is a non-trivial improvement in the score and would, in some sense, be more consistent with the result in [13]. Given that the BLANC score was not used in [13], investigating the performance of the system using the BLANC score and comparing that against these results would be interesting future work.

From Table 1, the most exciting observation about the SENTENCE models is that they can outperform the baseline. This is due to that the features used in the baseline are reasonable at capturing the structure of each mention but are deficient at capturing the context around each mention. The SENTENCE models perform better because they capture this additional information about what the sentence each mention is in means, and the meaning of neighbouring sentences.

From Table 1, Figure 4 shows the CoNLL score against the context size used for the SENTENCE models. The results here are quite intuitive. If we use a context window of zero, that is, use a vector for the sentence that each mention is in, we get a CoNLL score of 59.51 compared to 59 on the baseline. Increasing the size of the window leads to a maximal score of 60.02, which is better than the baseline, considering that this metric is the average of three other metrics. As we increase the size of the window to, for example 4 or more surrounding sentences, the performance drops quite rapidly. This is intuitive because the meaning of a pair of sentences more than 4 sentences apart may be quite different and may provide irrelevant information to the classifier.

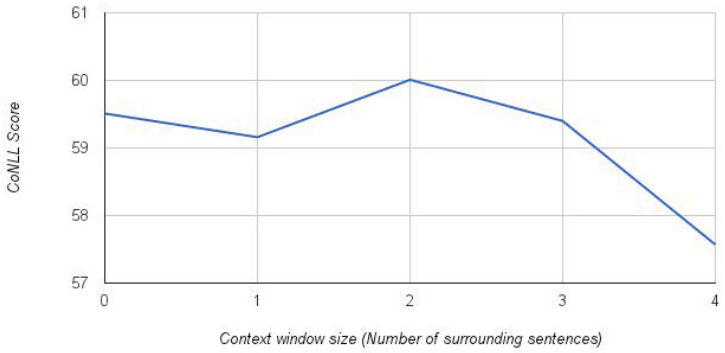


Fig. 4. SENTENCE Experiments - CoNLL Score on the test set against the size of the number of context sentences around the mention

5 Conclusion and Future Work

To circumvent the challenges associated with the scarcity of large labelled datasets for coreference resolution, the key contribution of this paper was to demonstrate the validity of the concept, of using neural embeddings in the context of a coreference resolver by using these neural emebddings as features. By constraining the complexity of the resolution strategy and the model, our experiments were controlled to ensure all changes in performance were due to the merit of the features used. As a result, we discovered that word embeddings do not contribute to the performance of the system. However, sentence embeddings, on the other hand, contribute far better than word embeddings.

For these investigations, we restricted ourselves to a simple model and a simple resolution strategy to determine whether these vector representations are useful, and as Section 4 shows, they do indeed improve performance against a suitable baseline. Therefore, the next step is to train a state of the art baseline and to try to improve its performance using these vector representations as features. Specifically, we propose that the ranking model used in [13] should be used together with embedding vector features. It would then be instructive to perform an ablative analysis [23] on the non-embedding vector features to determine which features are most discriminative when used together with the embedding vectors. This would provide insight into the kinds of information captured by these embedding vectors.

At the moment, we do not concretely understand the kind of information captured by each dimension of the embedding vector. As discussed in Section 2, others have shown that together the vector captures syntactic and semantic information, but we do not know concretely know how this information is captured. For example, is one dimension of the vector representation dedicated to

capturing Part-of-Speech information or Named Entity information? This would be another interesting line of future work.

At the moment, we are just selecting the neighbouring words for the NGRAM model, which may not always be the best choice and may introduce noise. As an avenue for future work, we propose using the dependency parse tree of the sentence that the mention is in, to find the best words to represent the mention. A dependency parse tree captures relationships between words and how they are modified by other words. For example, for each mention, find the the highest word W in the the tree from each mention. For this mention, select the following three vectors: the word vector of the parent of W in the tree, the word vector for W and the summed vectors of all of Ws children.

Acknowledgement. This work is partially supported by the Australian Research Council Linkage Grant LP110100050.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
2. Socher, R., Huang, E.H., Pennin, J., Manning, C.D., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Advances in Neural Information Processing Systems, pp. 801–809 (2011)
3. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)
4. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)
5. Socher, R., Lin, C.C., Ng, A.Y., Manning, C.D.: Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In: Proceedings of the 26th International Conference on Machine Learning (ICML) (2011)
6. Mikolov, T., Yih, W.-T., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL, pp. 746–751. Citeseer (2013)
7. Turian, J., Ratinov, L., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394. Association for Computational Linguistics (2010)
8. Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y.: Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In: Joint Conference on EMNLP and CoNLL-Shared Task, pp. 1–40. Association for Computational Linguistics (2012)
9. Björkelund, A., Farkas, R.: Data-driven multilingual coreference resolution using resolver stacking. In: Joint Conference on EMNLP and CoNLL-Shared Task, pp. 49–55. Association for Computational Linguistics (2012)
10. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics 27(4), 521–544 (2001)

11. Poon, H., Domingos, P.: Joint unsupervised coreference resolution with markov logic. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 650–659. Association for Computational Linguistics (2008)
12. Haghighi, A., Klein, D.: Unsupervised coreference resolution in a nonparametric bayesian model. In: Annual meeting-Association for Computational Linguistics, vol. 45, p. 848 (2007)
13. Durrett, G., Klein, D.: Easy victories and uphill battles in coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle (2013)
14. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 104–111. Association for Computational Linguistics (2002)
15. Ng, V.: Supervised noun phrase coreference research: The first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1396–1411. Association for Computational Linguistics (2010)
16. Denis, P., Baldridge, J.: Specialized models and ranking for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 660–669. Association for Computational Linguistics (2008)
17. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the 6th Conference on Message Understanding, pp. 45–52. Association for Computational Linguistics (1995)
18. Luo, X.: On coreference resolution performance metrics. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 25–32. Association for Computational Linguistics (2005)
19. Recasens, M., Hovy, E.: Blanc: Implementing the rand index for coreference evaluation. Natural Language Engineering 17(04), 485–510 (2011)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
21. Recasens, M., Hovy, E.: A deeper look into features for coreference resolution. In: Lalitha Devi, S., Branco, A., Mitkov, R. (eds.) DAARC 2009. LNCS, vol. 5847, pp. 29–42. Springer, Heidelberg (2009)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
23. Ng, A.: Advice for applying machine learning. CS229 Class Notes (2009)

Feature Selection in Anaphora Resolution for Bengali: A Multiobjective Approach

Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha

Department of Computer Science and Engineering,
Indian Institute of Technology, Patna, India
{utpal.sikdar,asif,sriparna}@iitp.ac.in

Abstract. In this paper we propose a feature selection technique for anaphora resolution for a resource-poor language like Bengali. The technique is grounded on the principle of differential evolution (DE) based multiobjective optimization (MOO). For this we explore adapting BART, a state-of-the-art anaphora resolution system, which is originally designed for English. There does not exist any globally accepted metric for measuring the performance of anaphora resolution, and each of MUC, B^3, CEAF, BLANC exhibits significantly different behaviours. System optimized with respect to one metric often tend to perform poorly with respect to the others, and therefore comparing the performance between the different systems becomes quite difficult. In our work we determine the most relevant set of features that best optimize all the metrics. Evaluation results yield the overall average F-measure values of 66.70%, 59.70%, 51.56%, 33.08%, 72.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively.

1 Introduction

The task of anaphora or coreference resolution refers to the task of identifying mentions (basically noun phrases) that denote the same real world objects, or entities. Many crucial applications involving Natural Language Processing (NLP), for example, Information extraction, question-answering, machine translation, text summarization etc. require the task of coreference resolution to be performed. Most of the existing works concern with some of the languages such as English [1,2], due to the availability of different lexical resources and large corpora like as ACE [3] and OntoNotes [4]. In this work we explore how a state-of-the-art English coreference system, BART [5] can be adapted for anaphora resolution in Bengali, a resource-scare language.

India is a multilingual country with great cultural and linguistic diversities. There has not been significant number of works in anaphora resolution involving Indian languages due to the following facts: Indian languages are resource constrained, i.e. annotated corpora, morphological analyzers, part of speech (PoS) taggers, named entity (NE) taggers, parsers etc. are not readily available in the required measure. Literature shows the existence of few works [6,7,8] for anaphora resolution in the languages like Hindi and Tamil. In recent times a generic framework for anaphora resolution in Indian languahas been reported in [9]. However, based on these works it is difficult to get a comprehensive view of the research on anaphora resolution related to Indian languages

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 252–263, 2015.
DOI: 10.1007/978-3-319-18111-0_20

because each of these was developed using the self-generated datasets. Therefore, it is not fair to compare between the algorithms reported in these works.

The first benchmark setup for anaphora resolution involving Indian languages was established in ICON-2011 NLP Tools Contest on Anaphora Resolution[1]. Six teams participated in this shared task with the systems, developed based on either machine learning or rules. Out of these six, four addressed the issues of anaphora resolution in Bengali, and one each for Hindi and Tamil. Apart from these an anaphora resolution system for Bengali is reported in [10], where various models for mention detection were developed, and their impact on anaphora resolution were reported. In another work, authors [11] showed how a off-the-shelf anaphora resolution system can be effectively used for Bengali. A more recent study on anaphora resolution in Bengali can be found in [12]. In contrast to the previous works, here we develop an efficient technique for feature selection in anaphora resolution based on the concept of multiobjective optimization (MOO) that incorporates differential evolution (DE) as an underlying optimization technique. Our approach is able to determine the best optimized feature sets for the five well-known evaluation metrics of coreference resolution. The method also demonstrates how systematic feature selection can help in achieving the reasonable performance with much reduced feature sets.

For anaphora resolution, there have been not much research for explicit automatic optimization except the one proposed in [2], where no significant performance improvement was observed over the baseline that was constructed with all the available features. A systematic effort of manual feature selection on the benchmark datasets was carried out by [13], who evaluated over 600 features. The very first attempt for automatic optimization of anaphora resolution was carried out by [14]. She investigated the usability of evolutionary genetic algorithms for automatic optimization of features and parameters with respect to a machine learning algorithm. She suggested that such a technique may yield significant performance improvements on the MUC-6/7 datasets. (MUC, CEAF or BLANC).

The concept of MOO for feature selection in anaphora resolution has been addressed in [15] for the English language. A genetic algorithm (GA) based MOO technique was developed for automatic feature selection in [15], where it has been shown how the method can simultaneously optimize more than one objective function, and determine near optimal features that achieve superior performance over the baseline model, developed with all the available features. In contrast to this previous study [15], we don't make use of GA as an optimization technique, and perform feature selection for a non-English language. It is to be noted that GA and DE are two different optimization algorithms. A single objective optimization (SOO) based feature selection method for performing feature selection for Bengali is recently been reported in [12]. The work reported in our current research differs from [10,11] in the sense that this work concerns with the development of a method for automatic feature selection based on a DE based MOO technique. MOO and SOO are fundamentally two different concepts. In SOO, we focus on optimizing only one objective function. But in MOO, our aim is to simultaneously optimize more than one objective function. The output of MOO produces a set of solutions on the Pareto optimal front. Each of these solutions is equally important

[1] http://ltrc.iiit.ac.in/icon2011/contests.html

from the algorithmic points of view. Hence, one interesting aspect of our algorithm is that depending upon the need user can pick up any solution.

The main focus of this work is three-fold, *viz.* (i) building a state-of-the-art anaphora resolution system for a resource-poor language like Bengali; (ii) adapting an existing state-of-the-art English co-reference resolution system for Bengali which has completely different orthography and characteristics; and (iii) multiobjective DE based feature selection technique to optimize features with respect to the evaluation metrics such as MUC, B^3, CEAFM, CEAFE and BLANC.

2 Mention Detection

Mention detection is an important component for anaphora resolution. We develop a mention detector based on the supervised machine learning algorithm, namely Conditional Random Field (CRF)[16]. The classifier is trained with the following set of features: Local contex within the previous two and next two tokens, Prefix and suffix strings of length upto three characters of the current token, Part-of-Speech (PoS) information of the current token, Named entity (NE) information (MUC categories like person, location and organization names) of the current token, noun phrase preceeding a pronoun, morphological constructs (*lemma* and *number information*) and several binary valued features. These binary valued features check whether the token is the first word of the sentence, whether the current token is a pronoun (e.g., *jeMon*[2], *kAro*, *tAhole*, *onnyoKe* etc.) that corresponds to non-anaphoric relations, whether it denotes a defin ite or demonstrative noun. In addition we prepare a list of frequently occurring suffixes that appear with the person names (e.g., *-bAbu*, *-der*, *-dI*, *-rA* etc.) and pronouns (e.g.,*-tI*, *-ke*, *-der* etc.), and define a feature that fires if the current word contains any of these suffixes. Evaluation results of this CRF based mention detector for the test data of ICON-2011 shared task on Anaphora Resolution in Indian Languages[3] are reported in Table 1.

Table 1. Results for mention detection on test data

Document id	precision	recall	F-measure
TestDoc-1	81.32	73.70	77.32
TestDoc-2	81.61	73.76	77.49
TestDoc-3	93.67	51.99	66.87

3 Pre-processing and Features of Anaphora Resolution

In this section we describe BART architecture and the features used for anaphora resolution.

[2] Bengali glosses are written in ITRANS notation.

[3] `http://ltrc.iiit.ac.in/icon2011/contests.html`

3.1 Brief Description of BART System Architecture

We use BART [5] as our underlying platform for anaphora resolution. It provides the state-of-the-art approaches, including syntax-based and semantic features. The flexibility of BART is that its design is very modular, and this provides effective separation across several tasks, including engineering new features that exploit different sources of knowledge, and improving the way that anaphora resolution is mapped to a machine learning problem. BART has five main components: *preprocessing pipeline*, *mention factory*, *feature extraction module*, *decoder* and *encoder*.

3.2 Markable Extraction

We extract the mentions following the approach described in the previous section. Thereafter we convert the mentions to the particular format required in BART, namely MMAX2s standoff XML format.

3.3 Features for Anaphora Resolution

We view coreference resolution as a binary classification problem. Following similar proposals for English [2], we use the learning framework proposed in [1] as a baseline. Each classification instance consists of two markables, i.e. an anaphor and its potential antecedent. Instances are modelled as feature vectors and used to train a binary classifier. The classifier has to decide, given the features, whether the anaphor and the candidate are co-referent or not. Given BART's flexible architecture, we explore the contribution of some features implemented in BART for coreference resolution in Bengali. We also implement some features specific to the language concerned. Given a potential antecedent RE_i and a anaphor RE_j, we compute the following set of features. Subset of these features were implemented after being motivated from the prior works [10].

1. **String match**: The feature compares the surface forms, and takes the value true if the candidate anaphor (RE_j) and antecedent (RE_i) have the same surface string forms, otherwise false.
2. **Sentence distance**: This feature denotes the distance between the anaphor and antecedent. The value of this feature is non-negative integer that captures the distance in terms of the number of sentences between a anaphor and its antecedent. The feature takes the value of 0 if both anaphor and antecedent are in the same sentence, the value of 1 is produced if their sentence distance is 1 and so on.
3. **Markable distance**: This non-negative integer feature captures the distance in terms of the number of mentions between the two markables.
4. **First person pronoun**: This feature is defined based on the direct and indirect speech. For a given anaphor-antecedent pair (RE_j, RE_i) a feature is set to true if RE_j is a first person pronoun found within a quotation and RE_i is a mention immediately preceding it within the same quote. If RE_i is outside the quote and appears either in the same sentence or in any of the preceding three sentences and is not the first person then the corresponding feature is also set to true. The feature also behaves in a similar way if the pair (RE_j, RE_i) appears outside the quotation.

5. **Second person pronoun**: This feature is defined for the pair (RE_j, RE_i) that appears in the same quote. If RE_i is not the first person and RE_j corresponds to a second person then this feature is set to true. The feature also fires if RE_j is inside the quotation, but RE_i is outside and ends with the suffix "*ke*".
6. **Third person pronoun**: If both mentions in the pair (RE_j, RE_i) denote the third person prnouns and are outside the quotation then the feature fires.
7. **Reflexive pronoun**: For a given pair (RE_j, RE_i), this feature checks whether RE_j is a reflexive pronoun and fires accordingly. This means if any antecedent is immediately followed by a reflexive pronoun then the feature is true, otherwise false.
8. **Number agreement**: This feature checks whether the anaphor and antecedent pair agree in the number information. If both agree in the number then the feature value is set to true, otherwise false. This feature is extracted from the Indian language shallow parser[4].
9. **Semantic class feature**: If the semantic types of both RE_j and RE_i are same then the value of this feature is set to true, otherwise false. The semantic types denote the MUC named entity (NE) categories.
10. **Alias feature**: It checks whether RE_j is an alias of RE_i or not. The feature value is then set accordingly.
11. **Appositive feature**: If RE_j is in apposition to RE_i then the value of this feature is set to true, otherwise it is false.
12. **String kernel**: String kernel similarity is used to estimate the similarity between two strings based on string subsequence kernel.
13. **Mention Type**: Following [1], we have encoded mention types (*name*, *nominal* or *pronoun*) of the anaphor and the antecedent. In addition, we check whether the anaphor RE_j is a definite pronoun or demonstrative pronoun or merely a pronoun. We also check whether each of the entities in the mention pairs denotes proper name.
14. **LeftRightMatch**: If RE_j is a prefix or suffix substring of RE_i or vice versa, then the value of this feature is set to true, otherwise it is false.

3.4 Learning Algorithm

In order to learn coreference decisions, we experiment with WEKA's [17] implementation of the C4.5 decision tree learning algorithm [18], with the features mentioned above. Training instances are created following [1]. Each pair of adjacent coreferent markables denote a positive training instance. A negative instance is created with the pairs of the anaphor and with any markable occurring between the anaphor and the antecedent.

3.5 Decoding

During testing, we perform a closest first clustering of instances deemed coreferent by the classifier. Each text is processed from left to right: each markable is paired with

[4] `http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php`

any preceding markable from right to left, until a pair labelled as coreferent is output, or the beginning of the document is reached. In the this step, the coreference chains are created by best-first clustering. Each mention is compared with all of its previous mentions with a probability greater than a fixed threshold value, and is clustered with the highest probability. If none has probability greater than the threshold, the mention becomes a new cluster.

4 Multiobjective Feature Selection for Coreference Resolution

4.1 Overview of Multiobjective Differential Evolution

Differential Evolution (DE) [19] is one of the popular evolutionary optimization techniques, and it performs a parallel direct search in complex, large and multi-modal landscapes, and provides near-optimal solutions. Parameters in the search space are encoded in the form of strings called chromosomes. A set of such strings is called a population denoted by NP. Each string denotes a D-dimensional parameter vector $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]$, $i = 1, 2, \ldots, NP$. The value of D represents the number of real parameters on which optimization or fitness function depends. For multiobjective version more than one objective or fitness function are associated with each string. Each of these fitness functions denotes the goodness of the string. The algorithm generates new parameter vectors by adding the weighted difference between two population vectors to a third vector, and this operation is called mutation. The param eters of the mutated vectors are mixed with the parameters of another predetermined vector, the target vector, to yield a new vector known as the trial vector. The process of parameter mixing is often referred to as crossover. Selection operation refers to the process of selecting the effective solutions. In this process the trial vectors are merged to the current population and then ranked based on the concept of domination and non-domination. In the next generation we select NP number of chromosomes from the ranked solutions using the crowding distance sorting algorithm. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

4.2 Problem Formulation

Suppose, there are D number of available features, and these are denoted by $F_1, \ldots, F_D$. Let, $\mathcal{A} = \{F_i : i = 1; D\}$. The problem of feature selection can then be stated as follows: Determine the appropriate subset of features $\mathcal{A}^{'} \subseteq \mathcal{A}$ such that when the concerned classifier is trained using these features should have optimized some metrics. In our proposed MOO based DE setting, we optimize five objective functions, namely the F-measure values of MUC, B^3, CEAFM, CEAFE and BLANC. All these metrics represent significantly different behaviours.

4.3 Problem Representation and Population Initialization

The features are encoded as binary valued strings in the chromosomes. Length of the chromosome is set equal to the number of features. The value of 1 in the i^{th} position

of a chromosome denotes the presence of the corresponding feature, and a value of 0 indicates that the respective feature does not participate while training the classifier.

4.4 Fitness Computation

The fitness computation corresponds to determining the values of the objective functions. If there are D features available in the chromosome, the classifier is trained only with these features. The trained model is evaluated on the development set. We compute the F-measure values for all the five objective functions that represent the evaluation scorers, namely MUC, B^3, CEAFM, CEAFE and BLANC. Our goal is to maximize these objective functions.

4.5 Mutation

In multiobjective DE, for each target vector $X_{i,G}$; $i = 1, 2, 3, \ldots, NP$, a mutant vector is generated according to

$$V_{i,G+1} = x_{r1,G} + F \times (x_{r2,G} - x_{r3,G}), \tag{1}$$

where $r1$, $r2$, $r3$ are mutually different random indices and belong to $\{1, 2, \ldots, NP\}$, G is the generation number and $F > 0$. The $r1$, $r2$ and $r3$ are chosen in such a way that they are different from the running current index i, so that the value of NP is at least equal to four. The parameter F controls the amplification of differential variation $(x_{r2,G} - x_{r3,G})$. Its value should be chosen within the range of $[0, 1]$. Here we set its value to 0.5. Mutated vector is denoted by $V_{i,G+1}$. After mutation operation if it is found that the value of $V_{i,G+1}$ is greater or equal to 0.5 then the value is projceted to 1, otherwise 0. A set of such NP mutant vectors is called the mutant population.

4.6 Crossover or Recombination

Crossover or recombination represents the parameter mixing of the target vector $X_{i,G}$ and mutant vector $V_{i,G+1}$. Exchange of information is performed in order to generate a better offspring that represents a promising solution. Diversity of the mutant vector can, thus, be increased. In order to perform this operation, a trial vector is formed as follows:

$$U_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \ldots, u_{D,i,G+1}) \tag{2}$$

where

$$u_{j,i,G+1} = v_{j,i,G+1} \text{ if } (r_j \leq CR) \text{ or } j = i_r \tag{3}$$

$$= x_{j,i,G} \text{ if } (r_j > CR) \text{ and } j \neq i_r \tag{4}$$

for $j = 1, 2, \ldots, D$,

In Equation 3, r_j is an uniform random number of the jth evaluation which belongs to $[0, 1]$. User should determine the value of the crossover constant CR that belongs to $[0, 1]$. Here we set its value to 0.5. An index, i_r that belongs to $\{1, 2, \ldots, D\}$, is chosen in such a way that it ensures that the parameters of $U_{i,G+1}$ gets at least one parameter from $V_{i,G+1}$. At the end of this process we obtain the trial population.

4.7 Selection

To select the best NP solutions for the next generation $G + 1$, trial population is merged to the current population, and this yields $2 \times NP$ chromosomes. These solutions are sorted based on the concept of domination and non-domination relations in the objective function space. As an example, the dominated and non-dominated relations are shown in Figure 1. In this figure non-dominated solutions (i.e. ranked solutions) are represented in the Pareto-optimal surface. Thereafter these ranked solutions are added to the population in the next generation until the number of solutions becomes less than or equal to NP. If the number of solutions exceeds NP, then crowding distance sorting algorithm is applied. This algorithm chooses the solutions starting from the beginning of the sorted rank solutions and keeps on including until it becomes equal to NP. This process ultimately determines the best NP chromosomes to be included in the next population.

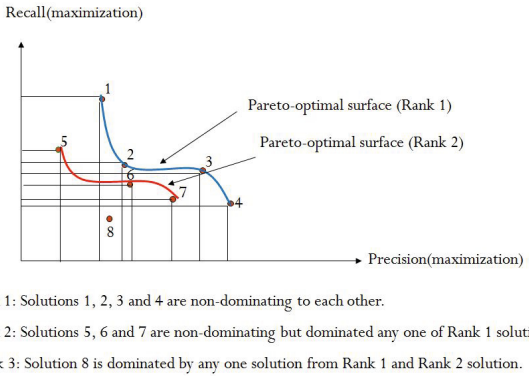


Fig. 1. Representation of dominated and non-dominated solutions

4.8 Termination Condition

The processes of mutation, crossover (or, recombination), fitness computation and selection are executed for a G_{Max} number of generations. Finally we obtain a set of non-dominated solutions on the final Pareto optimal front. Each of these solutions represents a set of (near)-optimal feature combinations.

4.9 Selecting the Best Solution

The MOO based feature selection yields a set of solutions on the Pareto optimal front. None of these solutions is better compared to the others, and therefore all are equally important from the algorithmic point of view. However we may have to select one solution at the end. Here we determine the final solution based on the F-measure value of the individual scores. We consider the top-ranked four solutions. For each of the five

objective functions, namely MUC, B^3, CEAFE, CEAFM and BLANC we select the particular solution that yields the highest F-measure value (for the respective metric) among the four solutions.

5 Experiments and Discussions

Our experiments are based on the datasets provided in the ICON NLP Tools Contest on Anaphora Resolution[5]. For training and development datasets, annotations were provided by the organizers. But no annotation was provided for the test data. In line with the annotations of training and development datasets, we manually annotated the test dataset. Also we re-annotated all these three datasets to prefer longer coreference chains. The statistics of the datasets in terms of number of the sentences and number of tokens present in each set are provided in Table 2. The datasets are of mixed domains, covering *tourism*, *short story*, *news article* and *sports*.

Table 2. Statistics of the datasets

Dataset	#sentences	#tokens
Training	881	10,504
Development	598	5,785
Test	572	6,985

Table 3. Results of baseline and manual feature selection models

Scorers	**Manual Feature Selection**			**Baseline**		
	recall	**precision**	**F-measure**	**recall**	**precision**	**F-measure**
MUC	57.80	79.00	66.70	38.80	67.40	49.30
BCUB	51.02	71.27	59.47	27.09	72.95	39.51
CEAFM	49.83	49.83	49.83	31.27	31.27	31.27
CEAFE	48.88	23.58	31.81	48.24	16.8	24.92
BLANC	70.66	70.99	70.82	54.98	63.19	56.77

In order to compare with our proposed method we construct a baseline model using a loose re-implementation of a subset of features defined in [1]. These include *number agreement, alias, string matching, semantic class agreement, sentence distance and appositive* (c.f. Section 3.3). Results of this baseline model are shown in Table 3 that yields the F-measure values of 49.30%, 39.51%, 31.27%, 24.92% and 56.77% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. Thereafter we train the classifier with all the features as mentioned in Section 3.3. Results of this model are shown in Table 3 that shows the F-measure values of 66.70%, 59.47%, 49.83%, 31.81% and 70.82% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. Comparisons show that the performance obtained in this model are significantly higher tha n the baseline model.

[5] `http://ltrc.iiit.ac.in/icon2011/contests.html`

Thereafter we apply our proposed multiobjective DE based feature selection method for determining the most relevant set of features for anaphora resolution. We optimize our algorithm based on the experiments that we perfomed on the developemnt data, and finally the best configuration is used for blind evaluation on the test data. The parameters of DE are set as follows: population size (NP) = 45, number of generations (G_Max) = 100, CR (probability of crossover) = 0.5 and F (mutation factor) = 0.5. The algorithm generates a set of solutions on the Pareto optimal front, and none these is strictly better than the others. We consider the solutions of the first rank, and finally select the best one following the technique as described in Section 4.9. It is to be noted that we select the optimized features from the four solutions of the first rank. The features, thus, selected are shown in Table 4. Detailed evaluation results when the classifier is trained wi th these four feature sets are presented in Table 5. The best performance achieved for each of these scorers correspond to 66.70%, 59.47%, 51.56%, 33.08% and 72.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. We observe performance improvements over the model developed with manual feature selection for all the metrics *except* MUC. The performance with respect to the B^3 metric does not improve, however, it is to be noted that we obtain the similar accuracy with a reduced feature set. This shows the effectiveness of the proposed feature selection technique. We also carried out experiments with the gold mentions, and this showed the F-measure values of 75.71%, 62.38%, 57.52%, 42.31% and 73.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively.

Table 4. Optimized set of features. Here, the following abbreviations are used: 'SM': String match, 'SD': Sentence distance, 'MD': Markable distance, 'FPP': First person pronoun, 'SPP': Second person pronoun, 'TPP': Third person pronoun, 'RP': Reflexive pronoun, 'NA': Number agreement , 'SCF': Semantic class feature, 'AF': Alias feature, 'MT': Mention type, 'APF': Appositive feature, 'SK': String kernel, 'MT': Mention type,'LRM': LeftRightMatch, '$Rank_{1_{Soln}}$': Solutions of rank one, 'X': Denotes the presence of the corresponding feature.

$Rank_{1_{Soln}}$	LRM	SK	NA	FPP	SPP	TPP	RP	AF	SM	SCF	MT	APF	SD	MD
$Rank_{1.1}$		X		X	X	X	X	X	X	X	X		X	X
$Rank_{1.2}$	X	X	X	X	X	X	X	X		X	X			
$Rank_{1.3}$	X	X		X	X	X	X			X	X		X	X
$Rank_{1.4}$	X	X	X	X	X	X	X		X	X	X		X	X

Our statistical significance tests using ANOVA [20] show that the performance gains in our proposed model are actually significant. In order to perform this analysis we executed our algorithm three times. Comparisons with the works reported in the ICON-2011 shared tasks show that the performance achieved in our proposed model is better compared to the others for some of the metrics. In particular we obtain much higher accuracy for the MUC scorer. The performance obtained for the BLANC scorer is also at par the state-of-the-art method, and often better in few points over most of the works carried out thereafter. However, the performance for the other three scorers need further attention. The lower performance in these three metrics may be attributed to the fact that

Table 5. Optimized F-measure values for the first-ranked solutions. Here, the following abbreviations are used: 'H_{Val}': Highest F-measure values for the corresponding scorer.

$Rank_{1_{Soln}}$	MUC	BCUB	CEAFM	CEAFE	BLANC
$Rank_{1.1}$	65.90	59.10	**51.56**	31.89	**72.75**
$Rank_{1.2}$	**66.70**	**59.70**	49.83	31.81	70.82
$Rank_{1.3}$	66.06	58.32	50.90	**33.08**	71.37
$Rank_{1.4}$	65.96	58.51	50.84	33.04	71.38
H_{Val}	**66.70**	**59.70**	**51.56**	**33.08**	**72.75**

we re-annotated the training, development and test datasets to include the longer coreference chains. For example, the coreference pairs like (*SachIn*, *Se*), (*SachIn*, *tAr*) are merged into a single coreference chain like (*SachIn*, *Se*, *tAr*). In contrast in the original datasets of ICON-11 shared task these were treated as two separate instances. This is one of the possible explanations why the link-based metric(s) such as MUC exhibits better performance and the others suffer. The method proposed in [11] is developed based on the benchmark setup of ICON-2011. They developed three models and obtained the average F-measure values of 66.6%, 68.9% and 77.1% in these three systems, respectively. However it is to be noted that along with the ICON-11 datasets, they also used additional four documents that contain 4,923 tokens. Hence, the performance reported here can't be directly compared with the method proposed in [11]. The method proposed in [12] deals with a SOO based feature selection. As we have already mentioned, here we present a MOO based feature selection technique, which is a conceptually different from SOO. The performance figures obtained in the MOO based approach are higher compared to SOO, and these are achieved even with a set of relatively l ess number of features.

6 Conclusion

In this paper we propose a multiobjective DE based feature selection technique for anaphora resolution. The proposed model is evaluated for a resource-poor language like Bengali. We adapted BART, a state-of-the-art coreference resolution model originally developed for English for the task. Our feature selection model was developed by simultaneously optimizing five evaluation metrics, namely MUC, B^3, CEAFM, CEAFE and BLANC. We conducted our experiments on a benchmark dataset that was created as part of a shared task. Our proposed multiobjective DE based method attains significant performance gains over the baseline model and the model developed with all the available features. Comparisons show that our system achieves encouraging performance with respect to the available systems. In the current setting we used only decision tree as the machine learning algorithm. Experiments with other machine learning algorithms such as maximum entropy and support vector machine will be the another direction for future work. We would also like to concentrate on porting the systems to other Indian languages, e.g. Hindi and Telugu, and domains(e.g. biomedical texts).

References

1. Soon, W.M., Chung, D., Lim, D.C.Y., Lim, Y., Ng, H.T.: A machine learning approach to coreference resolution of noun phrases (2001)
2. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 104–111 (2002)
3. Walker, C., Strassel, S., Medero, J., Maeda, K.: Ace 2005 multilingual training corpus: Ldc2006t06 philadelphia penn.: Linguistic data consortium (2006)
4. Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., Houston, A.: Ontonotes release 2.0:ldc2008t04 philadelphia penn.: Linguistic data consortium (2008)
5. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: Bart: A modular toolkit for coreference resolution. In: HLT-Demonstrations 2008 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, pp. 9–12 (2008)
6. Sobha, L., Patnaik, B.N.: Vasisth: An anaphora resolution system for indian languages. In: Proceedings Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications (ACIDCA), Monastir, Tunisia (2000)
7. Agarwal, S., Srivastava, M., Agarwal, P., Sanyal, R.: Anaphora resolution in hindi documents. In: Proceedings of Natural Language Processing and Knowledge Engineering (IEEE NLP-KE), Beijing, China (2007)
8. Uppalapu, B., Sharma, D.: Pronoun resolution for hindi. In: Proceedings of DAARC (2009)
9. Devi, S.L., Ram, V.S., Rao, P.R.: A generic anaphora resolution engine for indian languages. In: Proceedings of COLING 2014, pp. 1824–1833 (2014)
10. Sikdar, U., Ekbal, A., Saha, S., Uryupina, O., Poesio, M.: Adapting a state-of-the-art anaphora resolution system for resource-poor language. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 815–821. Asian Federation of Natural Language Processing (2013)
11. Senapati, A., Garain, U.: Guitar-based pronominal anaphora resolution in bengali. In: Proceedings of ACL, Sofia, Bulgaria (2013)
12. Sikdar, U.K., Ekbal, A., Saha, S., Uryupina, O., Poesio, M.: Differential evolution-based feature selection technique for anaphora resolution. Soft Computing, 1–13 (2014)
13. Uryupina, O.: Knowledge Acquisition for Coreference Resolution. PhD thesis, University of the Saarland (2007)
14. Hoste, V.: Optimization Issues in Machine Learning of Coreference Resolution. PhD thesis, Antwerp University (2005)
15. Saha, S., Ekbal, A., Uryupina, O., Poesio, M.: Single and multi-objective optimization for feature selection in anaphora resolution. In: Proceedings of the fifth International Joint Conference in Natural Langauge Processing (IJCNLP 2011), pp. 93–101 (2011)
16. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML, pp. 282–289 (2001)
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)
18. Quinlan, J.R.: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
19. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11(4), 341–359 (1997)
20. Anderson, T.W., Scolve, S.: Introduction to the Statistical Analysis of Data. Houghton Mifflin (1978)

A Language Modeling Approach for Acronym Expansion Disambiguation

Akram Gaballah Ahmed[2], Mohamed Farouk Abdel Hady[1], Emad Nabil[2], and Amr Badr[2]

[1] Microsoft, Redmond WA, USA
Mohamed.Abdel-Hady@microsoft.com
[2] Faculty of Computers and Information, Cairo University, Cairo, Egypt
a-akgaba@microsoft.com, {e.nabil,amr.badr}@fci-cu.edu.eg

Abstract. Nonstandard words such as proper nouns, abbreviations, and acronyms are a major obstacle in natural language text processing and information retrieval. Acronyms, in particular, are difficult to read and process because they are often domain-specific with high degree of polysemy. In this paper, we propose a language modeling approach for the automatic disambiguation of acronym senses using context information. First, a dictionary of all possible expansions of acronyms is generated automatically. The dictionary is used to search for all possible expansions or senses to expand a given acronym. The extracted dictionary consists of about 17 thousands acronym-expansion pairs defining 1,829 expansions from different fields where the average number of expansions per acronym was 9.47. Training data is automatically collected from downloaded documents identified from the results of search engine queries. The collected data is used to build a unigram language model that models the context of each candidate expansion. At the in-context expansion prediction phase, the relevance of acronym expansion candidates is calculated based on the similarity between the context of each specific acronym occurrence and the language model of each candidate expansion. Unlike other work in the literature, our approach has the option to reject to expand an acronym if it is not confident on disambiguation. We have evaluated the performance of our language modeling approach and compared it with tf-idf discriminative approach.

Keywords: word sense disambiguation, information extraction, language modeling.

1 Introduction

An abbreviation is a shortened form of a word or phrase. It consists of a group of letters taken from the word or phrase. They are frequently used in modern texts of several languages, especially English. Acronyms (including initialisms), are a special form of abbreviations. It is formed from the initial letters of the important terms in a phrase, known as the acronym expansion. Ammar et al. [1] mentioned that English

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 264–278, 2015.
DOI: 10.1007/978-3-319-18111-0_21

Wikipedia articles[1] contain an average of 9.7 abbreviations per article, and more than 63% of the articles contain at least one abbreviation. At sentence level, over 27% of news articles sentences have abbreviations. Acronyms are the most dynamic part of the lexicon of any language because a new acronym or a new definition for a known acronym can be introduced at any time by a certain community in a certain domain. Often acronyms have multiple common expansions, only one of which is valid for a particular context. For instance, Wikipedia lists 23 and 46 possible definitions (expansions) for ATM and AAC respectively. AAC refers to Atlanta Athletic Club in the sentence: "The AAC has hosted many non-golf events including the first two Southeastern Conference men's basketball tournaments in 1933 and 1934" while it refers to American Aeronautical Corporation in the sentence: "With a factory already in place in Port Washington, on Long Island, the AAC sponsored the construction of a seaplane base in the town."

Although the most known writing style requires that all acronyms have to be explicitly defined at their first occurrence in any document mentioning them, naturally occurring text often uses acronyms which are assumed to be well known in the domain. This can create serious understanding difficulties for non-expert readers and for the automated natural language processing tasks such as information extraction and machine translation, as well as information retrieval.

The aim of acronym sense disambiguation (ASD) is to identify the sense (or expansion) for a given acronym form (or spelling) occurring in a certain context. It is worth mentioning that the problem gets harder as the number of possible expansions for the given acronym increases (degree of ambiguity). Although ASD has been studied for the disambiguation of acronyms in the field of aviation [2] and within the biomedical domain [3], it has so far been an open problem for general-domain text. The property of polysemy (to have multiple senses that we called degree of ambiguity) is more frequent in acronyms than regular words. Zahariev [4] states that in a 2001 version of the WWWAAS (World-Wide Web Acronym and Abbreviation Server) database containing 16,823 terms (after cleanup), only 52.03% of acronym forms have only one sense (expansion) compared to 81.72% of WordNet [5] terms with only one sense.

Given an acronym lexicon of adequate coverage including list of acronyms with their possible expansions, acronym sense disambiguation represents a special case of the more general Word Sense Disambiguation (WSD) problem, one of the most difficult and elusive open problems in Natural Language Processing. The main difficulty of WSD lies in the fluid definition of word sense and the high costs of acquiring consistent sense repositories in order to create training sets of adequate coverage, and in conducting unbiased large-scale evaluation. In this paper, a language modeling approach is proposed to surmount the general WSD difficulties, using data and resources readily available on the Internet.

The remainder of the paper is organized as follows: Section 2 provides related work in the literature; Section 3 describes the collection of data; Section 4 explain the proposed approach for acronym expansion disambiguation; Section 5 describes the experimental setup and reports on results; Section 6 concludes the paper and proposes future work.

[1] http://dumps.wikimedia.org/enwiki/20100312/

Table 1. The polysemy of ACL and ACM acronyms

Acronym	Expansion
ACL	Access Control List
ACL	Adult and Community Learning
ACL	Advanced Concepts Laboratory
ACL	Asian Champions League
ACL	Association for Computational Linguistics
ACL	Association of Christian Librarians
ACL	Atlantic Coast Line
ACM	Abstract Control Model
ACM	Air Cycle Machine
ACM	Aluminum Composite Material
ACM	Asian Civilisations Museum
ACM	Association for Computing Machinery
ACM	Audio Compression Manager
ACM	Automatic Control Module
ACM	Automobile Club de Monaco

2 Related Work

Word-sense disambiguation (WSD) is the problem of determining in which sense a word, having a number of distinct senses, is used in a given sentence. Navigli [6] presented a comprehensive survey on WSD. While the syntactic ambiguity of a given word in a context largely can be resolved in language processing through part-of-speech taggers, word-sense disambiguation remains a challenge in the field of statistical natural language processing. In general, word-sense disambiguation includes both dictionary-based approaches, in which disambiguation is carried out using information from an explicit lexicon or knowledge base (e.g., matching words in various definitions of the word being disambiguated to the text where this ambiguous word occurs [7], and context-based [3, 2, 1] approaches. In the latter, words are disambiguated by using information gained through training on some corpus or context, rather than an explicit knowledge source. Most approaches, however, have been evaluated only on a small scale of sense disambiguation, limiting the results to theory, and without real applications. In our study, we have applied context-based disambiguation on a large scale in terms of average number of expansions per acronym.

2.1 Acronym Expansions Extraction

The AFP (Acronym Finding Program) system [8] first identifies candidate acronyms, which the authors define as uppercase words of three to ten letters. It then tries to find a definition for each acronym by scanning a 2n-word window, where n is the number of letters in the acronym. The algorithm tries to match acronym letters against initial letters in the definition words. Some types of words receive special treatment: stop-words can be skipped, hyphenated words can provide letters from each of their constituent words and, finally, acronyms themselves can be part of a definition. Given these special cases, the longest common sequence between acronym letters and initial letters in definitions is computed.

Another strategy, also developed for the medical field, is from Schwartz and Hearst [9]. The emphasis is on complicated acronym-definition patterns for cases in which only a few letters match (e.g., Gen-5 Related N-acetyltransferase [GNAT]). They first identify candidate acronym-definition pairs by looking for patterns, particularly acronym (definition) and definition (acronym). They require the number of words in the definition to be at most $\min(|A| + 5, |A| \times 2)$, where $|A|$ is the number of letters in the acronym. They then count the number of overlapping letters in the acronym and its definition and compare the count to a given threshold. The first letter of the acronym must match with the first letter of a definition word. They also handle various cases where an acronym is entirely contained in a single definition word.

Jian et al. [10] investigated three information sources for extracting and ranking acronym-expansion pairs, as provided by a large-scale search engine: the crawled web documents, the search engine logs, and the search results. In addition, several resources on the web maintain up-to-date abbreviation definitions and serve them for free (e.g. The Internet Acronym Server[2], Acronym Finder[3] and Abbreviations[4]).

2.2 Context-Based Acronym Expansions Disambiguation

Several machine learning approaches were used to solve the abbreviation expansion problem. In general text, [4] used a support vector machine (SVM) classifier with a linear kernel. A model is trained for each abbreviation, with distinct expansions representing different classes. Terms occurring in the same document as the abbreviation were used as features. Training data were obtained by searching the web for PDF documents containing both an abbreviation and any of its expansions. Though effective, building SVM models for each expansion of every abbreviation was computationally intensive. SVM attempted to assign different weights to different features.

Ammar et al. [1] presented an efficient retrieval-based method for English abbreviation expansion given a context. The system was trained on 16,415 unique expansions extracted from roughly 2.9 million Wikipedia articles. The performance was evaluated on only 500 English Wikipedia articles. The context was very hand crafted because the context for a target acronym was taken as the 10 words preceding, 10 words trailing the acronym (excluding stop-words). The level of ambiguity was limited compared to our experimental setup where the average number of expansions per acronym was only 1.54 with a variance of 1.66. Since the approach didn't take into consideration the capability of rejecting the predicted expansion, when considering 204 acronyms for which no expansions were seen in training, their approach achieved 53% accuracy.

Solving this problem for the medical domain captured the interest of many researchers due to the widespread use of abbreviations in the biomedical domain. Gaudan et al. [11] and Yu et al. [3] used SVM classifiers, and Stevenson et al. [12] used a vector space model. Terada et al. [2] designed a system for abbreviation expansion detection and disambiguation in the field of aviation.

---

[2] http://acronyms.silmaril.ie/

[3] http://www.acronymfinder.com/

[4] http://www.abbreviations.com/

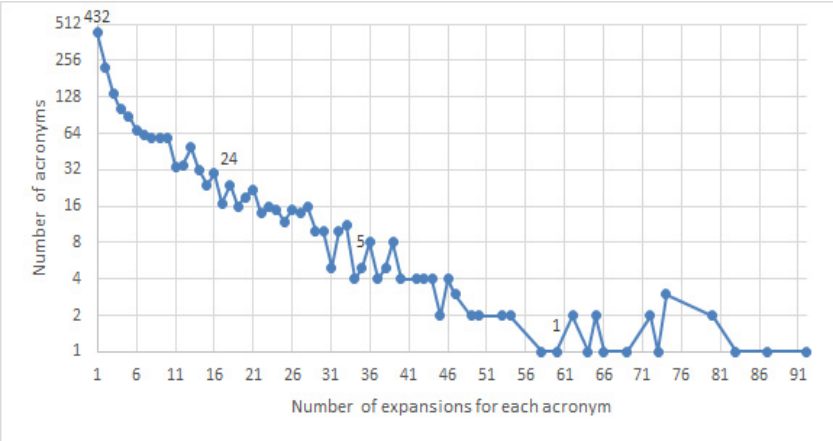


Fig. 1. Distribution of the number of acronyms paired with number of possible expansions/ senses (degree of ambiguity)

3 Data Collection

We created an acronym-expansion dictionary of 1,829 unique acronyms with 72,426 expansions, randomly selected from The Free Dictionary[5]. The World Wide Web (WWW) promises to be a good resource to automatically gather data for building domain specific language models [14]. Leveraging that, we used a popular search engine as documents retrieval system. The query alteration process made by the search engine were turned off so the retrieved results contain the search query as is. For each acronym-expansion pairs, we formed a query contracted as the concatenation of acronym and expansion. The query took the form "Expansion (Acronym)" so we can get the documents that actually contains the expansion not just sequence of words that may or may not be the required expansion. We issued the query and asked the search engine to retrieve the top 100 search results, from which we extracted the documents contents. We also retrieved the total number of search results for each query and the use of that is described at section 4.1. As a post processing step, the documents that contained more than two expansions of the same acronym were considered noisy data and were removed.

3.1 Positive and Negative Expansions

Let us suppose there are m expansions for acronym A and denote them by the set $E(\mathrm{A}) = \{E_j\}_{j=1}^{m}$. Because language modeling requires a reasonable amount of training examples, we discard an expansion if the search results are less than a threshold parameter. For each supported acronym A we define two sets ($E^{+}(A)$ and $E^{-}(A)$): If the frequency of an expansion is less than the threshold parameter, we put it into the

[5] http://www.thefreedictionary.com/

set $E^-(A)$ otherwise, we put it into the set $E^+(A)$. The acronym A is excluded from consideration if the set $E^+(A)$ is empty. In our experiment, the threshold parameter was set to 20. It is a low enough threshold to enable predictions for many expansions, yet sufficiently high to allow reasonably reliable learning. The number of expansions after this filtration significantly decreased to 17,241 expansions. The number of expansions per acronym was 9.47 on average with a standard deviation of 11.99. Figure 1 demonstrates the distribution of the acronyms per number of expansions. Then we created the vocabulary for each expansion after the stop-words[6] removal and performing Porter's stemming. Only 432 acronyms out of the 1,829 supported acronyms have a single definition which demonstrated how hard the disambiguation problem is.

4 Modeling the Language of Expansions

We build a language model from the textual representation for each acronym expansion pair. Language Models have been used in speech recognition, optical character recognition and machine translation and were originally proposed for information retrieval by Ponte and Croft [13]. We adopt the retrieval framework, treating each expansion as a pseudo-document, to build n-gram language models for all acronym candidate expansions. For each expansion, we estimate a distribution of terms associated with the expansion. We can then treat the surrounding contextual text of a target acronym as a query and estimate the probability that it was generated from a given expansion by sampling from the term distribution for that expansion.

$$P(E|T) = \frac{P(T|LM_E)P(E)}{P(T)} \quad (1)$$

where $P(T|LM_E)$ is the probability that the n-gram model of the expansion E, LM_E, generated the text $T = w_1 \ldots w_{|T|}$ and defined as

$$P(T|LM_E) = \prod_{i=1}^{|T|} P(w_i|w_1 \ldots w_{i-1}, LM_E) \approx \prod_{i=1}^{|T|} P(w_i|w_{i-(n-1)} \ldots w_{i-1}, LM_E) \quad (2)$$

In this study, we assume independence between terms, adopting the unigram model, where n=1 as defined below:

$$P(T|LM_E) = \prod_{i=1}^{|T|} P(w_i|LM_E) \quad (3)$$

We estimate the likelihood of an individual word w, given the language model of an expansion, using the maximum likelihood estimate (MLE), which maximizes the observed likelihood given the training data:

$$P(w|LM_E) = \frac{count(w, E)}{|E|} \quad (4)$$

[6] http://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html

where *count*(*w*, *E*) is the term frequency of the word *w* in expansion E, and |E| is the total number of words in the expansion training data.

In Equation 1, *P*(*T*), the prior probability of the target acronym textual context can be ignored since it is independent on all expansions and does not affect the ranking. the prior probability of expansion *P*(*E*) is discussed in the following section.

4.1 Popularity of Expansions

Not all expansions are equally likely to be the sense on some acronym. Instead, some expansions are inherently popular than others and more likely to be the candidate sense of acronym. For instance, we might want to take advantage of that fact that "University of South Alabama" is the expansion or definition of acronym USA has a very low prior, whereas the candidate definition "United States of America" has a much higher prior probability. To model this prior probability, we retrieved the estimated number of occurrences of the expansion and its acronym from the search engine results, as described in section 3. The prior, *P*(*E*), is computed as the number of documents in an expansion divided by the total number of documents of all the expansions for a given acronym. This leads to a model where expansions are ranked by $P(T|LM_E)\ P(E)$, the probability that the expansion model created the query text, a ranking approach known as query likelihood. It is noteworthy that Jian et al. [10] investigated three information sources for ranking acronym-expansion pairs that can be used to calculate their prior probabilities.

4.2 Smoothing

The use of a simple maximum likelihood estimator is problematic when dealing with sparse or missing data. In testing, when a word *w* occur in the surrounding text (context) of the target acronym that did not present in the training data of a candidate expansion *E*, it will have a "zero probability". In this case, all expansions not containing that word will have a probability of zero of "generating" the given text. To deal with this, smoothing approaches allocate a portion of the probability mass to unseen events. Naive Bayes approaches to sense disambiguation generally use the simple Laplace (add-one or add-k) smoothing. We used add-k approach to smooth the probability distribution and add *k* to each count. Since there are |*V*| words in the vocabulary, and each one got incremented, we also need to adjust the denominator to take into account the extra *k*|*V*| observations. Now the word probability becomes:

$$P(w|LM_E) = \frac{count(w,E) + k}{|E| + k\,|V|} \tag{5}$$

The result of training is a set of language models, one for each sense of a given acronym. On our experiments, we set the value of k to 0.1.

Algorithm 1. Language Modeling based Acronym Expansion Disambiguation

Require: - A: an Acronym
- T: surrounding text of the acronym (full document or the sentence containing the acronym)
- $\{LM_{E_1}, \ldots, LM_{E_m}\}$: the set of learned language models of candidate expansions of acronym A
- θ: rejection threshold

1: calculate cross entropy with background model LM_B : $H(T, LM_B)$
2: **for** $j = 1$ to m **do**
3: calculate cross entropy $H(T, LM_{E_j})$
4: **end for**
5: rank language models by cross-entropy values in ascending order
6: get the nearest model: $k1 = \arg\min_{0 \le j \le m} H(T, LM_{E_j})$
7: **if** $k_1 = 0$ (the nearest is the background model) **then**
8: **return** *reject expansion*
9: **end if**
10: get the second nearest model: $k_2 = \arg\min_{0 \le j \le m, j \neq k1} H(T, LM_{E_j})$
11: define margin(T) = $1 - \frac{H(T, LM_{E_{k1}})}{H(T, LM_{E_{k2}})}$
12: **if** margin $\le \theta$ **then**
13: **return** *reject expansion*
14: **end if**
15: **return** E_{k1} (expansion of acronym A with context T)

4.3 Disambiguation Using Cross-Entropy

The detailed description of the disambiguation phase is provided in Algorithm1. Disambiguation is performed into two steps: cross-entropy calculation and selection. Given the surrounding text T of a target acronym A that we wish to predict its expansion and the set of all possible expansions of that acronym $\{E_1, \ldots, E_m\}$. We rank the set of learned language models of candidate expansions $\{LM_{E_1}, \ldots, LM_{E_m}\}$ and the background language model LM_B by their perplexity to the surrounding text. We assume that the nearest language model, similar to the surrounding text T, is the one of the correct definition of acronym A. Selecting the expansion with the lowest perplexity is therefore equivalent to choosing the expansion with the lowest cross-entropy according to the expansion-specific language models and the background language model.

The cross-entropy of the text T with empirical n-gram probability distribution Q given a language model LM_E which has probability distribution P is:

$$H(T|LM_E) = -\sum_{i=1}^{|T|} Q(w_i) \log_2 P(w_i|LM_E)P(E) \quad (6)$$

where $Q(w_i) = \frac{1}{|T|}$ for each $1 \le i \le |T|$.

4.4 Disambiguation Rejection

We adopt two disambiguation-rejection levels to reduce the incorrect expansion prediction of acronym. This incorrect disambiguation has two main sources: either the surrounding context of the target acronym is very broad and carries insufficient non-discriminating information, or it belong to one of discarded or not supported expansions for that acronym. For many NLP application such as information extraction, information retrieval or augmenting reading, it is better to maintain the acronym without expansion than returning incorrect sense.

Table 2. the confusion matrix for the acronym CAH. (A) shows the result at is without expansion rejection, (B) shows the result after level-1 rejection, (C) is the result with full rejection levels at θ_{LM} =0.03, and (D) showes the F-1 score of each expansion at each level of disambiguation.

	E_1	E_2	E_3	E_4	R
E_1	22	0	0	0	0
E_2	0	22	0	0	0
E_3	0	2	20	0	0
E_4	0	6	0	10	0
N	0	10	10	0	0

(A)

	E_1	E_2	E_3	E_4	R
E_1	22	0	0	0	0
E_2	0	20	0	0	2
E_3	0	2	20	0	0
E_4	0	4	0	10	2
N	0	6	6	0	8

(B)

	E_1	E_2	E_3	E_4	R
E_1	20	0	0	0	2
E_2	0	20	0	0	2
E_3	0	0	20	0	2
E_4	0	2	0	10	4
N	0	2	4	0	14

(C)

F1	A	B	C
E_1	1	1	0.95
E_2	0.71	0.78	0.87
E_3	0.77	0.83	0.87
E_4	0.77	0.77	0.77
N	0	0.5	0.64

(D)

Level 1: Incorporating Background Model

In the context of modeling expansions, the background model LM_B is built using the training documents of all unknown expansions in the set E^-(A) for all the supported acronyms. The idea is that the insufficient expansion-independent context will be similar to the general background model than any expansion-dependent model. Take for example the acronym "CAH" that has four expansions at the set E^+(CAH) ={E_1, E_2, E_3, E_4}and one expansion at the set E^-(CAH) ={N}, the confusion matrices for testing this acronym with sample tests are shown at Table 2. Without applying any rejection criteria, the macro-averaged F1 score was 65%. After applying this level-1 rejection criteria, this score became 77.6%. After applying both level-1and level-2 (described next) rejection criteria, the macro-averaged F1 score increased to 82%.

Level 2: Cross-Entropy Difference

In some cases the surrounding text is confusing as it can't be used to discriminate between different candidate expansions. We accept the expansion prediction only if the margin between the lowest and the second lowest cross-entropy candidate

expansions is significantly large, based on an acronym-independent threshold (θ_{LM}). Otherwise, the expansion prediction is rejected. The objective is to maximize the margin. In a future work, we will estimate acronym-specific thresholds and study its influence on the performance. We expect that the range of margin scores differs from one acronym to another based on the similarity among its possible expansions.

Table 3. Description of the test data for both target expansions $E^{+}(A)$ and unknown expansions $E^{-}(A)$

expansions	context	#examples
Target	Documents	92,271
	Sentences	435,647
Unknown	Documents	42,293
	Sentences	134,230

5 Experiments

5.1 Setup

The retrieved documents of each expansion, as defined in the data collection section 3, is sorted by the search engine relevance then divided into two sets: the top 75% were used for training and the rest for testing. The occurrences of expansions were removed from the test data. The performance of the proposed language modeling approach is evaluated when the context is either the full document or only the sentence containing the target acronym. For this reason, Stanford CoreNLP group tool[7] is used to split documents into sentences. The total number of test examples for documents and sentences are provided in table 3.

We have divided the test data into four types: the set of documents, D+, and the set of sentences, S+, that their target expansions are in $E^{+}(A)$ of each acronym A and the set of documents, D-, and the set of sentences, S-, that their expansions are in $E^{-}(A)$ of each acronym A. The unigram background language model LM_B is built using the training documents of acronyms unknown expansions in $E^{-}(A)$. In order to evaluate the rejection ability of the approach, we run the learned models on the documents and sentences containing unknown expansions that are discarded from training (we called them negative examples).

For sake of comparison, we have implemented a nearest prototype [15] disambiguation approach based on tf-idf (term frequency inverse document frequency). At the training phase, we define the prototype of each acronym-expansion pair as the average of the tf-idf feature vectors of its training documents. At the disambiguation phase, we measure the vector cosine similarity metric between the tf-idf feature vector representing the surrounding text T of a target acronym x and the prototype of each candidate expansion y.

[7] http://nlp.stanford.edu/software/corenlp.shtml

$$\text{sim}(T, E_j) = \frac{x.y}{\|x\|\|y\|} = \frac{\sum_{i=1}^{|T|} x_i y_i}{\sqrt{\sum_{i=1}^{|T|} x_i^2}\sqrt{\sum_{i=1}^{|T|} y_i^2}} \quad (7)$$

Then we get the nearest prototype and the second nearest prototype:

$$k1 = \arg \max_{1 \le j \le m} sim(T, E_j) \quad (8)$$

$$k2 = \arg \max_{1 \le j \le m, j \ne k1} sim(T, E_j) \quad (9)$$

In order to have the ability of rejection, we first defined the margin

$$margin(T) = 1 - \frac{sim(T, E_{k2})}{sim(T, E_{k1})} \quad (10)$$

The objective is to maximize the margin. Hence, we expand the target acronym as E_{k1} if margin(T) is greater than threshold θ_{TFIDF} . Otherwise, the expansion of the acronym is rejected.

5.2 Results

The result of the proposed model and tf-idf approach over all acronym-expansion pairs showing the rate of target expansion prediction and unknown expansion rejection when the context of a target acronym is the full document (D+ and D-) or only

Table 4. Comparison of performance in the four test sets

	Correct D+	Reject D+	Reject D-	Correct S+	Reject S+	Reject S-
TF-IDF w/o Rej.	77.94	0	0	70.96	0	0
LM w/o Rej.	90.02	0	0	89.32	0	0
LM level 1 Rej.	85.67	8.12	66.44	88.09	2.67	40.26

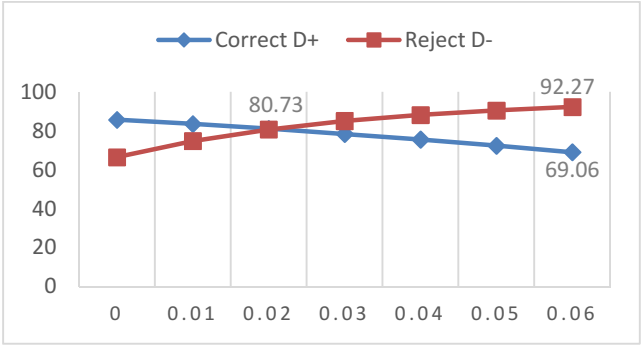


Fig. 2. Performance of the proposed approach on documents level test sets using different values for the margin θ_{LM}

the containing sentence (S+ and S-) is presented in Table 4. We reported the performance without rejection criteria and using level 1 rejection criteria. Using both levels of rejection depend on the value of θ_{LM}, Figure 2 present the results on documents level test sets using different margin values. The full results of both approaches using different margin values are reported at Table 5 and Table 6.

Table 5. The rate of target expansions prediction and unknown expansions rejection using TFIDF approach when the context level is the full document or only the containing sentence (the Correct and Reject percentage)

TFIDF	Target expansions				Unknown expansions	
context	Document		Sentence		Document	Sentence
θ_{tfidf}	Cor.	Rej.	Cor.	Rej.	Rej.	Rej.
0	77.94	0.00	70.96	0.07	0.00	0.00
0.1	77.27	2.07	70.00	3.99	6.96	9.28
0.2	76.47	4.34	68.92	7.75	13.94	17.93
0.3	75.51	6.72	67.62	11.72	21.00	26.64
0.4	74.44	9.42	66.09	15.69	28.17	35.57
0.5	73.19	12.38	64.21	20.13	35.46	44.16
0.6	71.52	15.81	62.15	24.56	43.00	52.57
0.7	69.39	19.79	59.38	29.98	51.20	61.19
0.8	66.40	24.73	55.77	35.99	59.88	70.04
0.9	60.89	32.84	49.53	44.92	70.22	79.63

Table 6. The rate of target expansions prediction and unknown expansions rejection using LM approach when the context level is the full document or only the containing sentence (the Correct and Reject percentage)

LM	Target expansions				Unknown expansions	
context	Document		Sentence		Document	Sentence
θ_{LM}	Cor.	Rej.	Cor.	Rej.	Rej.	Rej.
0	85.67	8.12	88.09	2.67	66.44	40.26
0.01	83.54	12.05	86.86	5.54	74.79	51.79
0.02	81.10	15.67	85.51	8.16	80.73	60.83
0.03	78.41	19.25	84.10	10.55	85.09	68.14
0.04	75.57	22.68	82.51	12.91	88.17	73.82
0.05	72.39	26.22	80.83	15.22	90.48	78.27
0.06	69.06	29.85	79.07	17.48	92.27	82.02
0.07	65.69	33.44	77.20	19.76	93.63	84.93
0.08	62.15	37.16	75.19	22.15	94.63	87.23
0.09	58.51	40.93	73.12	24.52	95.45	89.22
0.1	54.81	44.72	70.89	27.00	96.10	90.80

For the tf-idf Approach: we start without rejection then we increase the rejection threshold. We see decrease in the rate of incorrect target expansion prediction accompanied with another decrease in the correct prediction rate as the value of the rejection threshold increases. The approach rejects 28.17% of the documents containing unknown

expansions of known acronyms (unseen during the training) at rejection threshold θ_{TFIDF} = 0.4 and consequently decreased the incorrect disambiguation percent to 71.83%. To achieve this level, it sacrifices the correct disambiguation rate that decreases to 66.09% on sentence-level context and decreases to 74.44% on document-level context.

For the Language Modeling Approach:

- Without any rejection option, the rate of correct target expansions prediction was 90.02% (for documents) and 89.29% (for sentences), the incorrect rate was 9.98% (for documents) and 10.69% (for sentences) and consequently, all the unknown expansions documents and sentences are incorrectly associated to one of the known expansions.
- After adopting only level-1 rejection (setting θ_{LM} =0), the approach succeed to reject the prediction of 66.44% (for documents) and 40.26% (for sentences) of the unknown expansions (see the 4th row in Table 5). It is accompanied with reduction in the rate of correct target expansions prediction to 85.67% (for documents) compared to 77.94% for the tf-idf approach and 88.09% (for sentences) compared to 70.96% for the tf-idf approach and improvement in the incorrect prediction rate to 6.21% (for documents) and 9.24% (for sentences). This is thanks to the background language model LM_B that acts as the first line of rejection.
- Using both level-1 and level-2 rejection at θ_{LM} = 0.02, the rate of unknown expansions rejection is increased to 80.73% (for documents) and 60.83% (for sentences). As a consequence, the rate of correct target expansions prediction is reduced to 81.10% (for documents) and 85.51% (for sentences), the incorrect rate is reduced to 3.22% (for documents) and 6.33% (for sentences).

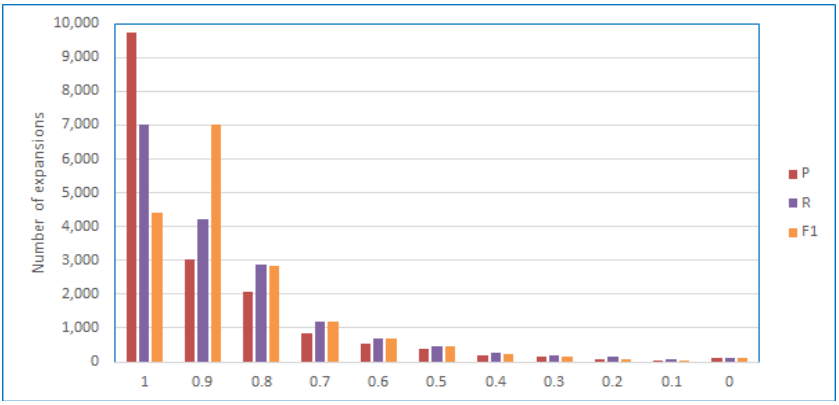


Fig. 3. Distribution of the number of expansions for different values of precision, recall and F1-measure where θ_{LM} =0.01

5.3 Discussion

We see a significant improvement in the rejection rate even with a small value of the rejection threshold. It is clear that the increase in the unknown expansions rejection rate is accompanied with a degradation in the correct target expansions prediction rate on test documents and sentences of trained expansions. The histogram in Figure 3 shows the number of expansions for different ranges of precision, recall and F1-measure where θ_{LM} =0.01. For instance, the three pillars at the third bin represent the number of expansions with precision, recall and F1 between 0.8 and 0.9. We find that 82.65% of the target expansions (14,250 out of 17,241) achieve F1 $\geq$ 0.8 while 14,866 and 14,134 expansions have precision and recall within the same range, respectively. These results shows the effectiveness of the proposed approach.

Some expansions don not perform well such as "United States of America" that has 0.86, 0.31 and 0.45 precision, recall and F1 while another expansion for the same acronym USA such as "University of South Alabama" has 0.98, 0.95 and 0.96 precision, recall and F1 . The reason is that the acronym USA has 18 possible expansions and 41 of the 59 test sentences of "United States of America" are either incorrectly assigned to one of the other 14 expansions or not expanded because the margin between its language model and the background model is too small. We discovered that some expansions are too broad and generic so that it is not practical to model their surrounding context. In a future work, we will investigate techniques to detect those expansions and discard them.

6 Conclusion

In this paper, we have presented a novel perspective of looking at the problem of context-based acronym expansion disambiguation based on probabilistic language modeling. In addition, we have presented a technique for rejecting the expansion of acronym with off-topic surrounding text. The rejection threshold depends in the margin between the two most probable candidate expansions. Our experimental results demonstrate that the performance on the four test sets was better than that obtained by the tf-idf baseline approach. Although the improvement in performance is a key point, more significant is that a different approach for disambiguation has been shown to be effective. The ability to think about retrieval in a new way can lead to insights that would be less obvious in other approaches. We have used additive smoothing in our trained models, we plan to investigate other smoothing techniques in the future. We also plan to investigate the use of clustering to reduce the number of language models.

References

1. Ammar, W., Darwish, K., El Kahki, A., Hafez, K.: ICE-TEA: In-context expansion and translation of english abbreviations. In: Gelbukh, A. (ed.) CICLing 2011, Part II. LNCS, vol. 6609, pp. 41–54. Springer, Heidelberg (2011)

2. Terada, A., Tokunaga, T., Tanaka, H.: Automatic expansion of abbreviations by using context and character. Information Processing and Management 40(1) (2004)
3. Yu, H., Kim, W., Hatzivassiloglou, V., Wilbur, J.: A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations. ACM Transactions on Information Systems 24(3) (2006)
4. Zahariev, M.: Automatic sense disambiguation for acronyms. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), pp. 124–132 (2004)
5. Fellbaum, C.: MIT Press (1998)
6. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys 41(2) (2009)
7. Klavans, J., Chodorow, M., Wachokder, N.: From dictionary to knowledge base via taxononym. In: Proceedings of the 6th Conference of the UW Contre for the New OED, pp. 41–54 (1990)
8. Taghva, K., Gilbreth, J.: Recognizing acronyms and their definitions. International Journal on Document Analysis and Recognition, 191–198 (1999)
9. Schwartz, A., Hearst, M.: A simple algorithm for identifying abbreviation definitions in biomedical texts. In: Proceedings of the Pacific Symposium on Biocomputing (PSB) (2003)
10. Jain, A., Cucerzan, S., Azzam, S.: Acronym-expansion recognition and ranking on the web. In: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2007), pp. 209–214 (2007)
11. Gaudan, S., Kirsch, H., Rebholz-Schuhmann, D.: Resolving abbreviations to their senses in medline. Bioinformatics 21(18), 3658–3664 (2005)
12. Stevenson, M., Guo, Y., Amri, A.A., Gaizauskas, R.: Disambiguation of biomedical abbreviations. In: BioNLP Workshop, HLT 2009 (2009)
13. Ponte, J., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), pp. 275–281 (1998)
14. Mahajan, M., Beeferman, D., Huang, X.D.: Improved topic-dependent language modeling using information retrieval techniques. In: Proceedings of ICASSP (1999)
15. Kuncheva, L., Bezdek, J.: An integrated framework for generalized nearest prototype classifier design. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 6(5), 437–457 (1998)

Web Person Disambiguation Using Hierarchical Co-reference Model

Jian Xu, Qin Lu, Minglei Li, and Wenjie Li

The Hong Kong Polytechnic University,
Department of Computing, Hung Hom, Hong Kong
{csjxu,csluqin,csmli,cswjli}@comp.polyu.edu.hk

Abstract. As one of the entity disambiguation tasks, Web Person Disambiguation (WPD) identifies different persons with the same name by grouping search results for different persons into different clusters. Most of current research works use clustering methods to conduct WPD. These approaches require the tuning of thresholds that are biased towards training data and may not work well for different datasets. In this paper, we propose a novel approach by using pairwise co-reference modeling for WPD without the need to do threshold tuning. Because person names are named entities, disambiguation of person names can use semantic measures using the so called co-reference resolution criterion across different documents. The algorithm first forms a forest with person names as observable leaf nodes. It then stochastically tries to form an entity hierarchy by merging names into a sub-tree as a latent entity group if they have co-referential relationship across documents. As the joining/partition of nodes is based on co-reference-based comparative values, our method is independent of training data, and thus parameter tuning is not required. Experiments show that this semantic based method has achieved comparable performance with the top two state-of-the-art systems without using any training data. The stochastic approach also makes our algorithm to exhibit near linear processing time much more efficient than HAC based clustering method. Because our model allows a small number of upper-level entity nodes to summarize a large number of name mentions, the model has much higher semantic representation power and it is much more scalable over large collections of name mentions compared to HAC based algorithms.

1 Introduction

Searching information about persons on the Internet is one of the most common activities for Internet users. An estimated 30% web queries contain person names [1-3]. In a Google 2013 survey of search trends, 3 out of the top 10 searches are person names. Statistics from the 1990 U.S. Census bureau show that out of 100 million population, only about 90,000 different names are used [1], which means that on average a name is shared by over a thousand people. Since a literal name as a lexical sequence can appear over the Internet in large quantity, many web pages containing the same name may not refer to the same person. For example, the name

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 279–291, 2015.
DOI: 10.1007/978-3-319-18111-0_22

mention "*Michael Jordan*" may refer to the American basketball player or the computer science professor at UC Berkeley. Also, celebrity or popular names tend to monopolize search results as most current search engines tend to return the most highly cited persons. Hence, users are inundated by a vast amount of information and need to add more query items to locate the target web pages. Web Person Disambiguation(WPD) aims to solve this problem. Its objective is that for a given search name, return as many different entities associated with the name as possible.

To conduct WPD, clustering methods are used to resolve entities mentions into entities. Thus, in technical terms, WPD is basically an entity resolution task. Since all mentions are contained in web documents, they are also referred to as cross-document co-reference resolution tasks. The traditional K-means clustering does not work, as there is no sure way to determine the number of clusters. Therefore, most researchers use the hierarchical agglomerative clustering (HAC) method because they can manually tune a threshold based on similarity measures to determine the number of clusters [4, 6, 7, 12, 25, 26]. Others have also tried to learn the threshold automatically from training data [8]. However, thresholds obtained either automatically or manually from training data may not be applicable to testing data. Categories of professions from some knowledgebase are often used in named entity disambiguation tasks [11]. But, it can also introduce noise because many people have more than one profession at different time throughout their career life.

In this paper, we report a novel semantic-based approach which extends the use of pairwise co-reference modeling to disambiguate persons in a hierarchical structure. As Web persons are named entities, we can make use of their co-reference information across documents to group different documents into one corresponding entity. The algorithm first form a forest with documents containing name mentions as observable leaf nodes. It then attempts to stochastically form an entity hierarchy by merging names into a sub-tree as a latent entity group if they have co-referential relationship across documents. As the joining/partition of nodes is based on co-reference-based comparative values, parameter tuning is not required. Person names are disambiguated by deciding whether two entity nodes are co-referential or not in a factor graph. Experiments on the publically available WePS2 dataset show that the proposed method outperforms all the HAC systems that learn the threshold automatically. It also achieves a comparable performance with the top two systems which manually tune the number of clusters. More importantly, our system using stochastic approach is more scalable due to its near linear performance as indicated by the performance evaluation.

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 gives algorithm design. Section 4 is performance evaluation. Section 5 discusses this algorithm and Section 6 concludes this paper.

2 Related Works

WPD aims to group web search results into different clusters with each cluster referring to the same person [3]. In this paper, a **name mention** refers to a lexical

sequence for a named entity and an **entity** is a specific, disambiguated individual in real life. WPD is a challenging task because entity to name mapping is a many-to-many problem. For example the entity, such as *Michael Jordan*, the American basketball player, can be described by multiple name mentions (e.g., "*Michael Jeffrey Jordan*", "*MJ*", "*Jordan*", "*Air Jordan*" and "*His Airness*") and a name mention such as "*Michael Jordan*" can refer to multiple entities, such as the American basketball player or the computer science professor at UC Berkeley.

To resolve ambiguous person names, most common methods use clustering based approaches, such as K-Means Clustering [17], Quality Threshold clustering [18], and Hierarchical Agglomerative Clustering (HAC) [6, 12, 16]. Due to the limitation of K-Means method and fuzzy clustering, most researchers use the HAC method because they can manually tune a threshold based on similarity measures to determine the number of clusters. Others also tried to learn the threshold automatically from training data [9, 18]. Gong and Oard (2009) explored both local and global features in SVM to find the thresholds for HAC. Romano et al. (2009) estimated the threshold from training data using the Quality Threshold clustering algorithm. Thresholds obtained either automatically or manually from the training data might not work well to the testing data, especially if the training data and testing are from different domains. Besides clustering, classification methods such as KNN classifier [11] are used because certain specific knowledge can be incorporated. Han and Zhao (2009) made use of the professional categories extracted from Freebase. However, using professional category to determine the number of persons can also introduce noise because many people can have multiple posts or roles simultaneously and people tends to have more than one profession at different time throughout their career life.

Multi-document co-reference resolution was found useful in entity disambiguation such as the experiments on the New York Times corpus [10, 17]. It was also used to resolve author co-reference in bibliographic databases [20-24]. Most of these works group name mentions across documents within citations into different clusters using pair-wise co-reference modeling by computing similarity between pairs of mentions [10, 17, 21, 22]. Pair-wise co-reference model does not scale up to large collections of mentions because of the quadratic number of comparisons between mentions. For this reason, Singh et al. (2011) and Wick et al. (2012) presented a hierarchical co-reference model for author co-reference in bibliographic data on large scale. In author co-reference, five features associated with authors are used, including paper title, co-authors, emails, venues, and domain-specific keywords for a paper [23, 24]. However, due to the heterogeneous nature of web pages and the scaling issue for tree join/split, no attempt has been made using co-reference resolution in WPD.

3 Algorithm Design

In this work, we propose to use co-reference information for web person disambiguation. The basic idea is to make use of different features contained in web documents for merging or splitting operations in the co-reference model. The key issues are, how the co-reference models are formed for web pages, what features to use for the algorithm, and how to handle the scaling issue for tree node join/split.

3.1 Hierarchical Co-reference Model (HCM)

We propose to use the hierarchical co-reference model [20, 23] because co-references naturally form a hierarchical structure. Given a collection of name mentions extracted from document text, co-reference resolution for WPD is to group name mentions into clusters such that mentions in the same cluster are referring to the same entity. Fig. 1 shows a sample search result of the query "*John Howard*" and features that can be used to identify entities are marked by colored boxes.

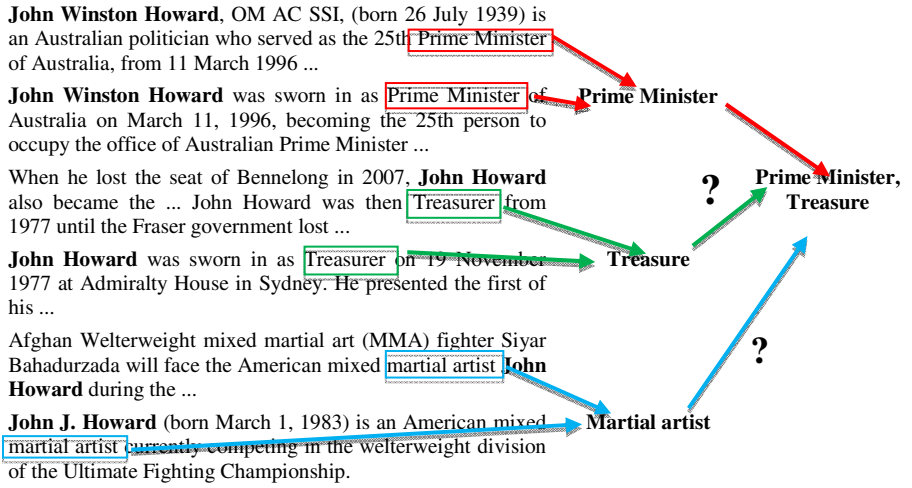


Fig. 1. Example Mentions of "John Howard" with the True Entities on the Right

For the name query "*John Howard*", the result has a list of name mentions such as "*John Winston Howard*", "*John Howard*", "*John J. Howard*". The six name mentions refer to two real world entities: a prime minister and a martial artist. Each tree should correspond to one entities, the merging of the mentions into tree structure are based on the different features describing them.

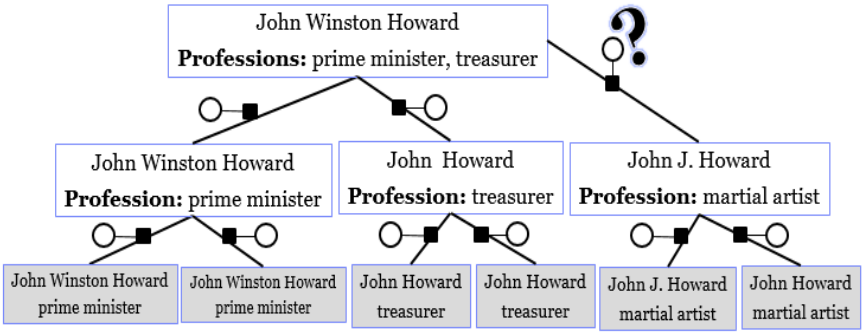


Fig. 2. Hierarchical Co-reference Model for "John Howard"

The basic idea of the hierarchical co-reference modeling algorithm is to iteratively form an entity hierarchy by merge names into a sub-tree as a latent entity group if they have co-referential relationship across documents. As illustrated in Fig. 2, name mentions for "John Howard" are observable leaf nodes in gray boxes. Decision variables in open circles indicate parent-child relationship between two entity nodes. Factor nodes in black boxes measure the compatibility between parent and child nodes. Latent entity nodes in white rectangles aggregate attributes from child nodes (sub-entity nodes or leaf nodes). The factor with a question mark decides whether the two sub-trees are co-referent or not. Using the hierarchical model, entity nodes in the tree can aggregate the features from their child nodes to form a more generalized representation, thus making it more scalable with rich feature representation. Co-reference resolution in our work is conducted between entity nodes instead of mention pairs. Thus, the number of decision variables is also much reduced.

Let e_i denotes the i^{th} entity node and m_j be the j^{th} name mention in the tree. Let y_{ij} be a binary decision variable indicating whether m_j co-refers to the parent entity e_i. Formally, the distribution of the hierarchical co-reference model is defined as,

$$p(\boldsymbol{y}, \boldsymbol{E}|\boldsymbol{m}) \propto \prod_{e_i \in E} \varphi_1(e_i)\varphi_2(e_i, e_i^p)$$

where $\boldsymbol{E}$ is the set of entity nodes. Factor φ_1 assesses the prior knowledge over the entity nodes. Factor φ_2 measures the compatibility between a child entity node and its parent node denoted by e_i^p. In our case, factors φ_2 takes the form of an exponential function: $\varphi_2 = exp\,(\boldsymbol{\theta} \cdot \emptyset(e_i, e_i^p))$, where $\emptyset(e_i, e_i^p)$ are feature functions for entity node e_i and its parent. In WPD, the feature functions can test whether two entity nodes have the same emails, organizations, or simply compute the cosine similarity between two entities using bag-of-words or other methods derived from their child nodes. The parameters $\boldsymbol{\theta}$ in φ_2 is a weight factor for these feature functions. To learn these parameters, we do not rely on training data. Instead, the Markov chain Monte Carlo (MCMC) inference algorithm is used to search for a configuration of the entity trees that has the highest probability [23]. In each step, MCMC randomly selects a pair of sub-trees as a sample with entity nodes e_i and e_j, then it proposes to either merge or split entity nodes [24]. To accept or reject these proposals, Metropolis Hastings (MH) sampler is used [23]. Based on the current co-reference configuration $\boldsymbol{y}$, a proposal function puts forward a new configuration $\boldsymbol{y}'$ by the merging or splitting operations. These proposed moves are accepted with the probability α defined as

$$\alpha(\boldsymbol{y}', \boldsymbol{y}) = min\left(1, \frac{p(\boldsymbol{y}')}{p(\boldsymbol{y})} \frac{q(\boldsymbol{y}|\boldsymbol{y}')}{q(\boldsymbol{y}'|\boldsymbol{y})}\right)$$

where q is a transition kernel. The MH sampler is a special case of the Markov chain. Since we assume that the chain is reversible in our case, thus $q(\boldsymbol{y}|\boldsymbol{y}') = q(\boldsymbol{y}'|\boldsymbol{y})$. The acceptance probability α is then reduced to,

$$\alpha(\boldsymbol{y}', \boldsymbol{y}) = min\left(1, \frac{p(\boldsymbol{y}')}{p(\boldsymbol{y})}\right)$$

which simply measures the model ratio between the current co-reference configuration $\boldsymbol{y}$ and the proposed configuration $\boldsymbol{y}'$. A sample being accepted is

called a valid sample and a rejected sample will not change the structure of the forest. Compared to the pair-wise model which moves only a single mention each time, the hierarchical model can move an entity that represents a collection of similar mentions, resulting in higher acceptance probability of samples [20].

3.2 Features

After introducing the hierarchical model, the next step for hierarchical co-reference is to select features from which the factors (compatibility functions) can be computed. In WPD, name mentions are often extracted from web text and the context surrounding a name mention is informative to identify a person. In this work, the **context** is defined by a window of fixed size around the name mention. When context words of name mentions overlap, the overlapped words are extracted only once. Within these context windows, we devise a set of semantic features in the web documents suitable for co-reference resolution as given below:

(1) Context words: Words mostly in lexical form insides the context window. In this work, only nouns, adjectives and verbs are used as they are semantically meaningful.
(2) HTML features: Title, snippet, urls, bold/italic text, and outgoing URL titles, often used in WPD clustering.
(3) Person-specific features: Full name, email, profession, dates, age, phone number and gender. Except profession, the other six features are extracted based on rules. Professions are extracted using manually crafted profession dictionary. We also use a rule-based gender extractor mostly based on patterns given in [13]. Designators for genders are *Mr.*, *Mrs.*, *Miss*, *Lady*, *Lord*, etc.
(4) Surrounding named entities and types: Persons, locations and organizations within the context of a name mention. Entity types are included as features too. The Stanford CoreNLP tool is used for entity identification.
(5) Keywords: Keywords and phrases that are informative for a person name mention. For example, keywords "tv series", "sound clip" and "imdb" can identify a person as an actor. Keyword extraction uses similar method reported in [25] in which the training data for keyword extraction use Wikipedia articles and anchor text in these articles are treated as manually assigned keywords. Keywords are then extracted using the CRFs model [15][1].
(6) Wikipedia categories: the categories in Wikipedia for article classification as additional semantic information to enrich the features of target person names. We extract Wikipedia categories for our feature data including keywords, named entities and bold text in the proximity of the name mentions. For example, the name mention "Amanda Lentz" has a neighboring named entity "North Carolina" which contains such category labels in Wikipedia as "Former British colonies", "State of Franklin", "Southern United States" and so on.
(7) Topics: Topics for each name mention. They are obtained through topic modeling method to find the topics for each name mention [5]. This is because we believe that if two name mentions refer to the same underlying entity, the context surrounding them should have similar topics.

[1] More details on the extraction of features (5) to (7) will be given in experiments.

Feature values of each type are put together using the simple bag-of-words model. The weights of the feature values (as "words" in the bag-of-words model) are determined by their frequency only. We then use six factors over these features exactly as defined by Wick et al [23, 24] given in Table 1. The *Cosine* factor measures the cosine similarity between child and parent's bag of features. *Entropy* and *Complexity* penalize an entity node's bag of features when an entity node has a large number of features. *Entity penalty* and *Sub-entity penalty* factors controls the depth of the hierarchical tree structure. *Name penalty* rewards two entities having the same middle name. In Table 1, $\boldsymbol{p}$ and $\boldsymbol{c}$ refer to parent and child bags of features and $\boldsymbol{b}$ is a bag of features for a node. w and t are weights for each factor. $f(e)$ returns 1 if the node e is the root and 0 otherwise. $g(e)$ returns 1 if the node e is neither a root nor a leaf in the tree and 0 otherwise.

Table 1. Definition for Six Types of Factors

Factors	Definition
Cosine	$w \times log(\|\boldsymbol{c}\|_1 + 2)\left(\frac{(\boldsymbol{p}-\boldsymbol{c})\cdot\boldsymbol{c}}{\|\boldsymbol{p}-\boldsymbol{c}\|_2\|\boldsymbol{c}\|_2} + t\right)$
Entropy	$-w \times \frac{H(\boldsymbol{b})}{log\|\boldsymbol{b}\|_0}$
Complexity	$-w \times \frac{\|\boldsymbol{b}\|_0}{\|\boldsymbol{b}\|_1}$
Entity penalty	$-w \times f(e)$
Sub-entity penalty	$-w \times g(e)$
Name penalty	$-min\big((w \times \|\boldsymbol{b}\|_0 - 1), -t\big)$

4 Performance Evaluation

The evaluation of our proposed hierarchical co-reference algorithm for WPD is conducted on the test data of WePS2 workshop which targets at clustering web pages of ambiguous person names [3]. WePS2 provides both training and testing data. We do not need to use the training data to tune our model. But, we use the testing data. The testing data has 30 ambiguous names with two sets of evaluation metrics: the BCubed Precision/Recall (BEP and BER in short) as an inter-cluster measure, and Purity/Inverse-Purity (IPur. in short) as an intra-cluster measure [3]. Two F-scores are also used: one giving equal weights to precision and recall (α=0.5) and another giving higher weight to recall (α=0.2). The context window size for our system is set to 55 words, directly taken from [10], which was experimentally determined to give optimal performance in the task of cross-document co-reference resolution. In this work, we use the Wikipedia dump with the timestamp: April 03, 2013 to obtain categories and keywords. The Stanford CoreNLP tool is used to preprocess these articles, including tokenization, part-of-speech tagging, and named entity recognition. To extract topics of each name mention, we run a topic modeling procedure with hyper-parameters α=0.1 and β=0.1 by 100 iterations [5]. The number of topics is set to 50 for each person name and ten topics with highest probabilities are used as features for each document. For weights w and t used in the six factors, we simply follow the configurations done by Wick et al. [24]. They are listed in Table 2 for reference.

Table 2. Weights for Factors in Hierarchical Co-reference Model

Features	Weights for Factors
keywords; emails neighbor categories	$w = 4.0, t = -0.25\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
italics; snippet; ages; entity types; dates professions; phone numbers bold text categories; keyword categories	$w = 4.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
local words topics	$w = 4.0, t = -0.25\ (cosine)$ $w = 0.75\ (entropy)$ $w = 0.25\ (complexity)$
neighboring entities urls	$w = 2.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
bold texts; title tokens url titles	$w = 3.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
middle names; gender	$w = 1.0, t = 16\ (name\ penalty)$
structural prior	$w = 4.0\ (entity\ exist\ penalty)$ $w = 0.25\ (subentity\ penalty)$

In the first set of experiments, our system, $HIER_{coref}$, is compared to three of the systems in WePS2 workshop which have tuned the threshold automatically, namely *XMEDIA_3* [18], *UMD_4* [9], and *CSSIANED_4* [11]. Results are given in Tables 3.

Table 3. Comparison in BCubed & Purity to Systems that Automatically Tune Threshold

Systems	Macro-averaged Scores (%)							
	BCubed F-scores				Purity F-scores			
	α= 0.5	α= 0.2	BEP	BER	α= 0.5	α= 0.2	Purity	IPur.
XMEDIA_3	72	68	82	66	80	76	91	73
UMD_4	70	63	**94**	60	81	76	**95**	72
CASIANED_4	63	68	65	75	73	77	72	83
$\boldsymbol{HIER_{coref}}$	**81**	**78**	88	**78**	**86**	**84**	91	**83**

Table 3 shows that our system obtains the highest F-scores in both BCubed and Purity scores. When comparing to *XMEDIA_3*, $HIER_{coref}$ obtains 9% and 10% increase in BCubed F0.5 and F0.2; 5% and 8% increase in Purity F0.5 and F0.2. *XMEDIA_3* applied the Quality Threshold clustering algorithm and the threshold for merging two documents is learned using the SVM regression model. This means that *XMEDIA_3* needs training data for their learning model. Similar to *XMEDIA_3*, *UMD_4* learns the threshold for its HAC model by SVMs. *CSSIANED_4* disambiguates person names by categorizing them into a profession taxonomy from Freebase using the KNN classifier. It is worth noting that *UMD_4*, which uses HAC has the best performance in purity and BEP both are related to precision.

The next set of experiments compares $HIER_{coref}$ to the top four systems in the WePS2 workshop with the top 3 systems using the HAC method and the top 4 system using Quality Threshold clustering. The four systems are *PolyUHK* [6, 7[2]], *UVA_1* [4], *ITC_UT_1* [12], *XMEDIA_3* [18], and *UMD_4* [9]. The BCubed and Purity scores are given in Table 4. Table 4 shows that our system achieves a comparable performance with the top two systems *PolyUHK* and *UVA_1* in both BCubed and Purity F-scores. Note that the top two systems are all using the HAC algorithm which indicates that HAC is quite effective for WPD. However, HAC requires threshold tuning to find the number of clusters. It is also worth noting that our $HIER_{coref}$ system obtains good BEP and Purity scores. This implies that our system can find the clustering solutions with less intersection between clusters (high BEP score) and less noise within each cluster (high Purity score).

Table 4. Comparison in Bcubed & Purity to Top Four Systems

Systems	**Macro-averaged Scores (%)**							
	BCubed F-scores				Purity F-scores			
	α= 0.5	α= 0.2	BEP	BER	α= 0.5	α= 0.2	Purity	IPur.
PolyUHK	**82**	**80**	87	79	**88**	**87**	**91**	86
UVA_1	81	**80**	85	**80**	87	**87**	89	**87**
ITC_UT_1	81	76	**93**	73	87	83	95	81
XMEDIA_3	72	68	82	66	81	76	95	72
$\boldsymbol{HIER_{coref}}$	81	78	88	78	86	84	**91**	83

To further examine the sensitivity of the HAC algorithm to threshold values, the 3rd set of experiments is conducted on HAC with different threshold values. In this experiment, we used token features including nouns, verbs and adjectives with the thresholds: 0.1, used by the top system *PolyUHK*, and 0.135, manually tuned by our system. BCubed and Purity scores are given in Table 5.

Table 5. Token-based HAC algorithm using BCubed & Purity Scores

Systems	**Macro-averaged Scores (%)**							
	BCubed F-scores				Purity F-scores			
	α= 0.5	α= 0.2	BEP	BER	α= 0.5	α= 0.2	Purity	IPur.
$HAC_{0.1}$	70	**79**	62	**90**	79	**86**	71	**94**
$HAC_{0.135}$	**75**	76	**79**	78	**83**	84	**84**	86

Obviously, tiny variation in the thresholds can change the performance of the WPD system significantly. In other words, the number of clusters returned by the HAC algorithm is quite sensitive to thresholds especially when the number of clusters per person name has a large variation among the 30 persons (from 1 up to 56 different persons sharing the same name) [3].

[2] [7] is a journal version of the work of [6] with much more details.

5 Complexity Analysis of Hierarchical Co-reference Model

In the 4th set of experiment, we try to evaluate the complexity of HCM with respect to document size. Since the HCM method is a randomized stochastic algorithm for sample selection, it is difficult to evaluate its complexity directly. One practical way to estimate its complexity is by running experiments using different number documents. Out of the 30 names in WePS2 data, we use 23 names whose corresponding document size is larger than 100. Because HCM is a randomized algorithm for sample selection, iteration should stop only after the algorithm can reach a set of stable valid samples. Obviously, reaching stable valid sample size depends on the number of files to be clustered. Therefore, we need to obtain the appropriate sample size for different sizes of document sets. Fig. 3 shows that the performance of the algorithm using the 23 names indeed stabilizes after certain number of iterations (indicated by time here) for document size N=25. Once the performance is stabilized, there is no point to run the sampling algorithm anymore. The same experiments are done for N=*50, 75, 100*. All of them exhibit similar behavior. That is, they all stabilize after certain iterations.

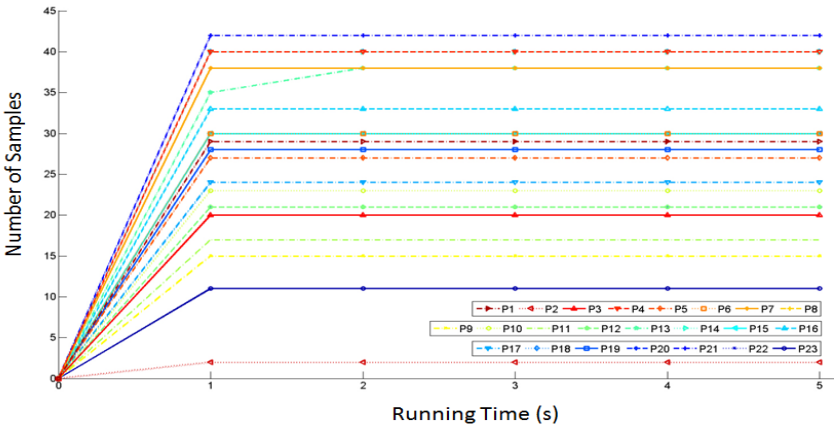


Fig. 3. Sampling Performance for 23 Ambiguous Person Names

Fig. 4A plots the relationship between document size and the size of valid samples when HCM has stabilized. Fig. 4A shows that the relationship between the valid sample size and the number of documents is roughly a linear relation. This is true at least when the number of documents is less than 100 in our experiment. However, the running time of the algorithm is determined by the total number samples including the rejected samples (made by the proposed merging or splitting operations). Fig. 4B further shows the relationship between document size and total number of samples when HCM has stabilized. Again, the graph shows a near linear behavior with the tendency to slow down when document size increases. This is consistent with the claim by Wick et al. [23, 24] that the randomized algorithm is more efficient.

In practice, we know that people hardly go beyond 100 documents to search for a person. So, the automated method is practical for use. Theoretically speaking, using pairwise method requires a quadratic comparisons of mentions ($O(N^2)$ in HAC algorithm [19]).

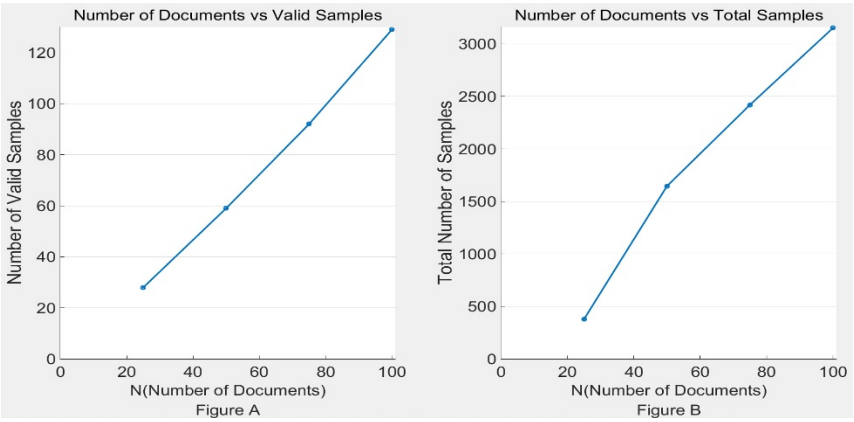


Fig. 4. Relationship between Number of Documents and Sample Size

Even though the algorithm use a rich set of features, person-specific features are rule-based and other semantic features can be obtained as general knowledge before the HCM training starts. In fact, compared to the state-of-the-art system using HAC method [6], evaluating the proposal in HCM use a smaller number of factors compared to HAC method, thus making HCM feasible to use in practice.

6 Conclusion and Future Work

This paper proposes to use a semantic-based hierarchical co-reference resolution technique which does not need threshold tuning. Our disambiguation method is semantic-based and training data independent. Experiments on the WePS2 dataset show that the proposed method outperforms all the systems that learn the threshold automatically and also achieves a comparable performance with the top two systems which manually tune the number of clusters.

This model allows a small number of upper-level entity nodes to summarize a large number of name mentions. Thus, the model's representation power and it is much more scalable over large collections of name mentions because co-reference decisions can be made between two entity nodes instead of mention pairs. This is particularly important for disambiguating millions of name mentions. The work presented in this paper focuses on Web Persons Disambiguation. Works to extend it to entity disambiguation can broaden its use in many other applications.

References

1. Artiles, J., Gonzalo, J., Verdejo, F.: A testbed for people searching strategies in the WWW. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 569–570 (2005)
2. Artiles, J., Gonzalo, J., Sekine, S.: The semEval-2007 WePS evaluation: establishing a benchmark for the web people search task. In: SemEval 2007, pp. 64–69 (2007)
3. Artiles, J., Gonzalo, J., Sekine, S.: WePS 2 evaluation campaign: overview of the web people search clustering task. In: 18th WWW Conference on 2nd Web People Search Evaluation Workshop (WePS 2009) (2009)
4. Balog, K., He, J., Hofmann, K., et al.: The University of Amsterdam at WePS2. In: WePS (2009)
5. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. JMLR 3, 993–102 (2003)
6. Chen, Y., Lee, S.Y.M., Huang, C.: PolyUHK: a robust information extraction system for web personal names. In: WePS (2009)
7. Chen, Y., Lee, S.Y.M., Huang, C.: A robust web personal name information extraction system. Expert Systems with Applications 39(3), 2690–2699 (2012)
8. Elmacioglu, E., Tan, Y.F., et al.: PSNUS: web people name disambiguation by simple clustering with rich features. In: SemEval 2007, pp. 268–271 (2007)
9. Gong, J., Oard, D.: Determine the entity number in hierarchical clustering for web personal name disambiguation. In: WePS (2009)
10. Gooi, C.H., Allan, J.: Cross-document co-reference on a large scale corpus. In: NAACL 2004 (2004)
11. Han, X., Zhao, J.: CASIANED: web personal name disambiguation based on professional categorization. In: WePS 2009 (2009)
12. Ikeda, M., Ono, S., et al.: Person name disambiguation on the web by two stage clustering. In: WePS 2009 (2009)
13. Ji, H., Lin, D.: Gender and animacy knowledge discovery from web-scale ngrams for unsupervised person mention detection. In: PACLIC, vol. 23, pp. 220–229 (2009)
14. Kozareva, Z., Vazquez, S., Montoyo, A.: UA-ZSA: web page clustering on the basis of name disambiguation. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), pp. 338–341 (2007)
15. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML 2001, pp. 282–289 (2001)
16. Long, C., Shi, L.: Web person name disambiguation by relevance weighting of extended feature sets. In: Third Web People Search Evaluation Forum (WePS-3), CLEF (2010)
17. Rao, D., Garera, N., Yarowsky, D.: JHU1: an unsupervised approach to person name disambiguation using web snippets. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), pp. 199–202 (2007)
18. Romano, L., Buza, K., Giuliano, C.: XMedia: web people search by clustering with machinely learned similarity measures. In: WePS (2009)
19. Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. Comput. Journal 16, 30–34 (1973)
20. Singh, S., Subramanya, A., Pereira, F., McCallum, A.: Large-scale cross-document coreference using distributed inference and hierarchical models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 793–803 (2011)
21. Wellner, B., McCallum, A., Peng, F., Hay, M.: An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching. In: Uncertainty in Artificial Intelligence (UAI), pp. 593–601 (2004)

22. Wick, M., Culotta, A., Rohanimanesh, K., McCallum, A.: An Entity-based Model for Coreference Resolution. In: SIAM International Conference on Data Mining (SDM) (2009a)
23. Wick, M., Singh, S., McCallum, A.: A discriminative hierarchical model for fast coreference at large scale. In: ACL 2012, pp. 379–388 (2012)
24. Wick, M., Kobren, A., McCallum, A.: Large-scale author co-reference via hierarchical entity representations. In: Proceedings of the 30th International Conference on Machine Learning (2013)
25. Xu, J., Lu, Q., Liu, Z.: Combining classification with clustering for web person disambiguation. In: WWW 2012, pp. 637–638 (2012)
26. Xu, J., Lu, Q., Liu, Z.: Aggregating skip bigrams into key phrase-based vector space model for web person disambiguation. In: KONVENS (2012)

Semantics and Dialogue

From Natural Logic to Natural Reasoning*

Lauri Karttunen

Stanford University

Abstract. This paper starts with a brief history of Natural Logic from its origins to the most recent work on implicatives. It then describes on-going attempts to represent the meanings of so-called 'evaluative adjectives' in these terms based on what linguists have traditionally assumed about constructions such as *NP was stupid to VP*, *NP was not lucky to VP* that have been described as factive. It turns out that the account cannot be based solely on lexical classification as the existing framework of Natural Logic assumes.

The conclusion we draw from this ongoing work is that Natural Logic of the classical type must be grounded in a more inclusive theory of Natural Reasoning that takes into account pragmatic factors in the context of use such as the assumed relation between the evaluative adjective and even the perceived communicative intent of the speaker.

1 What Is Natural Logic?

Natural Logic attempts to do formal reasoning in natural language making use of syntactic structure and the semantic properties of lexical items and constructions. It contrasts with approaches that involve a translation from a natural to a formal language such as predicate calculus or a higher-order logic.

Figure 1 sketches the history of Natural Logic as told by Johan van Benthem in [3] and in lectures.

The short version goes as follows. Natural Logic has been around over 2000 years. It started out pretty well with Aristotle and the Greeks who invented syllogisms, some two dozen valid patterns of inference in ancient Greek. In the medieval times all of this was ported into Latin and extended by people like William of Ockham and Buridan. With the waning of the Middle Ages began a decline in logic that bottoms out in the works of De Morgan in the middle of the 19th century. But soon came the rise of modern logic first with Gottlob Frege in the 1890s and on the Natural Logic side with Charles Sanders Peirce about the same time. The current revival in Natural Logic was started by Johan van Benthem [2] and his student Víctor Sánchez-Valencia [26] in the 1990s. Among the latest advances is the work by Jan van Eijck [8], Bill MacCartney [20] and the recent papers by Thomas Icard [10] and Larry Moss [11] that build on the work of MacCartney and Christopher Manning [21].

* The original work reported in this paper is part of a joint project with Cleo Condoravdi, Stanley Peters, and Annie Zaenen at the Center for the Study of Language and Information. Special thanks to Annie Zaenen for the content and form of this paper.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 295–309, 2015.
DOI: 10.1007/978-3-319-18111-0_23

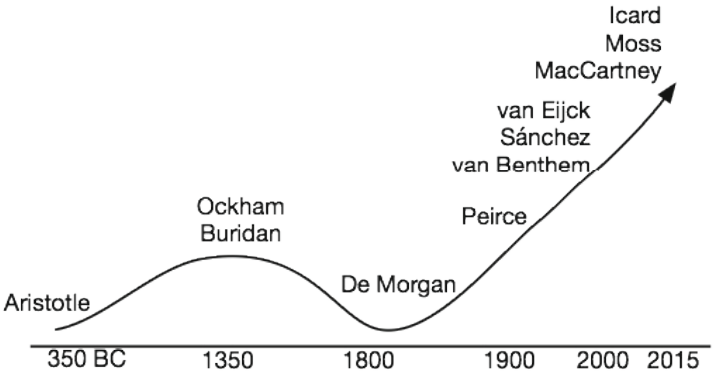


Fig. 1. A Brief History of Natural Logic

Augustus De Morgan is famous for De Morgan Laws:

The negation of a conjunction is the disjunction of two negations:
$\neg(p \wedge q) \equiv \neg p \vee \neg q$
The negation of a disjunction is the conjunction of two negations:
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Why does the does the curve of Natural Logic fall to its lowest point at his time? As van Benthem and Sánchez-Valencia point out, the laws were already clearly articulated by medieval logicians such as Ockham and Buridan. De Morgan's sole contribution is the formulation we now use. But more importantly, De Morgan was unsuccessful in his attempts to give a formal explanation of the validity and invalidity of some simple examples that medieval logicians had succeeded in explaining. For example, since horses are animals, it is obvious that (1a) is true, but since some other animals also have tails (1b) is false in our world.

(1) a. Every tail of a horse is a tail of an animal.
b. Every tail of an animal is a tail of a horse.

The difficulty in explaining the obvious in logical terms is this. Assume that we start with a tautology such as (2) that contains two instances of the word *horse*.

(2) Every tail of a horse is a tail of a horse.

The substitution of a more general term, *animal*, for the second occurrence of *horse* is a valid inference yielding (1a). But the substitution of *animal* for the first occurrence of *horse* is an invalid inference resulting in (1b).

Suppose we start with the tautology in (3).

(3) Every tail of an animal is a tail of an animal.

Replacing *animal* with the more specific term *horse* is valid in the first instance but invalid for the second one.

Sánchez-Valencia and van Benthem report that medieval logicians, such as William of Ockham,[1] did have a right solution to the puzzle although it was wrapped up in a complex theory of *Suppositions*. One reason why the problem was difficult for them was that there was no theory to describe the syntactic structure of the Latin analogues of (2) and (3).

De Morgan thought that his rules of inference validated the correct inferences from (2) and (3) to (1a) but Sánchez-Valencia shows that they also allow the invalid inferences that yield (1b). That was the low point of Natural Logic.

The first logician in modern times who gave the right answer to the puzzle was the American Charles Sanders Peirce late in the 19th century. His system is also rather complicated but it is based on the right idea. The validity of substituting a general term like *animal* for a specific one like *horse*, or vice versa, depends on the position of the target word in the syntactic structure of the sentence.

A term that occurs in a **downward monotonic** (= **antitone**) context as the first occurrence of *animal* in in (3) can be replaced by a more specific term like *horse* as in (1a). A term that occurs in an **upward monotonic** (= **monotone**) context as the second occurrence of *horse* in (2) can be replaced by a more general term such as *animal*.

The truth of (1a) and the falsity of (1b) are obvious to any speaker of English but it is not trivial for a beginning student of logic to prove (1a) expressed in first-order logic starting with (2) and the premise that horses are animals. A proof by Natural Deduction or Sequent Calculus is a substantive homework assignment. It takes many lines of reasoning in a formal language to show the validity of such simple monotonicity inferences in ordinary English.

2 Monotonicity

The distinction between 'more specific' and 'more general' terms does not only apply to nouns like *horse* and *animal*. It is applicable to expressions of any syntactic category. If X and Y are expressions of he same syntactic type, we say that X is more specific than Y if all instances of X are instances of Y but not vice versa. In the notation introduced by Bill MacCartney [20] we write this $X \sqsubset Y$ where $\sqsubset$ is a symbol for inclusion, a generalized entailment relation. For example, Figure 2 illustrates the fact that *with* is an upward monotonic operator, ↑, and *without* a downward monotonic operator, ↓.

Figure 2 shows graphically that any action done with a knife is included in actions done with a tool: *with a knife* $\sqsubset$ *with a cutter* $\sqsubset$ *with a tool*. The preposition *without* reverses these inclusion relations: *without a tool* $\sqsupset$ *without a cutter* $\sqsupset$ *without a knife*.

Table 1 codes the monotonicity properties for some English determiners as they are traditionally described.[2] For example, as we have already seen in (2) and (3), *every* creates a downward monotonic context for its first argument, a nominal phrase, and an upward monotonic context for the second argument, a verb phrase.

Such a table is however misleading in that it suggests incorrectly that the upward/downward monotonicity can be determined locally. In fact the ↓ marks in Table 1 should be

[1] Ockham was also a pioneer of three-valued logic.

[2] Some determiners do not have any monotonicity effects. *Five* is ↑↑ but *exactly five* is ==, that is, it yields no monotonicity inferences on either of its two arguments.

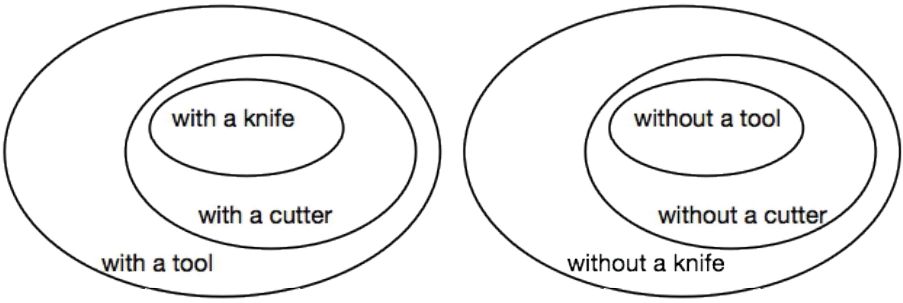


Fig. 2. With↑ NP vs. without↓ NP

Table 1. Classical monotonicity signatures for some determiners

Det	Code	Example
every	↓↑	Every house was damaged in a fire. ⊏ Every small house was damaged.
some	↑↑	Some small house was damaged in a fire. ⊏ Some house was damaged.
no	↓↓	No house was damaged. ⊏ No small house was damaged in a fire.
few	=↓	Few students have a car. ⊏ Few students have a fancy car.
many	=↑	Many professors have a fancy car. ⊏ Many professors have a car.

changed to ↑, and vice versa, if the expression appears in a negative context. If we put the construction *some NP VPs* under negation as in (4a), the valid inference pattern for *some* turns into the same as for *no* in Table 1. That is, (4a) entails (4b).

(4) a. It is not the case that some house was damaged.
b. No small house was damaged in a fire.

The same holds for *every*. In (5a) *student* is in a downward monotonic context and *cheap car* is in an upward monotonic context. Consequently, we can replace *student* by the more specific *poor student* and *cheap car* by the more general *car*. (5a) entails (5b).

(5) a. Every student has a cheap car.
b. Every poor student has a car.

But everything flips if we replace *every* by *not every*. (6a) entails (6b).

(6) a. Not every poor student has a car.
b. Not every student has a cheap car.

In (6) the first nominal argument of *every* is in an upward monotonic context that licenses the replacement of the specific term *poor student* with the more general *student*. In contrast, the phrase *has a car* in (6) is a downward entailing context that justifies replacing *car* by the more specific *cheap car*.

These complications brought in by negation have, of course, always been known to logicians and they are probably one reason why the medieval logicians, not having a

syntactic structure to build on, came up with the hard-to-understand theory of suppositions. The modern approaches to the problem starting with van Benthem and Sánchez-Valencia intially took their cue from Peirce's work and set up a two-step monotonicity calculus.

In the first step the nodes in the parse tree are marked with + or - signs using the lexical signatures for determiners in Table 1 and other functors such as *with*, *without* and *not*. The result for the example (6a) is shown schematically in Figure 3 (a). Since *not* is a downward monotonic operator it's sentential argument gets a minus sign.

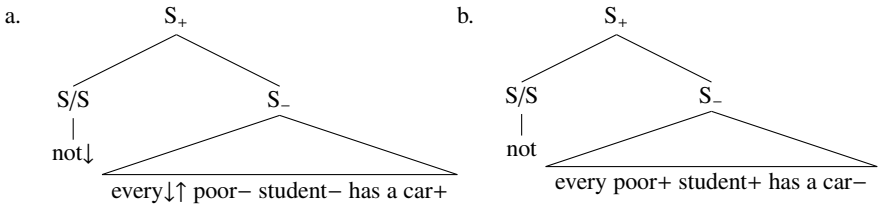


Fig. 3. Two step computation of monotonicity

The second step of the van Benthem-Sánchez algorithm traces the paths from the leaves of the parse tree to the root counting the minus signs on the path. If the number of minuses is even, the final sign is + to indicate an upward monotonic context; if the number of minuses is odd, the node is marked with - to show that it is a downward monotonic phrase. Figure 3 (b) shows that the effect in this case is to reverse the initial assignments of signs. The final marking justifies the inference from (6a) to (6b).

The two step-method of computing monotonicity is unnecessarily convoluted. David Dowty [6] describes a system that derives the marking directly in a categorial grammar. Unfortunately his bottom-up method necessitates the duplication of many lexical categories. Van Eijck [8] describes a simple top-down algorithm that computes the desired result in a single pass always starting with a positive sign on the highest node of the parse tree. That is an optimal solution to the problem.

3 Beyond Monotonicity

Historically studies of monotonicity tend to focus on the semantics of determiners and quantifier phrases. But monotonicity inferences arise with many areas of language that semanticist only recently have begun to describe such as the meaning of comparative constructions [25].

The dramatic rise in our understanding of this type of reasoning pictured in Figure 1 is not an exaggeration. Natural Logic has advanced more in the last few decades than at any time since Greek philosophers. Aristotle's 24 classical syllogisms, valid patterns of reasoning, are now understood in terms of the monotonicity properties of a few determiners and the axioms of symmetry and existential import (no empty classes) [7].

Because we now have better theories of syntax than any previous generations, we are not at all baffled by the *tail of a horse* puzzles that occupied previous generations of semanticists for centuries.

3.1 Implicatives

In this section we will focus on studies that extend the classical scope of Natural Logic from simple sentences to constructions with infinitival clauses and embedded sentences. The discussion is based on Lauri Karttunen's descriptive work on implicative constructions [15,16] and its computational implementation by Nairn et al. [22] and MacCartney [20]. The question is whether the proposition implicit in an infinitival clause is presented as true, false, or not entailed either way.

A good place to start is the example (7) in MacCartney and Manning [21]:

(7) a. James Dean refused to move without blue jeans.
b. Dean didn't dance without trousers.

Obviously—or upon reflexion at least—(7a) entails (7b). Because *without* is a downward entailing operator we would expect the entailment *without trousers* $\sqsubset$ *withouth blue jeans*. But here the entailment goes in the opposite direction, from he more specific term *blue jeans* to the more general category *trousers*. In positive contexts *dance* $\sqsubset$ *move* but in (7) the relationship is reversed because of the negative implication of *refused*.

The construction *refuse to VP* is one of the several types of implicative patterns discussed in [15] and [16]. They include two classes of **two-way implicatives** and four types of **one-way implicatives**. Table 2 contains a few examples of the first kind.[3]

Table 2. Two types of two-way implicative verbs

$++ \mid --$ implicatives	$+- \mid -+$ implicatives
manage to	fail to
bother to	neglect to
remember to	forget to
see fit to	refrain from . . . ing
happen to	avoid . . . ing

The $++ \mid --$ implicatives are constructions that in a positive context entail the truth of the infinitival clause (++). In negative contexts they entail that the infinitival clause is false (−−). Examples in (8).[4]

[3] The examples in this section involve simple verbs. See [16] for a discussion of phrasal implicatives such as *take the time/opportunity/trouble to VP*.

[4] In addition to their entailment properties all the constructions in Table 2 suggest something else as well. For example, *remember to VP* and *forget to VP* both imply that the protagonist intended or was expected to VP. That can lead to arguments such as *I didn't* ***forget*** *to go to the party. I never intended to go there* that are not about what happened but about whether forgetting was involved.

(8) a. The culprit managed to get away. ⊏ The culprit got away.
b. She didn't bother to explain. ⊏ She didn't explain.
c. He saw fit to ask her for another chance. ⊏ He asked her for another chance.
d. Kim didn't remember to have breakfast. ⊏ Kim didn't have breakfast.

The + − | − + implicatives also yield an entailment both in positive and negative context but they reverse the polarity.

(9) a. He had failed to get into Oxford. ⊏ He had not gotten into Oxford.
b. She didn't avoid getting caught. ⊏ She got caught.
c. He didn't neglect to return her call. ⊏ He returned her call.
d. Kim forgot to have breakfast. ⊏ Kim didn't have breakfast.

Constructions such as *manage to VP* and *fail to VP* are perfectly symmetrical in that they yield an entailment both in affirmative and negative contexts. There are four types of verbs that yield an entailment about their complement clause only under one or the other polarity.

Table 3. Four types of one-way implicative verbs

++ implicatives	+− implicatives	−− implicatives	−+ implicatives
cause NP to	refuse to	can	hesitate to
force NP to	be unable to	be able to	
make NP to	prevent NP from		

The examples in (10) illustrate these one-way implicative constructions. In all cases, reversing the polarity does away with any logical entailment unlike the examples of two-way implicatives in (8) and (9).

(10) a. They forced the crowd to disperse. ⊏ The crowd dispersed.
b. Dean refused to move. ⊏ Dean didn't move.
c. He was not able to get up. ⊏ He did not get up.
d. She didn't hesitate to help him. ⊏ She helped him.

If a person says *I was able to log in* one is inclined to conclude that she did, and that may well be what the speaker intends to convey. However, it is not a contradiction for her to continue *but I did not do it*.[5]

The computation of inferences from implicative verbs has been implemented, [22], [20], as the same kind of top-down process as van Eijck's method of computing monotonicity.

Figure 4 traces the assignment of polarity marks in structure containing three stacked two-way implicatives: *not*, *fail* and *remember*. Starting with positive polarity on the top

[5] An attested example of this type: *He was able to sin, but did not; he was able to do wrong, but would not.* `http://www.liturgies.net/saints/paulinusofnola/readings.htm`. Replacing *was able* by *managed* would create a contradiction.

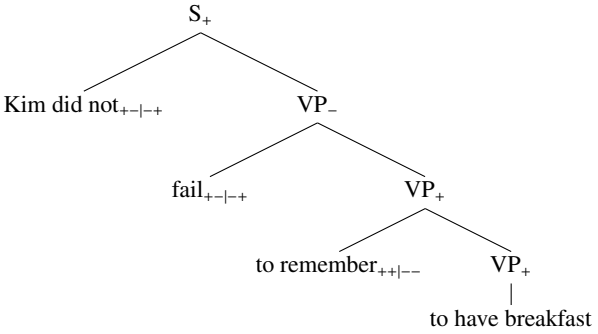


Fig. 4. Kim did not fail to remember to have breakfast ⊏ Kim had breakfast

node, the chain of VPs gets marked with + or - determined by the lexical signature of *not* or the higher verb and the polarity passed onto that clause from above.

This example entails the innermost clause, *Kim had breakfast*, because *not* and *fail* reverse the incoming polarity and *remember* preserves it. Replacing *fail* by *happen* would result in the entailment that Kim did not have breakfast. If the implicative chain is broken, say by replacing *remember* by a non-implicative verb such as *propose*, there would be no entailment about whether anyone had breakfast.

The computations with implicative verbs are a natural extension of the monotonicity calculus discussed in the previous section. This is not the case for the next topic and the subject matter on the next section. We are about to cross the boundaries of Natural Logic.

3.2 Factives

Factives and counterfactives are well-known classes of verbs that take sentential or infinitival complements, first discussed by Kiparsky and Kiparsky [19]. They were among the first types of linguistic data that sparked the debate about **presuppositions**, still inconclusive 35 years later.

Philosophers had been talking about presuppositions for much of the 20th century focusing on a few examples like *The present king of France is bald* and *Have you stopped beating your wife?* The first is due to Bertrand Russell [24], the second to Eubulides (4th century BC).[6]

When linguists got fascinated with presuppositions in the late 1960s, within a few years they made a whole zoo of 'presupposition triggers' that included a large collection of lexical items and constructions, factives and counterfactives being among the first. In hindsight, the fundamental error at the time was not to recognize the newly discovered 'triggers' were not all of the same species. They should not have all been put into the same cage. The quest for a grand unified theory of presupposition as conceived in that period has been a failure.[7]

[6] Eubulides also bequeathed us the Liar Paradox: *what I now say is false*.

[7] That is the conclusion of the article by David Beaver and Bart Geurts in the Stanford Encyclopedia of Philosophy. `http://plato.stanford.edu/entries/presupposition/`

Nevertheless, the basic observations about the meaning of factive and counteractive constructions remain unchallenged. Table 4 lists some of these constructions.

Table 4. A few factive and counterfactive constructions

factive	counterfactive
remember that	pretend that
forget that	pretend to
be bad to	
be bad that	

The difference between *remember to* vs. *remember that* is striking. In affirmative sentences and under negation *remember that* commits the speaker to the view that the embedded clause is true. With *remember to* we get a positive or negative entailment depending on the polarity of the upstairs clause, with *remember that* we only get a positive inference in (11) regardless of the polarity of the upstairs clause.

(11) a. She remembered to lock the door. $\sqsubset$ She locked the door.
b. She did not remember to lock the door. $\sqsubset$ She did not lock the door.
c. She remembered that she locked the door. $\preceq$ She locked the door.
d. She did not remember that she locked the door. $\preceq$ She locked the door

In the case of both (11a) and (11c) the speaker is committed to the proposition that she locked the door, but not in the same way. (11a) is a two-way implicative construction that yields a negative entailment under negation in (11b). (11c) and its negation (11d) presuppose that she locked the door. As before we use $\sqsubset$ for entailment and a new symbol, $\preceq$, for presupposition.

The difference between $\sqsubset$ and $\preceq$ is that presuppositions 'project' from embedded clauses in a way that entailments do not. [14] Presuppositions are impervious to negation as in (11d) and they project from the antecedent of conditionals as in (12).

(12) If she remembered that she locked the door, she did not have to drive back home. $\preceq$ She locked the door.

The difference between the two-way implicative *remember to* and the factive *remember that* cannot be pinned on the complementizer, *to* vs. *that*. The constructions *be bad to* and *be bad that* are both factive, *pretend that* and *prentend to* are both counterfactve as illustrated in (13).

(13) a. It was not bad for us that we had one day of rain on our trip. $\preceq$ We had one day of rain.
b. It was not bad for us to have one day of rain on our trip. $\preceq$ We had one day of rain.
c. Kim pretended that she had everything under control. $\preceq$ Kim did not have everything under control.
d. Kim pretended to have everything under control. $\preceq$ Kim did not have everything under control.

There are no general systems that we know of for computing inferences based on factive and counterfactive constructions or many other types of 'projective meaning.' The computational tools developed in the framework of Discourse Representation Theory, [13], [12], [4], are mainly focused on anaphoric expressions and definite noun phrases, a small subset of the phenomena that have been called presuppositions.

4 Beyond Natural Logic

The progress of Natural Logic has demonstrated that it is possible to do formal reasoning on rather shallow representations of natural language sentences. For the topics covered in the previous sections it is not evident that one could do better by a translation into a formal language such as predicate calculus or a higher-order logic. The shortcomings and unsolved problems with Natural Logic that we survey in this last section are challenging in any framework for semantics. The common thread of the inference problems discussed below is the need to take into account pragmatic factors, the context of use and even the perceived intent of the speaker.

The issue we start with is that people make inferences that go beyond what the sentence logically entails or presupposes. We call them **soft inferences** because they may explicitly cancelled but, if there is no indication otherwise, they may well a be part of what the speaker intends to convey. The second problem we uncovered in our investigation of evaluative adjectives such as *stupid*, *clever*, etc. It turns out that the interpretation of expressions like *NP was not stupid to VP* as implicative or factive depends on the relationship between the evaluative adjective and the action expressed by the VP. We call it the **consonance/dissonance effect**. Finally we discuss the curious case of *lucky* revisiting the issues first uncovered in [17] highlighting the fact that the choice of meaning may depend on the perceived intent of the speaker.

4.1 Soft Inferences

One-way implicatives yield a definite entailment only under one polarity, but in many contexts they are interpreted as if they were two-way implicatives. Although *be able* is logically a $--$ implicative (see Footnote 5), in the vast majority of occurrences 'in the wild' are like (14). The intent is certainly to convey that not only was Williamson able to deliver but also did so.

(14) New Zealand called and Kane Williamson was able to deliver.

Here *be able* is clearly used to mean *manage*.

A similar observation can be made of some $+-$ implicatives like *prevent*. Examples like (15) are not contradictory. If something is not prevented it might still not happen for other reasons.[8]

(15) Her mother did not prevent her from visiting her father, but she never did.

[8] For an insightful study of *prevent* see [5].

But such examples are vanishingly rare. In the common usage *prevent* behaves like a two-way $+-\mid -+$ implicative. When there is no reason to assume otherwise, examples like (16) are understood—and undoubtedly meant to be understood—to mean that a few laughs were had even across the language barrier.

(16) The language barrier did not prevent us from having a few laughs together.

There must be some unknown pragmatic explanation why we tend to assume that when someone says that something was allowed or not prevented to occur she means that it did occur if there is nothing to suggest that it didn't. This seems to be a universal principle, not a fact about the usage of English. This may be related to the phenomenon of **conditional perfection** discussed by Michael Geis and Arnold Zwicky [9] that pushes us to interpret simple conditionals *if p then q* as biconditionals *if p and only if p then q*.

Another case of non-logical inference is illustrated in (17).

(17) a. I meant to answer your email right away.
b. I didn't mean to hurt your feelings.

The speaker of (17a) probably hasn't quickly answrered the addressee's email. The speaker of (17b) probably thinks that she has hurt the addressee's feelings. There construction *mean to VP* is of course not logically of type $+-\mid -+$. But there is a pragmatic reason why we are inclined to draw such inferences. If we feel responsible for some bad outcome, a standard way of excusing ourselves is to assert that what happened was not what we intended. The soft inference arises from the understanding the situations where the speaker would be likely to use the expressions in (17), not from any semantic relation between the sentence and the infinitival clause.

4.2 Consonance/Dissonance Effect

Most of the work descriptive work on presupposition and entailment has focused on verbs, there is very little literature on adjectives. In our joint Stanford project [18] we decided to explore the semantics of evaluative adjectives such as *stupid*, *clever*, *wise*, *brave*, *rude*, *kind*, etc. in constructions of the form *NP was ADJ to VP* and *NP was not ADJ to VP*.[9] The only substantive treatise on this topic we found is a dissertation by Neal Norrick from the 1970s [23].

According to Norrick's classification evaluative adjectives (his term) are factive. But Norrick does not discuss any examples of the type *NP was not ADJ to VP* that would bring out the difference between factives and implicatives. Norrick's judgement has been passed on from author to author including the often cited paper by Chris Barker [1] without ever having been evaluated with real data.

We decided to study the issue with due diligence, This now means asking for judgements not just of your students, friends, and colleagues but of large set of workers on the Amazon Mechanical Turk (*Turkers* they are called). We involved over nine hundred people in subsequent iterations of our crowdsourcing experiment.

[9] This section based on joint research with Cleo Condoravdi, Stanley Peters, and Annie Zaenen but my colleagues are not responsible for the views expressed here.

Because negative sentences clearly distinguish factive and implicative constructions we were interested to know the response to sentences such as (18).

(18) Paul was not smart to take the middle piece.

It turned out that the great majority of our Turkers gave this type of sentence a factive interpretation: Paul was not smart and took the middle piece. A minority of our Turkers chose the implicative reading: Paul was not smart and did not take the middle piece. This finding suggests that there is a dialect split. The majority of our subjects prefer the factive reading, a minority prefers the implicative reading.

The surprising finding was that both populations can be pushed towards their non-favored interpretation by manipulating the interplay with the adjective and the content of the VP. Running the study on (19) gave different results for the two variants.

(19) a. Paul was not smart take the best piece.
b. Paul was not smart take the worst piece.

In (19a) the adjective *smart* and the VP *take the best piece* are in a **consonant** relation: taking the best piece would be smart. In (19b) the adjective and the VP are **dissonant**: taking the worst piece would not be smart. The results of our study so far indicate that a negation of a consonant relation such as (19a) favors the implicative interpretation: Paul did not take the best piece. The negation of a dissonant relation as in (19b) biases the Turkers towards the factive reading: Paul took the worst piece.

Figure 5 summarizes our overall findings for the 21 adjectives in our study. Assuming that there in no consonance/dissonance effect in the neutral case, the middle columns give an estimate of the proportion of factive and implicative speakers.[10]

The figure shows that about 80% of the Turkers gave a factive interpretation to dissonant examples such as (19b). In a consonant case such as (19a) the majority still preferred the factive interpretation but the number of implicative interpretations doubled from the neutral case. The consonance/dissonance was particularly strong for adjectives such as *lucky* and *fortunate*.

The sobering finding of this study that we are now in the progress of replicating with a more careful experimental design suggests that some very basic inferences such as whether the event described by an infinitival complement happened or not depend on opinions that are not part of the literal meaning of the sentence. This is a difficult problem for compositional semantics and for Natural Logic as well.[11]

[10] The Either columns shows the number of subjects who said they couldn't decide between the two possible interpretations, factive or implicative.

[11] In setting up the original experiment we tried to guess what people's opinions were, say, about how lucky it would be to go to San Francisco or live in Europe. Both neutral we thought, but the results show that for our subjects *lucky to go to San Francisco* was consonant but *lucky to live in Europe* a case of dissonance.

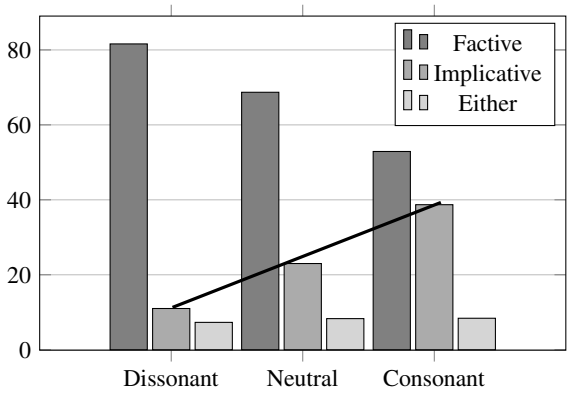


Fig. 5. Results: Percentage of Factive, Implicative, and Either choices for NP *was not* A*dj to* VP

4.3 Lucky

In addition to the strong consonance/dissonance effect, the adjective *lucky* is an interesting case for another reason. In affirmative sentence with a future tense it has two possible interpretation. A sentence such as (20) has a positive sense for most people.

(20) My future boyfriend will be so lucky to have me cook yummy food like this for him every day.

It is understood as a promise of benefits to the future boy friend. Seen as a caption to a picture of a table with delicious dishes, (20) has no other plausible interpretation.

In contrast, examples like (21) are typically interpreted as conveying a pessimistic warning: you will probably not get any return on your investments.

(21) Your will be lucky to ever get any return on your investments.

After all, what else would license the negative polarity items *ever* an *any* in the seemingly positive environments that contains none of the usual triggers of negative polarity?

An example such as (22) also suggests the pessimistic 'probably not' interpretation.

(22) You will be lucky to avoid a jail sentence.

The choice between the two meanings depends on many factors. With a small change the interpretation of (22) can be flipped:

(23) At least you will be lucky to avoid a jail sentence.

What *at least* least does here conversationally is to indicate that the speaker is trying to find something positive to say in an obviously bad situation, looking for a silver

lining on a dark cloud. The fact that we recognize the speaker's intention to console the addressee is enough to suppress the 'probably not' interpretation. See [17] for further discussion.

5 Conclusion

We are impressed by the great progress in Natural Logic in the last few years and very aware of the need to ground it in a more comprehensive framework of Natural Reasoning that supplements logical relations with pragmatic inferences. Overall we are very optimistic about the future of this enterprise. Natural Logic is very suited for computational tasks.The improvements in hardware, the software for machine learning, and the easy way to collect data from the data on the web and by experiments with tools like AMT will advance the state of the art. The challenge is the integration of pragmatic and logical information,

References

1. Barker, C.: The dynamics of vagueness. Linguistics and Philosophy 25(1), 1–36 (2002)
2. van Benthem, J.: Language in Action: categories, lambdas and dynamic logic. Studies in Logic, vol. 130. Elsevier, Amsterdam (1991)
3. van Benthem, J.: A brief history of natural logic. In: Chakraborti, M.K., Löwe, B., Mitra, M.N., Sarukkai, S. (eds.) Logic, Navya-Nyāya & Applications, Homage to Bimal Krishna Matilal. College Publications, London (2008)
4. Bos, J.: Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. Computational Linguistics 29(2), 179–210 (2003), http://dx.doi.org/10.1162/089120103322145306
5. Condoravdi, C., Crouch, D., Everett, J., Paiva, V., Stolle, R., Bobrow, D., van den Berg, M.: Preventing existence. In: Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS 2001, pp. 162–173. ACM, New York (2001), http://doi.acm.org/10.1145/505168.505184
6. Dowty, D.: The role of negative polarity and concord marking in natural language reasoning. In: Harvey, M., Santelman, L. (eds.) Proceedings of SALT 4, pp. 114–144. University of Rochester (1994)
7. van Eijck, J.: Syllogistics=monotonicity+symmetry+existential import, Technical Report, vol. SEN-R0512. CWI, Amsterdam (2005), http://db.cwi.nl/rapporten/
8. Van Eijck, J.: Natural logic for natural language. In: ten Cate, B.D., Zeevat, H.W. (eds.) TbiLLC 2005. LNCS (LNAI), vol. 4363, pp. 216–230. Springer, Heidelberg (2007)
9. Geis, M.L., Zwicky, A.M.: On invited inferences. Linguistic Inquiry 2(4), 561–566 (1971), http://www.jstor.org/stable/4177664
10. Icard III, T.: Inclusion and exclusion in natural language. Studia Logica 100(4), 705–725 (2012)
11. Icard III, T., Moss, L.: Recent progress on monotonicity. In: Zaenen, A., Condoravdi, C., de Paiva, V. (eds.) Perspectives on Semantic Representations for Textual Inference. LILT, vol. 9, pp. 167–194. CSLI Publications, Stanford (2014)
12. Kamp, H.: The importance of presupposition. In: Rohrer, C., Roßdeutscher, A., Kamp, H. (eds.) Linguistic Form and its Computation. CSLI Publications, Stanford (2001)
13. Kamp, H., Reyle, U.: From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT. Kluwer, Dordrecht(1993)

14. Karttunen, L.: Presuppositions of compound sentences. Linguistic Inquiry 4(2), 169–193 (1973)
15. Karttunen, L.: Implicative verbs. Language 47, 340–358 (2012)
16. Karttunen, L.: Simple and phrasal implicatives. In: *SEM 2012, June 7-8, pp. 124–131. Association for Computational Linguistics, Montréal (2012), http://www.aclweb.org/anthology/S12-1020
17. Karttunen, L.: You will be lucky to break even. In: King, T.H., de Paiva, V. (eds.) From Quirky Case to Representing Space. Papers in Honor of Annie Zaenen, pp. 167–180. CSLI Publications, Stanford (2013)
18. Karttunen, L., Peters, S., Zaenen, A., Condoravdi, C.: The chameleon-like nature of evaluative adjectives. In: Empirical Issues in Syntax and Semantics 10, pp. 233–250. CSSP, Paris (2014)
19. Kiparsky, P., Kiparsky, C.: Fact. In: Bierwisch, M., Heidolph, K.E. (eds.) Progress in Linguistics, pp. 143–173. Mouton, Hague (1970)
20. MacCartney, B.: Natural Language Inference. PhD thesis. Stanford University (2009)
21. MacCartney, B., Manning, D.C.: An extended model of natural logic. In: Proceedings of the Eight International Conference on Computational Semantics, pp. 140–156. Association for Computational Linguistics (2009)
22. Nairn, R., Condoravdi, C., Karttunen, L.: Computing relative polarity for textual inference. In: ICoS-5, pp. 67–76 (2006)
23. Norrick, N.R.: Factive adjectives and the theory of factivity. Niemeyer, Tübingen (1978)
24. Russell, B.: On denoting. Mind 14(56), 479–493 (1905)
25. Smessaert, H.: Monotonicity properties of comparative determiners. Linguistics and Philosophy 8(3), 295–336 (1996)
26. Sánchez-Valencia, V.: Studies on Natural Logic and Categorial Grammar. PhD thesis. University of Amsterdam (1991)

A Unified Framework to Identify and Extract Uncertainty Cues, Holders, and Scopes in One Fell-Swoop

Rania Al-Sabbagh[1], Roxana Girju[1], and Jana Diesner[2]

[1] Department of Linguistics and Beckman Institute
[2] Graduate School of Library and Information Science,
University of Illinois at Urbana-Champaign, USA
{alsabba1,girju,jdiesner}@illinois.edu

Abstract. We present a unified framework based on supervised sequence labelling methods to identify and extract uncertainty cues, holders, and scopes in one-fell swoop with an application on Arabic tweets. The underlying technology employs Support Vector Machines with a rich set of morphological, syntactic, lexical, semantic, pragmatic, dialectal, and genre-specific features, and yields an average F_1 score of 0.759.

Keywords: Uncertainty Automatic Analysis, Supervised Sequence Labeling, Unified Frameworks, Morphologically-Rich Languages, Twitter.

1 Introduction

Uncertainty refers to the language aspects that express hypotheses and speculations where propositions are held as (un)certain, (im)probable, or (im)possible. Different terms have been used to refer, more or less, to the same concept, including commitment [1], epistemic modality [2], evidentiality [3], factuality [4], speculation [5], and veridicality [6].

Automatic uncertainty analysis is crucial for many NLP applications to distinguish between factual (i.e. certain) and nonfactual (i.e. uncertain or negated) information. Some of these applications are rumour detectors that identify statements with unverified truth values [7], question-answering systems that evaluate the truth value of Web-based information before using it for answers [8], credibility analysers that detect disinformers who endorse rumours and further spread them [9,10], topical expertise finders that select trustful information holders about specific topics [11], and medical text analyzers that decide whether a patient definitely suffers, probably suffers, or does not suffer from an illness [12].

A comprehensive automatic system for uncertainty analysis ideally comprises three tasks: (1) uncertainty detection to identify and extract uncertainty linguistic cues, (2) uncertainty attribution to ascribe the cues to their holders, and (3) uncertainty scope extraction to identify the linguistic constituents encoding the propositions being modified by the cues. To-date, however, and to the best of our knowledge, automatic systems for uncertainty analysis have been limited to

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 310–334, 2015.
DOI: 10.1007/978-3-319-18111-0_24

uncertainty detection and scope extraction; whereas uncertainty attribution has been either ignored [13], or simplistically handled by setting the text writer as the default holder [1] or using a predefined set of prototypical holders [14].

Current research on uncertainty automatic analysis has also been limited to specific languages and linguistic genres. There is a plethora of work on English [15,16,17,18,19], yet nothing for agglutinative morphologically-rich languages except for the recent work on Hungarian [20]. Meanwhile, there is plenty of research on biomedical and newswire texts [21,22,23,24], Wikipedia articles [25], and product reviews [5], but only [26] have recently started research on uncertainty automatic analysis for tweets. Agglutinative morphologically-rich languages, including Hungarian and Arabic, as in this research, present significant complexities for standard approaches to uncertainty automatic analysis developed for English, including data sparsity due to the high number of variable tokens, and packaging information about the cue, its holders, and sometimes its scope in single tokens. Furthermore, Arabic flexible word order challenges standard approaches to English uncertainty scope extraction. Tweets are also challenging given that they can be grammatically incorrect, inconsistently punctuated, or even incomplete.

One main contribution of our research here is that we address the aforementioned limitations by developing a comprehensive system with three pipelined machine learning models to identify and extract uncertainty cues, holders, and scopes in Arabic tweets. That is, we work on an understudied uncertainty task, i.e., attribution, an understudied genre, i.e., tweets, and an understudied language, i.e., Arabic. The result is a novel tool with a practical impact on NLP applications that rely on uncertainty automatic analysis.

Another main contribution of our research here is the unified framework that we propose to identify and extract uncertainty cues, holders, and scopes in one fell-swoop. We cast each component of our system as a token sequence labelling task and use the predictions of one task to inform the predictions of the next task in the pipeline, starting with uncertainty detection, followed by uncertainty attribution, and then uncertainty scope extraction. There are two main advantages of this proposed unified framework: first, many features are shared across the three tasks; and hence, the time needed for feature extraction is reduced to speed up the system; second, the fact that the predictions of one task inform the predictions of the next task reduces the number of candidate tokens as the pipeline proceeds from one task to another; this reduces processing time and enhances performance. For instance, once a token is predicted as encoding an uncertainty holder, it is excluded from candidate tokens for uncertainty scope extraction because a single token cannot encode information about uncertainty holders and scopes at the same time. This exclusion process boosts performance, especially that tweets are typically short texts with a few tokens.

The rest of this paper is organized as follows: Section 2 describes our unified framework, including challenges, framework and tasks description, data, classification features, experimental set-up and results, discussions, and error analyses;

Section 3 compares and contrasts our work to closely related work; and Section 4 gives a future outlook to overcome the shortcoming of our current research.

2 Our Unified Framework

2.1 Challenges

Arabic has specific properties that make our current research a challenging one, compared to what has been done for other languages, especially English. Arabic is an agglutinative, morphologically-rich language, with a flexible word order. As a result, nouns, adjectives, and verbs inflect for gender, number, person, mood, and aspect, leading to data sparsity due to the high number of variable tokens. Furthermore, inflecting for person entails that an uncertainty cue and its holder can be both encoded in the same token. In example 1, the uncertainty cue أعتقد *>Etqd*[1] (gloss: think.1.sg.imprf; English: I think) is inflected for the first person; hence, the uncertainty holder is the same as the Twitter user who posted the uncertainty-laden tweet.

1. – أعتقد محدش هيشارك في # الاستفتاء
   – ***>Etqd*** *mHd$ hy$Ark fy #AstftA'* [2]
   – **think.1.sg.imprf** not-one-not will-participate.3.sg.msc.imprf in the-referendum.
   – I **think** no one will participate in the referendum.[3]

Agglutination also packages important uncertainty information in single tokens. In example 2, the first part of the scope is the object pronoun encliticized to the uncertainty cue فاكر *fAkr* (gloss: thinks.3.sg.msc.imprf; English: he thinks).

2. – #مرسي فاكرنا عيال هتي
   – *#mrsy* ***fAkr****nA EyAl hty*.
   – #Morsi **thinks.3.sg.msc.imprf-**us kids idiot.
   – #Morsi **thinks** we are some idiot kids.

Example 3 illustrates the two challenges of person morphological inflections and agglutination. Uncertainty holder information is represented in the 2nd person morphological inflection; instead of being encoded by a separate morpheme. Meanwhile, the scope starts at the object pronoun encliticized to the cue تحسب *tHsb* (gloss: think.2.sg.msc.imprf; English: you think).

[1] Buckwalter's transliteration scheme: `http://www.qamus.org/transliteration.htm`

[2] For all examples, uncertainty cues are in boldface, holders are underlined, and scopes are double underlined.

[3] For each example, the 1st line is the original Arabic tweet. The 2nd line is the Buckwalter's transliteration. The 3rd line is a gloss reflecting Arabic morphology, syntax, and semantics. For example, *think.1.sg.imprf* means the verb is inflected for the 1st person singular in the imperfective aspect; *will-participate.3.sg.msc.imprf* means the verb is procliticized to the future marker and is inflected for the 3rd person singular masculine in the imperfective aspect; and *the-referendum* means the noun is procliticized to the definite article. The 4th line is an English translation.

3. – تحسبهم جميعا وقلوبهم شتى #جبهة #الانقاذ
   – ***tHsbhm*** *jmyEA wqlwbhm $tY #jbhp #AlAnqAz*
   – **think.2.sg.msc.imprf-them** united but they are not #front #salvation
   – You **think** they are united but they are not. #salvation #front

In addition to Arabic agglutination and rich morphology that present challenges for uncertainty detection and attribution, Arabic flexible word order challenges uncertainty scope extraction. Arabic recognizes continuous uncertainty scopes that can either precede or follow their cues as in examples 4 and 5, respectively, and discontinuous uncertainty scopes where cues interrupt their scopes as in example 6.

4. – الإخوان مايعرفوش إن اليوتيوب ذاكرته حديدية
   – *Al<xwAn* ***mAyErfw$*** ***<n*** *Alywtywb zAkrth Hdydyp*
   – the-Brotherhood **not-know.3.pl.imprf-not that** the-YouTube memory-its iron
   – The Brotherhood **does not know that** YouTube has a strong memory.

5. – الحرب علينا بدأت فيما يبدو
   – *AlHrb ElynA bd>t* ***fymA ybdw***
   – the-war on-us started.3.sg.fm.prf **in-what seems.3.sg.msc.imprf**
   – The war against us has started, **seemingly**.

6. – فض الاعتصام بالقوة نتائجه في الأغلب غير طيبة
   – *fD AlAEtSAm bAlqwp ntA}jh* ***fy Al>glb*** *gyr Tybp*
   – dissolving the-sit-in with-the-force results-its **in the-most** not good
   – Using force to dissolve the sit-in **probably** has bad consequences.

The aforementioned challenges are pertinent to Arabic as an agglutinative morphologically-rich language, with a flexible word order. Yet, there are also language-independent challenges pertinent to uncertainty automatic analysis. First, uncertainty cues come in a variety of grammatical categories, including adjectives, adverbs, nouns, auxiliary verbs, lexical verbs, and particles. Second, uncertainty cues can be unigram or multiword expressions. Third, a single uncertainty cue may have one or more scopes as in example 7. Finally, two or more coordinating uncertainty cues may share the same scope as in example 8.

7. – أولادنا فاهمين إن دم أخواتهم راح هدر وإن ثأرهم مع الشرطة
   – *>wlAdnA* ***fAhmyn*** ***<n*** *dm >xwAthm rAH hdr* w***<n*** *v>rhm mE Al$rTp*
   – children-our **think.3.pl.imprf that** blood friends-their went.3.sg.msc.prf in.vain and**-that** revenge-their with the-police
   – Our children **think that** their friends were killed for no reason and **that** they have to take revenge from the police.

8. – البرادعي عارف ومتأكد إن ١٢ % بس هيصوتوا له
   – *ElbrAdEy* ***EArf*** w***mt>kd*** ***<n*** *12% bs hySwtwA lh*
   – Elbaradei **knows.3.sg.msc.imprf** and **sure.sg.msc that** 12% only will-vote.3.pl.imprf for-him.
   – Elbaradei **knows** and **is sure that** only 12% will vote for him.

2.2 Framework and Tasks Description

We construct a pipeline with three tasks built on top of one another:

- **Task 1**: uncertainty detection to identify uncertainty cues.
- **Task 2**: uncertainty attribution to ascribe each identified cue to its holder, one cue at a time.
- **Task 3**: uncertainty scope extraction to extract the scope(s) of each identified cue, one cue at a time.

According to the aforementioned design of our pipeline, we first identify cues, and then for each identified cue, we identify its holder, and then its scope(s), one cue at a time. Our intuition to identify holders first and then scopes is that holders might be easier to identify being encoded by shorter and syntactically simpler linguistic constituents, compared to scopes. According to our corpus observations, many holders in our corpus are base phrase noun phrases; whereas most scopes are complex complement clauses. As a result, we assume that if we start with the relatively easier Task 2 and use its predictions along with the predictions from Task 1 and other features to inform Task 3, we are likely to boost the performance for scope extraction.

We cast each task as a token sequence labelling problem and apply Support Vector Machines (SVMs) as our machine learning method. We use the YamCha implementation[4] that has been used for multiple sequence labeling problems, especially in the literature of Arabic NLP, including [27,28,29]. SVMs and Conditional Random Fields (CRFs) have been both used in the literature of uncertainty automatic analysis. To the best of our knowledge, only [16] has compared the performance of the two machine learning methods in the context of English uncertainty detection to find out that CRFs marginally improve prediction accuracy. We keep the comparison between SVMs and CRFs for our three tasks for a future work.

Starting with Task 1 of uncertainty detection, the classifier is trained to label each token as the beginning of an uncertainty cue (B-C), inside an uncertainty cue (I-C), or outside any uncertainty cues (O-C). With this BIO scheme, we manage to represent both unigram and multiword uncertainty cues as in Table 1.

Task 2 of uncertainty attribution is built on top of Task I, so that for each identified cue, one cue at a time, the classifier identifies and extracts the linguistic constituents that encode the holder. Similar to Task I, we cast Task II as a token sequence labelling problem in which B-H is the beginning of an uncertainty holder, I-H is a token inside the uncertainty holder phrase, and O-H is a token that does not encode any uncertainty holder information. As we mentioned earlier, holder information can be encoded in the morphological inflections of the cues rather than be represented by separate morphemes. In these cases, we place the BIO-H labels on the cues themselves as in Table 2 to indicate that the holder information is encoded in the morphology of the cue.

[4] `http://chasenorg/~taku/software/yamcha`

Table 1. Uncertainty cues represented in the BIO scheme and formatted based on YamCha requirements

A Unigram Cue				
Arabic	**Trans.**	**gloss**	**English**	**BIO**
أنا	*>nA*	I	I	O-C
أظن	***>Zn***	**think.1.sg.imprf**	**think**	**B-C**
مفيش	*mfy$*	there-no	there is no	O-C
فايدة	*fAydp*	benefit	benefit	O-C
من	*mn*	from	from	O-C
المقاطعة	*AlmqATEp*	the-boycott	the boycott	O-C
A Multiword Cue				
Arabic	**Trans.**	**gloss**	**English**	**BIO**
من	***mn***	**from**	**it is**	**B-C**
المتوقع	***Almtwqe***	**the-expected**	**expected**	**I-C**
تعيين	*tEyyn*	appointing	to appoint	O-C
العريان	*AlEryAn*	Aleryan	Aleryan	O-C
رئيسا	*r}ysA*	prime	as a Prime	O-C
للوزراء	*llwzrA'*	minister	Minister	O-C

Task 3 of uncertainty scope extraction is also defined as a token sequence labelling problem, represented in the BIO scheme where the three classes to predict are the beginning of an uncertainty scope (B-S), inside an uncertainty scope (I-S), or outside any uncertainty scopes (O-S). Task 3 is built on top of both Tasks 1 and 2: for each identified cue, we train the classifier to predict its scopes, one cue at a time; and, we use predicted cue and holder information as features for scope extraction. Table 3 shows an example of the BIO-S scheme.

Table 4 shows examples of raw tweets and how predictions are added as our pipeline proceeds from Task 1 to Task 3. In the final output, tokens, which have been identified as uncertainty cues in Task 1 and as encoding uncertainty holders in Task 2, are labelled as B-CH or I-CH, where CH stands for Cue/Holder. Furthermore, all tokens that have been labelled as O-C, O-H, and O-S for Tasks 1, 2, and 3, receptively, are eventually given the label NULL to indicate that they do not encode any uncertainty information.

2.3 Data

We use the uncertainty-annotated corpus from [30] for our research here. The corpus comprises 21,716 tweets with 521,786 word tokens and 64,445 word types, where words are defined as white-space delimited strings. All tweets belong to the political domain as they discuss political topics of interest in the Arab World, in general, and in Egypt, in particular. The tweets come in two Arabic varieties: Modern Standard Arabic (MSA), the formal Arabic variety typically used by press agencies, and Egyptian Arabic (EA), the local Arabic variety of Egypt. The two Arabic varieties are not mutually exclusive; that is, they can co-exist within the same tweet. Out of the 21,716 tweets, 7,461 tweets are annotated as not including any uncertainty information. The rest of the tweets comprise

Table 2. Uncertainty holders represented in the BIO scheme and formatted based on YamCha requirements

Arabic	Trans.	gloss	English	BIO
مرسي	*mrsy*	Morsi	Morsi	B-H
فاكر	***fAkr***	**thinks.3.sg.msc.imprf**	**thinks**	**O-H**
نفسه	*nfsh*	himself	he is	O-H
إله	*<lh*	god	a god	O-H
مظنش	***mZn$***	**not-think.1.sg.imprf-not**	**I do not think**	**B-H**
أن	*>n*	**that**	**that**	**I-H**
عمر	*Emr*	Omar	Omar	O-H
سليمان	*slymAn*	Suleiman	Suleiman	O-H
قادر	*qAdr*	capable.sg.msc	is capable	O-H
على	*ElY*	of	of	O-H
خوض	*xwD*	fighting	fighting	O-H
معركة	*mErkp*	battle	a battle	O-H

17,317 uncertainty cues, of which 3,697 are unigrams and 13,620 are multiword expressions. Each cue is annotated for holders, of which 7,992 are encoded in the morphological inflections of the cues, whereas the rest are represented via personal pronouns or any other base or complex noun phrases. Each cue is also annotated for scopes.

2.4 Classification Features

We use a rich set of nine feature categories illustrated in Table 5.

Contextual Features (CFs) describe the lexical and morpho-syntactic contexts of each given token. The lexical context is the sequence of tokens around each given token; whereas the morpho-syntactic context is the sequence of Part-of-Speech (POS) tags. The morpho-syntactic CFs are extracted via MADAMIRA v1.0 [31], a toolkit for Arabic morphological analysis, tokenization, and POS tagging.

Dialectal Features (DFs) identify the Arabic dialect of each given token and each given tweet, as to whether it is MSA or EA. DFs can be informative for uncertainty detection because some words can function as uncertainty cues in one Arabic variety but not the other. For example, شكلنا *$klnA* functions as an uncertainty cue only in EA where it means *it seems that*, but in MSA it is either a common noun encliticized to a possessive pronoun meaning *our look*, or a perfective verb conjugated for the 1st person plural meaning *we formed*. Likewise, the particle قد *qd* functions as an uncertainty cue only in MSA, in which it means either *indeed* if followed by a perfective verb, or *may* if followed

Table 3. Uncertainty scopes represented in the BIO scheme and formatted according to the YamCha requirements

Arabic	Trans.	gloss	English	BIO
أنا	*>nA*	I	I	O-S
أرى	***>rY***	**see.1.sg.imprf**	**see**	**O-S**
أن	***>n***	**that**	**that**	**O-S**
الثورة	*Alwrp*	the-revolution	the revolution	B-S
لن	*ln*	not	will not	I-S
تنتصر	*tntSr*	win	win	I-S
مادمنا	*mAdmnA*	as.long.as-we	as long as	I-S
في	*fy*	in	we are in	I-S
خلاف	*xlAf*	dispute	an ongoing	I-S
مستمر	*mstmr*	ongoing	dispute	I-S

Table 4. Example output of our pipeline for uncertainty automatic analysis

Arabic	Trans.	Input gloss	English	Tasks 1	2	3	Output
اعرف	***AErf***	**know.1.sg.imprf**	**I know**	B-C	B-H	O-S	B-CH
ان	***An***	**that**	**that**	I-C	I-H	O-S	I-CH
مصر	***mSr***	**Egypt**	**Egypt**	O-C	O-H	B-S	B-S
في	***fy***	**in**	**is in**	O-C	O-H	I-S	I-S
مصيبة	***mSybp***	**trouble**	**trouble**	O-C	O-H	I-S	I-S
النظام	*AlnZAm*	the-regime	The oppressive	O-C	B-H	O-S	B-H
القمعي	*AlqmEy*	the-oppressive	regime	O-C	I-H	O-S	I-H
يظن	***yZn***	**thinks.3.sg.msc.imprf**	**thinks**	B-C	O-H	O-S	B-C
أن	***An***	**that**	**that**	I-C	O-H	O-S	I-C
الشعب	*Al$Eb*	the-people	the people	O-C	O-H	B-S	B-S
سينسى	*synsY*	will-forget.3.sg.msc.imprf	will forget	O-C	O-H	I-S	I-S
مش	*m$*	not	I do not	O-C	O-H	O-S	NULL
عارفة	*EArfp*	know.1.sg.fm.imprf	know	O-C	O-H	O-S	NULL
هيحصل	*hyHSl*	will-happen.3.sg.msc.imprf	what will	O-C	O-H	O-S	NULL
إيه	*<yh*	what	happen	O-C	O-H	O-S	NULL
متهيالي	***mthyAly***	**think.1.sg.imprf**	**I think**	B-C	B-H	O-S	B-CH
المقاطعة	*AlmqATEp*	the-boycott	the boycott	O-C	O-H	B-S	B-S
هي	*hy*	is	is	O-C	O-H	I-S	I-S
الحل	*AlHl*	the-solution	the solution	O-C	O-H	I-S	I-S

Table 5. Classification features for automatic uncertainty analysis

No	Feature	Description
Contextual Features (CFs)		
1	lexical	token sequences around each token
2	morpho-syntactic	POS sequences around each token
Dialectal Features (DFs)		
3	token-dialect	the Arabic dialect of each token
4	tweet-dialect	the Arabic dialect of each tweet
Lexicon Feature (LF)		
5	token-in-lexicons	the presence/absence of each token in the Arabic uncertainty lexicons
Pragmatic Features (PFs)		
6	reported-speech	the presence/absence of (in)direct reported speech linguistic markers
7	token-location	the location of each token as to whether it comes before or after the (in)direct reported speech linguistic markers, if any
Semantic Features (SemFs)		
8	gender	the gender of each token, if applicable
9	number	the number of each token, if applicable
10	person	the person of each token, if applicable
Syntactic Features (SynFs)		
11	base-phrase-type	the type of the base phrase of which each token is a part
12	position-in-base-phrase	the position of each token within its base phase
13	syntactic-dependencies	syntactic dependencies of each token
Twitter Features (TFs)		
14	tweet-length	the number of tokens per tweet
15	token-position	the position of each token within its tweet
16	hashtag-count	the number of hashtags within each tweet, if any
17	URL-count	the number of URLs within each tweet, if any
Uncertainty Cue Features (UCFs)		
18	cue	text segment representing each identified cue
19	cue-position	the position of each identified cue in its tweet
20	cue-length	whether each identified cue is a unigram or a multiword expression
21	cue-location	whether each token comes before or after the identified cue in each tweet
22	cue-distance	the distance between each token and the identified cue in each tweet
Uncertainty Holder Features (UHFs)		
23	holder	the text segment representing the holder of each identified cue
24	holder-location	whether each token comes before or after the identified holder in each tweet
25	holder-distance	the distance between each token and the identified holder in each tweet

by an imperfective verb. Yet, in EA قد *qd* is only a comparative particle meaning *as ... as*. DFs are extracted via the Arabic dialect identifier, AIDA [38].

Lexicon Feature (LF) is only used for uncertainty detection. It is a binary feature: if a given token is in the Arabic uncertainty lexicons, the feature value is set to *true*; otherwise to *false*. For this feature, we use two lexicons: (1) a lexicon of 3,289 unigram uncertainty cues [39]; and (2) a lexicon of 4,795 multiword uncertainty cues [40].

Pragmatic Features (PFs) comprise two features: (1) a binary feature to determine whether there are linguistic markers for (in)direct reported speech,

including quotation markers and reported speech verbs such as قال *qAl* (gloss: said.3.sg.msc.prf; English: he said), زعم *zEm* (gloss: claimed.3.sg.msc.prf; English: he claimed), and أعلن *>Eln* (gloss: declared.3.sg.msc.prf; English: he declared), among many others; and (2) a binary feature to locate each token as either occurring before or after the linguistic markers of (in)direct reported speech, if there are any. Based on our corpus observation, when a direct quote has uncertainty cues, the holders come before the colon (:), which is the typical punctuation marker used with direct reported speech as in example 11. In contrast, when an indirect quote has uncertainty cues, the holder typically comes after the linguistic marker of the indirect reported speech as in example 12.

11. – أمير قطر: أؤمن أن الوطن العربي جسد واحد وأوصيكم بالثبات على الحق
    – *>myr qTr: >&mn >n AlwTn AlErby jsd wAHd w>wSykm bAlvbAt ElY AlHq*
    – prince Qatar: **believe.1.sg.imprf that** the-world the-Arab body one and-ask.1.sg.imprf-you.pl to-standing for the-right
    – The prince of Qatar: I **believe that** the Arab world is a unity and I ask you to stand for what is right.

12. – أعلن المجلس الوطني الانتقالي الليبي أنه يتوقع سقوط سرت بالكامل
    – *>Eln Almjls AlwTny AlAntqAly Allyby >nh* ***ytwqE*** *sqwT srt bAlkAml*
    – declared.3.sg.msc.prf the-council the-national the-transitional the-Libyan that-it **expects. 3.sg.msc.imprf** collapse Sert by-the-full
    – The National Transitional Council of Libya declared that it **expects** the full collapse of Sert.

Semantic Features (SemFs) describe the gender, number, and person features for each token, if applicable. SemFs are especially informative for uncertainty attribution. According to Arabic syntax, if the cue is a verb, a present participle, a noun, or an adjective, we have to expect its holder to have the same gender, number, and person features. To extract our SemFs, we rely on a few resources:

- The ATB tagset: MADAMIRA v1.0 uses the Penn Arabic TreeBank (ATB) tagset [35], which explicitly encodes gender, number, and person information if and only if they are morphologically marked by such affixes as: the feminine plural suffix of ات *At* in بنات *bnAt* (gloss: girls; English: girls), the the feminine singular suffix ة *p* in ابنة *Abnp* (gloss: daughter; English: a daughter), the 3rd person masculine imperfective prefix ي *y* in يعتقد *yEtqd* (gloss: thinks.3.sg.msc.imprf; English: he thinks), and the 1st person plural prefix ن *n* (n) in نعتقد *nEtqd* (gloss: think.1.pl.imprf; English: we think), among many others.
- Since gender, number, and person are not always morphologically represented, we also use the Arabic lexicon of semantic features from [36] that comprises 30,000 entries labeled for gender and number.
- We also use the database from [29] that comprises the words of the Penn Arabic TreeBank labeled for gender and number, among other semantic features.

Syntactic Features (SynFs) comprise two types of features: (1) shallow parsing features that describe the type of the base phrase of which each token is a part and the position of each token within its base phase; and (2) dependency parsing features that describe the syntactic dependencies among the parts of complex clauses. Many cues, holders, and scopes are base phrases. Example base phrase cues are: the prepositional phrase على الأغلب *ElY Al>glb* (gloss: on the-most; English: most probably) and the adverbial phase ربما *rbmA* (gloss: maybe; English: maybe). Likewise, holders and scopes can be base phrases such as the base noun phrase holder الرئيس *Alr}ys* (gloss: the-president; English: the president), and the base deverbal noun scope نجاح *njAH* (gloss: success; English: success) in examples 13 and 14, respectively.

13. – يعتقد الرئيس أن الخونة يسأكلوننا
    – ***yEtqd*** *Alr}ys* ***>n*** *Alxwnp sy>klwnnA*
    – **thinks-3.sg.msc.imprf** the-president **that** the-traitors will-eat.3.pl.imprf-us
    – The president **thinks that** we will be defeated by the traitors.
14. – الشعب غير واثق من نجاح مرسي
    – *Al$Eb* ***gyr wAvq mn*** *njAH mrsy*
    – the-people **not sure.sg.msc from** success Morsi
    – The people **are not sure that** Moris can succeed.

Complex cues, holders, and scopes are not uncommon, however, especially that the annotation guidelines of the corpus we use are based on [32]'s maximal length principle, according to which the marked text segments for holders and scope must include all related complements and adjuncts. Consequently, we decide to use both shallow and dependency parsing information. Shallow parsing features are extracted via the Arabic shallow parser of AMIRA v2.0 [33]. Dependency parsing features are extracted via the CATiB dependency parser [34].

Twitter Features (TFs) describe for each tweet (1) its length (i.e. number of tokens), (2) the number of hashtags, if any, (3) the number of URLs, if any, and (4) the position of each token in the tweet. Twitter-based features have been found useful for English uncertainty detection [26].

Uncertainty Cue Features (UCFs) are extracted from the output of Task 1, i.e., uncertainty detection, and used for the next two tasks in the pipeline, namely uncertainty attribution and scope extraction. For each detected cue, we describe the following UCFs:

- **Cue:** the text segment representing the cue.
- **Cue-Length:** whether the cue is a unigram or a multiword expression.
- **Cue-Position:** the position of the cue in the tweet. Typically, cues at tweet-initial positions have their holders encoded in their morphological inflections for person.
- **Cue-Location:** whether each token comes before or after the identified cue.
- **Cue-Distance:** the distance between each token and the cue, defined as a numeric value.

Uncertainty Holder Features UHFs: are extracted from the output of Task 2, i.e., uncertainty attribution, and are used to inform Task 3, i.e., uncertainty scope extraction. For each identified holder, we extract the following UHFs:

- **Holder:** the text segment representing the holder
- **Holder-Location:** whether each token comes before or after the identified holder
- **Holder-Distance:** the distance between each token and the identified holder, defined as a numeric value.

2.5 Experimental Set-Up

We use the same experimental set-up for each task in our pipeline. To find our optimal machine learning model, we implement a 10-fold cross validation method in which the whole corpus is partitioned into 10 disjoint segments: for each fold we train on 9 segments and test on the 10^{th}. For each task, we run our experiments to find the optimal:

- **Feature category combination**, using a greedy algorithm. For the first round of the algorithm, we start by evaluating each feature category on its own, and then we select the highest performing feature category, compared to the baseline. For the second round of the algorithm, we combine the best category from the first step with other feature categories, making 2-feature-category combinations; and then we select the highest performing 2-feature-category combination. For the third round, we use the best combination from the second step, and combine it with one feature category at a time, forming 3-feature-category combinations; and then we select the highest performing 3-feature-category combination. We continue the algorithm until we reach the largest best combination of feature categories.
- **Linear context width**, which is the window of tokens whose features are considered. For instance, a linear context width of ± 2 means that the feature vector for any given token includes, in addition to its own features, those of 2 tokens before and after it as well as the predictions of 2 tokens before it.
- **Polynomial degree**, starting with 2, the default polynomial degree of YamCha SVMs implementation.
- **Parsing direction**: forward (left to right) vs. backward (right to left).
- **Multiclass method**: one-against-the-rest vs. pairwise.

Accuracy, precision, recall, and F_1 scores reported in the next subsection are all averaged over the 10-fold cross validation runs. For Tasks 2 and 3, the reported results are based on the pipeline results not the gold UCFs and UHFs. This gives a more realistic view of the performance of our system.

2.6 Experimental Results and Discussions

Uncertainty Detection. As a baseline, we use a lexicon look-up model given that lexicons of Arabic uncertainty cues do exist. This baseline is the same as the lexicon feature number 5 from Table 5.

Our experiments show that one-against-the-rest multiclass classification in a forward parsing direction (i.e. left to right) is the best configuration with the default YamCha polynomial kernel degree of 2. The optimal linear context width is found to be ±4.

Table 6. Results for uncertainty detection with the best feature category combination marked with an asterisk

No	Feature Category Combinations	Accuracy	Precision	Recall	F_1
0	Baseline	0.451	0.428	0.538	0.477
1	CFs	0.778	0.748	0.723	0.735
2	CFs+SynFs	0.835	0.799	0.782	0.790
3	CFs+SynFs+LF	0.856	0.832	0.791	0.811
4	CFs+SynFs+LF+SemFs	0.888	0.848	0.793	0.819
5	CFs+SynFs+LF+SemFs+DFs*	0.904	0.854	0.813	0.838
6	CFs+SynFs+LF+SemFs+DFs+TFs	0.909	0.849	0.830	0.839

According to Table 6, our greedy algorithm for feature selection finds that the best stand-alone feature category, compared to the baseline, is the CFs category. This is expected. About 78.65% of the cues in our corpus are multiword expressions that consist of a head verb/noun/adjective and (1) a complementizer such as يعرف إن *yErf <n* (gloss: knows.3.sg.msc.imprf that; English: he knows that), (2) a preposition such as واثق في *wAvq fy* (gloss: sure.sg.msc in; English: sure that), or (3) a preposition and a complementizer like يؤمن بأن *y&mn b>n* (gloss: believes.3.sg.msc.imprf in-that; English: he believes that). CFs contribute to identifying such cue-distinguishing subcategorization frames starting with complementizer and/or prepositions.

In the second round of the feature selection greedy algorithm, the optimal 2-feature category combination comprises CFs and SynFs, with an F_1 increase of 0.055, compared to the first round of the algorithm. SynFs are found useful to detect discontinuous multiword cues. In Arabic, the parts of multiword uncertainty cues are not always adjacent to one another. For instance, the multiword cue من المتوقع أن *mn AlmtwqE >n* (gloss: from the-expected that; English: it is expected that) can have several linguistic constituents inserted in-between its parts, some of which are:

- the negation particle غير *gyr* (not) in من غير المتوقع أن *mn gyr AlmtwqE >n* (gloss: from not the-expected that; English: it is not expected that)
- the adverbial phrase جدا *jdA* (very) in من المتوقع جدا أن *mn AlmtwqE jdA >n* (gloss: from the-expected very that; English: it is very expected that)
- the negation particle غير *gyr* (not) and the adverbial phrase أبدا *>bdA* (at all) in من غير المتوقع أبدا أن *mn gyr AlmtwqE >bdA >n* (gloss: from not the-expected at.all that; English: it is not expected at all that)

- a prepositional phrase like من العرب *mn AlErb* (gloss: from the-Arabs; English: from the Arabs) in من المتوقع من العرب أن *mn AlmtwqE mn AlErb >n* (gloss: from the-expected from the-Arabs that; English: it is expected from the Arabs that)
- an even longer prepositional phase as in من المتوقع من العرب عموما والمصريين خصوصا أن *mn AlmtwqE mn AlErb EmwmA wAlmSryyn xSwSA >n* (gloss: from the-expected from the-Arabs generally and-the-Egyptians particularly that; English: it is expected, from the Arabs, in general, and the Egyptians, in particular, that)

Out of 13,620 multiword cues in our corpus, 10,544 do not have any linguistic constituents in-between their parts and are base phrases, 1,875 have one in-between linguistic constituent, 710 have two, 260 have three, and the rest have four or more. As a result, SynFs significantly improve performance for uncertainty detection, by relating the different non-adjacent parts to the heads of the multiword cues and by detecting base phrase cues.

SemFs introduce a small precision increase, which is statistically significant according to our paired t-test on the 10-fold cross validation runs (p-value = 0.01228). Similarly, DFs increase the recall rate with a p-value of 0.0093, that is also statistically significant.

In the last round of our greedy feature selection algorithm, the highest performing feature category combination includes the TFs. Yet, the difference between the F_1 scores of combinations numbers 5 and 6 is only 0.001, which is not statistically significant (p-value = 0.768). Consequently, we stop our search and consider combination number 5 as our optimal feature category combination for uncertainty detection.

Uncertainty Attribution. As a baseline, we use a simple bag-of-words model, based on token sequences around each given token. This baseline model is the same as the lexical contextual feature number 1 in Table 5.

Table 7. Results for uncertainty attribution with the best feature category combination marked with an asterisk

No	Feature Category Combinations	Accuracy	Precision	Recall	F_1
0	Baseline	0.372	0.413	0.489	0.448
1	CFs	0.664	0.698	0.708	0.703
2	CFs+SynFs	0.699	0.733	0.718	0.725
3	CFs+SynFs+UCFs	0.727	0.761	0.742	0.751
4	CFs+SynFs+UCFs+PFs	0.736	0.770	0.747	0.758
5	CFs+SynFs+UCFs+PFs+SemFs	0.742	0.776	0.750	0.763
6	CFs+SynFs+UCFs+PFs+SemFs+TFs*	0.750	0.784	0.762	0.773
7	CFs+SynFs+UCFs+PFs+SemFs+TFs+DFs	0.746	0.780	0.768	0.774

Similar to uncertainty detection, one-against-the-rest multiclass classification in a forward parsing direction (i.e. left to right) is the best configuration with

the default YamCha polynomial kernel degree of 2. The optimal linear context width is found to be -10 and +4. It is expected for uncertainty attribution to need a larger linear context width to find the holder of each identified cue, one cue at a time, given the following three facts about Arabic syntax: (1) the flexible word order of Arabic accepts holders to precede or follow their cues; (2) long dependencies between cues and their holders occur more frequently when holders precede their cues as in example 15; that is why the optimal left linear context width for uncertainty attribution is as large as -10; and (3) holders tend to be adjacent to their cues when the holders follow the cues as in example 16; as a result, the best right linear context width is only +4.

15. – جلال أمين: الدولة البوليسية ليست دولة قوية بل دولة فاسدة في رأيي
    – *jlAl >myn: Aldwlp_Albwlysyp_lyst_dwlp_qwyp_bl_dwlp_fAsdp* ***fy r>yy***
    – Galal Ameen: the-state_the-police_not_state_strong_but_state_corrupt **in opinion-my**
    – Galal Ameen: a_police_state_is_not_a_strong_state,_but_a_corrupt_one, **in my opinion**.

16. – يحسب الناس أن يتركوا دون عقاب
    – ***yHsb*** *AlnAs* ***>n*** *ytrkwA_dwn_EqAb*
    – **think.3.sg.msc.imprf** the-people **that** left.3.pl.imprf.passive_without_punishment
    – The people **think that** they will not be punished

Similar to uncertainty detection, CFs and SynFs win the second round of the greedy feature selection algorithm. As per expectation, morpho-syntactic contextual and syntactic features abstract away from the surface tokens, that are highly variable given Arabic rich morphology; and, hence they can capture phrase and clause structures encoding holders more successfully.

UCFs significantly improve performance with an F_1 score increase of 0.026. One main advantage of using UCFs is reducing the number of tokens to be considered for uncertainty attribution. As we mentioned earlier, only 7,992 holders out of 17,317 are encoded in the morphological inflections of their cues. This entails that in the majority of cases a token that has been labeled as B-C or I-C is unlikely to be considered for uncertainty attribution. This elimination process of noisy tokens is one main advantage of our proposed unified framework, in which the predictions of one task informs the predictions of the next task in the pipeline.

PFs make it to the fourth round of the feature selection greedy algorithm. As we mentioned earlier, we have noticed that (in)direct reported speech is very systematically structured in our corpus: for indirect reported speech, holders typically come after the reported speech linguistic markers; and for direct reported speech, holders tend to come before the reported speech linguistic markers. SemFs and TFs introduce small, yet statistically significant, improvements with paired t-test p-values of 0.0136 and 0.0345, respectively. Yet, DFs do not yield any significant improvements as we compare the outputs of the sixth and seventh rounds of our feature selection greedy algorithm; paired t-test p-value $= 0.837$.

Uncertainty Scope Extraction. Similar to uncertainty attribution, we use a simple bag-of-words model, based on token sequences around each given token,

as our baseline model, which is again the same as the lexical contextual feature number 1 in Table 5. The baseline model performs worse for uncertainty scope extraction than it does for uncertainty attribution. This is expected given that linguistic structures encoding scopes are typically longer and more complex than those encoding holders.

Table 8. Results for uncertainty scope extraction with the best feature category combination marked with an asterisk

No	Feature Category Combinations	Accuracy	Precision	Recall	F_1
0	Baseline	0.405	0.359	0.381	0.369
1	CFs	0.584	0.531	0.492	0.511
2	CFs+SynFs	0.618	0.574	0.528	0.550
3	CFs+SynFs+UCFs	0.640	0.610	0.584	0.597
4	CFs+SynFs+UCFs+UHFs	0.683	0.655	0.642	0.648
5	CFs+SynFs+UCFs+UHFs+SemFs	0.699	0.669	0.650	0.659
6	CFs+SynFs+UCFs+UHFs+SemFs+TFs*	0.702	0.678	0.653	0.665
7	CFs+SynFs+UCFs+UHFs+SemFs+TFs+PFs	0.705	0.677	0.661	0.669

Similar to both uncertainty detection and attribution, one-against-the-rest multiclass classification in a forward parsing direction (i.e. left to right) is the best configuration with the default YamCha polynomial kernel degree of 2. Although we mentioned earlier that in Arabic uncertainty scopes can precede or follow their cues, in our corpus, about 93.4% of the scopes follow their cues. As a result, the optimal right linear context width for uncertainty scope extraction is as large as +11; whereas the optimal left linear context width is only -3.

CFs, SynFs, and UCFs make it right away to the third round of our greedy algorithm for feature selection similar to uncertainty attribution. The UHFs, however, win the fourth round of the algorithm, mainly because UHFs combined with UCFs eliminate even more tokens from being considered for scopes. The efficiency of UCFs and UHFs is supported by the typical short length of tweets: once tokens are labeled as encoding cues and others as encoding holders, a few tokens remain to be considered for scopes. This highlights, one more time, the advantage of the unified framework that we propose in this research to identify and extract uncertainty cues, holders, and scopes in one fell-swoop.

SemFs and TFs introduce small significant improvements, with paired t-test p-values of 0.0342 and 0.0629, respectively. Significant performance improvements stop at the sixth round of the feature selection greedy algorithm with PFs, giving insignificant improvement (p-value = 0.3079).

2.7 Error Analyses

In this section, we highlight the points of weakness of our optimal models for the identification and extraction of uncertainty cues, holders, and scopes.

Uncertainty Detection. In the output of our uncertainty detector, we identify five error triggers, arranged below from the most to the least frequent. The first

error trigger is tokens that occur in the same lexical and morpho-syntactic context, whether or not they convey uncertainty. In both examples 17 and 18, افتكرت *Aftkrt* occurs in a tweet-initial position followed by a complementizer although in 17 it denotes uncertainty, but in 18 it does not.

17. – افتكرت ان الناس حتثور لما تشوف التعذيب
    – ***Aftkrt An*** *AlnAs Htvwr lmA t$wf AltE*yb*
    – **thought.1.sg.prf that** the-people will-rebel.3.sg.fm.imprf when witness.3.sg.fm.imprf the-torture
    – I **thought that** the people will rebel when they witness the torture
18. – افتكرت ان الناس شافت بنت بتتسحل وسكتت
    – *Aftkrt An AlnAs $Aft bnt bttsHl wskt sAEthA*
    – remembered.1.sg.prf that the-people witnessed.3.sg.fm.prf girl.sg.fm tortured.3.sg.fm.prf .passive and-remained.silent.3.sg.fm.prf
    – I remembered that the people witnessed a girl being tortured and took no action.

The second error trigger is highly biased tokens such as لازم *lAzm*. In 97.4% of its occurrences, it denotes obligation as in example 19, and in the rest it denotes uncertainty as in example 20.

19. – لازم الشعب يفوق قبل فوات الأوان
    – *lAzm Al$Eb yfwq qbl fwAt Al>wAn*
    – must the-people wake.up.3.sg.msc.imprf before missing the-opportunity
    – The people must wake up before it is too late.
20. – لازم مرسي خايف من الصوابع
    – ***lAzm*** *mrsy xAyf mn AlSwAbE*
    – **must** Morsi afraid.sg.msc of the-fingers
    – **It must be that** Morsi is afraid of conspiracies.

The third error trigger is discontinuous multiword cues with very long in-between linguistic constituents. In example 21, the complementizer أن *>n* (gloss: that; English: that) is nine tokens apart to the right of its head cue verb استبعدت *AstbEdt* (gloss: excluded.3.sg.fm.prf; English: she excluded the possibility), because the noun phrase that represents the holder falls in-between.

21. – استبعدت داليا زيادة المدير التنفيذي لمركز ابن خلدون للدراسات الإنمائية أن يكون روبرت فورد هو السفير الأمريكي
    – ***AstbEdt*** *dAlyA zyAdp Almdyr Alnfyzy lmrkz Abn xldwn lldrAsAt Al<nmA}yp* ***>n*** *ykwn rwbrt fwrd hw Alsfyr Al>mryky*
    – **excluded.3.sg.fm.prf** Mai Zeyada the-executive the-manager for-center Ibn Khaldwn for-the-studies the-developmental **that** be Robert Ford the-ambassador the-American
    – Mai Zeyada, the executive manager of Ibn Khaldwn Center for Developmental Studies, **excluded the possibility that** Robert Ford is the (coming) American ambassador

The fourth error trigger is cues with incomplete subcategorization frames. As we mentioned earlier in Section 2.6, one reason that CFs and SynFs yield high results for uncertainty detection is that they contribute to identifying cue-distinguishing subcategorization frames starting with complementizers and/or prepositions. Sometimes, due to stylistic preferences, such complementizers and prepositions are removed. Hence, cues miss one key identifying feature, as in example 22.

22. – متهيالي الناس مش حتسكت المرة دي
– ***mthyAly*** *AlnAs m$ Htskt Almrp dy*
– **think.1.sg.imprf** the-people not will-remain.silent.3.sg.fm.imprf the-time this
– I **think** people will not let it go this time

Finally, tokenization and POS tagging errors contribute to the uncertainty detection errors, especially that the current available version of MADAMIRA v1.0 does not fully support Arabic dialects that make a good portion of our corpus.

Uncertainty Attribution. Two main factors contribute to uncertainty attribution errors. The first and the most frequent is long, syntactically complex clauses encoding holders. Syntactic complexity results from multiple coordinating phrases as in example 23, recursive descriptive relative clauses as in example 24, and apposition[5] as in example 25, among many other linguistic structures.

23. – مبارك وطنطاوي والمجلس العسكري ومؤيدينهم واتباعهم واللي يتشدد لهم شايفين إن الشيعة أخطر من اليهود
– *mbArk wTnTAwy wAlmjls AlEskry wm&ydynhm wAtbAEhm wAlly yt$dd lhm* ***$Ayfyn <n*** *Al$yEp >xTr mn Alyhwd*
– Mubarak and-Tantawy and-the-council the-military and-supporters-their and-followers -their and-who supports.3.sg.msc.imprf-them **think.3.pl.imprf that** the-Shiites more .dangerous than the-Jews
– Mubarak, Tantawy, the Military Council, their followers, and their supporters **think that** the Shiites are more dangerous than the Jews.

24. – الناس اللي مش عاجبها كلامي واللي بتتريق عليا مقتنعين ان مرسي كان شغال في ناسا
– *AlnAs Ally m$ EAjbhA klAmy wAlly bttryq ElyA* ***mqtnEyn >n*** *mrsy kAn $gAl fy nAsA*
– the-people who not like.3.sg.fm.imprf talk-my and-who mocking.3.sg.fm.imprf on-me **convinced.pl that** Morsi was working in NASA
– The people who do not like my arguments and who are mocking me **are convinced that** Morsi was working for NASA.

25. – شادي حميد مدير بروكنغز في قطر: من غير المرجح إن يكون الإخوان المسلمين جزء من المشهد السياسي مستقبلا
– *$Ady Hmyd mdyr brwkngz fy qTr:* ***mn gyr AlmrjH >n*** *ykwn Al<xwAn Almslmyn jz' mn Alm$hd AlsyAsy mstqblA*
– Shady Hameed manager Brookings in Qatar: **from not the-likely that** be the-Brotherhood the-Muslim part from the-scene the-political future
– Shady Hameed, the manager of Brookings, Qatar: **it is unlikely that** the Muslim Brotherhood will be politically active in the future

The second factor contributing to uncertainty attribution errors is holders inserted in-between the boundaries of multiword cues as in examples 13, 16, 21 in previous sections and 26 below.

26. – أكد الناشط الحقوقي نجاد البرعي أنه لا يوجد سبب واضح للهجوم على المراكز
– ***>kd*** *AlnA$T AlHqwqy njAd AlbrEy* ***>nh*** *lA ywjd sbb wADH llhjwm ElY AlmrAkz*
– **assured.3.sg.msc.prf** the-activist the-humanitarian Nijad Alborei **that** no exists .3.sg.msc.imprf reason clear to-attack on the-headquarters
– Nijad Alborei, the humanitarian activist, **assured that** there is no clear reason for attacking the headquarters.

[5] Apposition is a grammatical construction in which two elements, normally noun phrases, are placed side by side, with one element serving to identify the other in a different way.

Uncertainty Scope Extraction. Arranged based on their frequency, scope extraction errors include (1) scopes starting with pronouns encliticized to their cues, (2) syntactically complex scopes, typically comprising subordinate clauses, (3) scopes outside the sentence boundaries of their cues, and (4) scopes outside the tweets of their cues.

As we mentioned earlier, given that Arabic is an agglutinative language, the first parts of scopes can sometimes be object pronouns encliticized to their cues as in example 2 in Section 2.1 and example 27 below. Typically, the tokenizer should split those object pronouns. Yet, because the tokenizer we use, MADAMIRA v1.0, does not fully support Egyptian Arabic, many object pronouns go untokenized.

27. – العسكر فاكرينا هنخاف منهم
– *AlEskr* ***fAkrynA*** *hnxAf mnhm*
– the-army **thinks.3.pl.imprf-**us will-fear.3.pl.imprf from-them
– The army leaders **think** we are afraid of them.

Although tweets are typically short and syntactically simpler compared to texts from other genres, some tweets can include complex sentences as in example 28 where the scope comprises two coordinating subordinate clauses.

28. – أعتقد أنه لن يتم التوصل لاتفاق بين قوى المعارضة حتى لو مر ١٠٠ عام حتى لو مات كل المصريين
– ***>Etqd >nh*** *ln ytm AltwSl lAtfAq byn qwY AlmEArDp HtY lw mr 100 EAm HtY lw mAt kl AlmSryyn*
– **think.1.sg.imprf that** not reach.3.sg.imprf.passive to-agreement among power the-opposition even if passed.3.sg.msc.prf 100 year even if died.3.sg.msc.prf all the-Egyptians
– I **think that** the opposition will not get to an agreement, even if they spend a 100 years trying, even if all Egyptians die

Due to stylistic variations, scopes are not always in the same sentence of their scopes as in example 29. Furthermore, scopes are not always in the same tweet as their cues. Tweets are like ongoing conversations among the users in which uncertainty information can be scattered across several tweets. However, inter-tweets scopes are only 500 cases in our corpus.

29. – يعني الإخوان هيسكتوا على حرق مقراتهم؟ ... مظنش
– *yEny Al<xwAn hysktwA ElY Hrq mqrAthm?* ... ***mZn$***
– meaning the-Brotherhood will-remain.silent.3.pl.imprf on burning headquarters-their ... **not-think.1.sg.imrp-not**
– is it that: the Brotherhood will not react against burning its headquarters? ... I **do not think** (so)

3 Related Work

As we mentioned in our introduction, there are no automatic systems that work on uncertainty detection, attribution, and scope extraction in one fell-swoop. Yet, there is plenty of work on separate tasks for uncertainty automatic analysis, especially uncertainty detection and scope extraction. Furthermore, the

identification and extraction of linguistic cues, holders, and scopes is not restricted to uncertainty. Negation and opinion expressions have also undergone much research to identify their cues, holders, and scopes. Due to space limitations, we will not review such vast literature. In this section, we briefly point out some substantial differences between our work on the three uncertainty tasks and others' work.

[20] has worked on uncertainty detection in the morphologically-rich language of Hungarian, casting uncertainty detection as a token sequence labeling problem. She has not explicitly mentioned how Hungarian rich morphology challenges standard approaches to automatic uncertainty detection designed for English; yet we assume that the challenges are, more or less, the same as the ones we mentioned earlier for Arabic. A substantial difference between our task of uncertainty detection and [20]'s work is that we use a simpler definition of uncertainty. She trains her classifier to identify eight different types of uncertainty cues based on which linguistic markers are used to express uncertainty. The eight types are: epistemic modality, hedges, weasels, peacocks, investigation, dynamic, doxastic, and condition. In contrast, we do not follow this fine-grained classification and regardless of the type of the linguistic markers used to express uncertainty, we label each token as B-C, I-C, or O-C. As a result, we attain higher results for uncertainty detection with an F_1 score of 0.838, compared to her macro-F_1 score of 0.396 averaged across her eight types. [20] and [26], who also use the same fine-grained classification on uncertainty cues, do not mention the practical implications for their fine-grained classification. For instance, none of them claims that weasels may denote higher/lower uncertainty degrees; that peacocks have different syntactic realizations of other uncertainty-related information like holders and scopes; or that hedges are more likely to be used for formal linguistic genres or are more likely associated with specific language varieties. As a result, we do not see our decision to dispense with their fine-grained classification of uncertainty cues as a simplistic approach but rather as a more practical one. [41,42,43,44], among others, do not use such a fine-grained classification, either.

Many researchers work simultaneously on both uncertainty detection and scope extraction. Yet, they do not use a unified framework for both tasks. Some researchers cast uncertainty detection as a token sequence labeling problem, and then use hand-crafted rules to extract scopes [15,44,48,49]. Others define both tasks as token sequence labeling problems; yet use separate feature sets for each tasks or do not use the output of one task to inform the other [50]. In our unified framework, we use pretty much similar features for all our three tasks, reducing the required time for feature extraction. Furthermore, we extensively use the output of one task to inform the next tasks in the pipeline and eliminate noisy candidate tokens from being considered for further labeling, e.g., once a token is given a holder label, it cannot be considered for scope labeling.

As we have mentioned earlier, there is no prior work on uncertainty attribution to the best of our knowledge. Setting a default holder as in [1], is unlikely to work for the genre of tweets. As we have already seen through the aforementioned

examples in previous sections, the users who posted the uncertainty-laden tweets are not always the same as the holders. Work on holder identification and extraction has been mostly done for opinion expressions. Approaches include using prototypical holders [14], hand-crafted rules [45], dependency parsers [46], and semantic parsing [47]. None of the aforementioned works defines holder identification and extraction as a token sequence labeling problem or uses holder information to acquire further opinion information such as scopes.

4 Conclusion and Outlook

We presented a unified framework to identify and extract uncertainty cues, holders, and scopes in one fell-swoop. The main ideas behind our proposed framework are (1) to use almost the same feature set for the three tasks to reduce the time required for feature extraction, and (2) to use the output of one task to inform the next task and eliminate noisy candidate tokens so as to boost performance. We applied our framework to an understudied type of languages in the context of uncertainty automatic analysis, namely agglutinative, morphologically-rich languages with flexible word orders such as Arabic, and also to an understudied linguistic genre, i.e., tweets. We also worked on uncertainty attribution that is usually overlooked while most attention has been given to uncertainty detection and scope extraction. Furthermore, our research results in a novel NLP tool with a practical impact and an averaged F_1 score of 0.759 for uncertainty detection, attribution, and scope extraction.

For future work, there are a few ideas to work on. First, we did not measure the performance of individual features within each feature category. Instead, we used each feature category as a whole. Some individual features may need to be filtered out. Second, we used SVMs like many previous studies on uncertainty automatic analysis. Yet, it might be a good idea to compare SVMs to CRFs. Only [16] made the comparison for English uncertainty detection and found out that CRFs marginally improve the accuracy of the predictions, but substantially improve speed.

Acknowledgement. We would like to thank Mona Diab, Heba Elfardy, and Mohamed Al-Badrashiny for processing our data through the Arabic dialect identifier, AIDA. We would also like to thank Nizar Habash and Yuval Marton for providing a beta version of the CATiB dependency parser.

References

1. Diab, M., Levin, L., Mitamura, T., Rambow, O., Prabhakaran, V., Guo, W.: Committed Belief Annotation and Tagging. In: Proceedings of the 3rd Linguistic Annotation Workshop, Suntec, Singapore, pp. 68–73 (2009)
2. Palmer, F.R.: Mood and Modality. Cambridge University Press, Cambridge (1986)
3. Aikhenvald, A.Y.: Evidentiality. Oxford University Press, UK (2004)

4. Saurí, R., Pustejovsky, J.: FactBank: A Corpus Annotated with Event Factuality. Language Resources and Evaluation 43, 227–268 (2009)
5. Díaz, N.: Detecting Negated and Uncertain Information in Biomedical and Review Texts. In: Proceedings of the Student Research Workshop Associated with RANLP 2013, Hissar, Bulgaria, pp. 45–50 (2013)
6. de Marneffe, M., Manning, C., Potts, C.: Did it Happen? The Pragmatic Complexity of Veridicality Assessment. Computational Linguistics 38, 301–333 (2012)
7. Qazvinian, V., Rosengren, E., Radev, D., Mei, Q.: Rumor has it: Identifying Misinformation in Microblogs. In: Procedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 1589–1599 (2011)
8. de Marneffe, M., Grimm, S., Potts, C.: Not a Simple Yes or No: Uncertainty in Indirect Answers. In: Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pp. 136–143. Queen Mary University of London (2009)
9. Castillo, C., Mendoza, M., Poblete, B.: Information Credibility on Twitter. In: Proceedings of the 20th International Conference on World Wide Web, Heydrabad, India, pp. 675–684 (2011)
10. Soni, S., Mitra, T., Gilbert, E., Eisenstein, J.: Modeling Factuality Judgments in Social Media Text. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), Baltimore, Maryland, USA, pp. 415–420 (2014)
11. Wagner, C., Liao, V., Pirolli, P., Nelson, L., Strohmaier, M.: It's not in their Tweets: Modeling Topical Expertise of Twitter Users. In: Proceedings of the 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT, Washington DC, USA, pp. 91–100 (2012)
12. Mowery, D.L., Velupillai, S., Chapman, W.: Medical Diagnosis Lost in Translation: Analysis of Uncertainty and Negation Expressions in English and Swedish Clinical Texts. In: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), Montreal, Canada, pp. 56–64 (2012)
13. Baker, K., Bloodgood, M., Dorr, B.J., Callison-Burch, C., Filardo, N.W., Piatko, C., Levin, L., Miller, S.: Modality and Negation in SIMT. Computational Linguistics 38(2), 411–438 (2012)
14. Wiegand, M., Klakow, D.: Prototypical Opinion Holders: What We can Learn from Experts and Analysts. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2011), Missar, Bulgaria, pp. 282–288 (2011)
15. Orelid, L., Velldal, E., Oepen, S.: Syntactic Scope Resolution in Uncertainty Analysis. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijin, China, pp. 1379–1387 (2010)
16. Prabhakaran, V.: Uncertainty Learning Using SVMs and CRFs. In: Proceedings of the 14th Conference on Computational Natural Language Learning: Shared Task, Uppsala, Sweden, pp. 132–137 (2010)
17. Prabhakaran, V., Bloodgood, M., Diab, M., Dorr, B., Levin, L., Piatko, C., Rambow, O., Van Durme, B.: Statistical Modality Tagging from Rule-based Annotations and Crowdsourcing. In: Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics, Jeju, Republic of Korea, pp. 57–64 (2012)
18. Tjong, E., Sang, K.: A Baseline Approach for Detecting Sentences Containing Uncertainty. In: Proceedings of the 14th Conference on Computational Natural Language Learning: Shared Task, Uppsala, Sweden, pp. 148–150 (2010)

19. Szarvas, G., Vincze, V., Farkas, R., Möra, G., Gurevych, I.: Cross-Genre and Cross-Domain Detection of Semantic Uncertainty. Computational Linguistics 38(2), 335–367 (2012)
20. Vincze, V.: Uncertainty Detection in Hungarian Texts. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014): Technical Papers, Dublin, Ireland, pp. 1844–1853 (2014)
21. Kilicoglu, H., Bergler, S.: Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective. In: Proceedings of BioNLP 2008: Current Trends in Biomedical Natural Language Processing, Ohio, USA, pp. 46–53 (2008)
22. Zhou, H., Li, X., Huang, D., Li, Z., Yang, Y.: Exploiting Multi-Features to Detect Hedges and Their Scope in Biomedical Texts. In: Proceedings of the 14th Conference on Computational Natural Language Learning: Shared Task, Uppsala, Sweden, pp. 106–113 (2010)
23. Vincze, V., Szarvas, G., Móra, G., Ohta, T., Farkas, R.: Linguistic Scope-Based and Biological Event-Based Speculation and Negation Annotations in the BioScope and Genia Event Corpora. Journal of Biomedical Semantics 2(5), 1–11 (2011)
24. Szarvas, G., Gurevych, I.: Uncertainty Detection for Natural Language Watermarking. In: Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan, pp. 1188–1194 (2013)
25. Vincze, V.: Weasels, Hedges and Peacocks: Discourse-Level Uncertainty in Wikipedia Articles. In: Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan, pp. 383–391 (2013)
26. Wei, Z., Chen, J., Gao, W., Li, B., Zhou, L., He, Y., Wong, W.: An Empirical Study on Uncertainty Identification in Social Media Context. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 58–62 (2013)
27. Shaalan, K., Abo Bakr, H., Ziedan, I.: A Hybrid Approach for Building Arabic Diacritizer. In: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, Athens, Greece, pp. 27–35 (2009)
28. Habash, N., Roth, R.: 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, Portland, Oregon, pp. 875–884 (2011)
29. Alkuhlani, S., Habash, N.: Identifying Broken Plurals, Irregular Gender, and Rationality in Arabic Text. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, pp. 675–685 (2011)
30. Al-Sabbagh, R., Girju, R., Diesner, J.: 3arif: A Corpus of Modern Standard and Egyptian Arabic Tweets Annotated for Epistemic Modality Using Interactive Crowdsourcing. In: Proceedings of the 25th Conference on Computational Linguistics (COLING 2014), Dublin, Ireland, pp. 1521–1532 (2014)
31. Pasha, A., Al-Badrashiny, M., Diab, M., Elkholy, A., Eskandar, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.: MADAMIRA: a Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, pp. 1094–1101 (2014)

32. Szarvas, G., Vincze, V., Farkas, R., Csirik, J.: The BioScope Corpus: Annotation for Negation, Uncertainty and their Scope in Biomedical Texts. In: Proceedings of BioNLP 2008: Current Trends in Biomedical Natural Language Processing, Columbus, Ohio, pp. 38–45 (2008)
33. Diab, M.: Second Generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools, Cairo, Egypt, pp. 285–288 (2009)
34. Marton, Y., Habash, N., Rambow, O.: Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features. Computational Linguistics 39(1), 161–194 (2013)
35. Maamouri, M., Bies, A., Krouna, S., Gaddeche, F., Bouziri, B.: Penn Arabic Treebank Guidelines. In: Linguistic Data Consortium (2009)
36. Elghamry, K., Al-Sabbagh, R., ElZeiny, N.: Cue-Based Bootstrapping of Arabic Semantic Features. In: Proceedings of the 9th International Conference on Statistical Text Analysis, Lyon, France, pp. 85–95 (2008)
37. Alkuhlani, S., Habash, N.: A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers, pp. 357–362 (2011)
38. Elfardy, H., Al-Badrashiny, M., Diab, M.: AIDA: Identifying Code Switching in Informal Arabic Text. In: Proceedings of the 1st Workshop on Computational Approaches to Code Switching, Doha, Qatar, pp. 94–101 (2014)
39. Al-Sabbagh, R., Girju, R., Diesner, J.: Using the Semantic-Syntactic Interface for Reliable Arabic Modality Annotation. In: Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2014), Nagoya, Japan, pp. 410–418 (2013)
40. Al-Sabbagh, R., Girju, R., Diesner, J.: Unsupervised Construction of a Lexicon and a Repository of Variation Patterns for Arabic Modal Multiword Expressions. In: Proceedings of the 10th Workshop on Multiword Expressions (MWE), Göthenburg, Sweden, pp. 114–123 (2014)
41. Moncecchi, G., Minel, J., Wonsever, D.: Improving Speculative Language Detection Using Linguistic Knowledge. In: Proceeding of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics, Jeju, Republic of Korea, pp. 37–46. of Korea (2012)
42. Velupillai, S.: Shades of Certainty: Annotation and Classification of Swedish Medical Records. PhD thesis, Stockholm University (2012)
43. Verbeke, M., Frasconi, P., Van Asch, V., Morante, R., Daelemans, W., De Raedt, L.: Kernel-Based Logical and Relational Learning with kLog for Hedge Cue Detection. In: Muggleton, S.H., Tamaddoni-Nezhad, A., Lisi, F.A. (eds.) ILP 2011. LNCS, vol. 7207, pp. 347–357. Springer, Heidelberg (2012)
44. Yang, H., De Roeck, A., Gervasi, V., Willis, A., Nuseibeh, B.: Speculative Requirements: Automatic Detection of Uncertainty in Natural Language Requirements. In: Proceedings of 20th IEEE International Conference on Requirements Engineering, pp. 11–20 (2012)
45. Wiegand, M., Klakow, D.: The Role of Predicates in Opinion Holder Extraction. In: Proceedings of the Workshop on Information Extraction and Knowledge Acquisition, Hissar, Bulgaria, pp. 13–20 (2011)
46. Lu, B.: Identifying Opinion Holders and Targets with Dependency Parser in Chinese News Texts. In: Proceedings of the NAACL HLT 2010 Student Research Workshop, Los Angeles, California, pp. 46–51 (2010)

47. Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D.: Extracting Opinion Propositions and Opinion Holders Using Syntactic and Lexical Cues. In: Computing Attitude and Affect in Text: Theory and Applications, pp. 125–141. Springer Netherlands (2006)
48. Apostolova, E., Tomuro, N., Demner-Fushman, D.: Automatic Extraction of Lexico-Syntactic Patterns for Detection of Negation and Speculation Scopes. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers, Portland, Oregon, pp. 283–287 (2011)
49. Velldal, E., Ovrelid, L., Oepen, S.: Resolving Speculation: MaxEnt Cue Classification and Dependency-Based Scope Rules. In: Proceedings of the 14th Conference on Computational Natural Language Learning: Shared Task, Uppsala, Sweden, pp. 48–55 (2010)
50. Zhao, Q., Sun, C., Liu, B., Cheng, Y.: Learning to Detect Hedges and their Scope Using CRFs. In: Proceedings of the 14th Conference on Computational Natural Language Learning: Shared Task, Uppsala, Sweden, pp. 100–105 (2010)

Lemon and Tea Are Not Similar: Measuring Word-to-Word Similarity by Combining Different Methods

Rajendra Banjade, Nabin Maharjan, Nobal B. Niraula,
Vasile Rus, and Dipesh Gautam

Department of Computer Science, The University of Memphis,
Memphis, TN, 38152, USA
{rbanjade,nmharjan,nbnraula,vrus,dgautam}@memphis.edu

Abstract. Substantial amount of work has been done on measuring word-to-word relatedness which is also commonly referred as similarity. Though relatedness and similarity are closely related, they are not the same as illustrated by the words *lemon* and *tea* which *are related but not similar*. The relatedness takes into account a broader ranLemge of relations while similarity only considers subsumption relations to assess how two objects are similar. We present in this paper a method for measuring the semantic similarity of words as a combination of various techniques including knowledge-based and corpus-based methods that capture different aspects of similarity. Our corpus based method exploits state-of-the-art word representations. We performed experiments with a recently published significantly large dataset called Simlex-999 and achieved a significantly better correlation ($\rho = 0.642$, $P < 0.001$) with human judgment compared to the individual performance.

Keywords: Similarity, Relatedness, Word-to-Word Similarity.

1 Introduction

Understanding the meaning (semantics) of texts is one of the core problems in the field of Natural Language Processing (NLP). Semantic similarity, i.e. quantifying and deciding how similar the meanings of two given texts are, is one approach to the natural language understanding problem. In this paper, we focus on the more specific task of measuring the similarity of words, i.e. quantifying to what extent two words have similar meanings. The dictionary definition of similarity is: *resembling without being identical* (cf. Oxford Dictionary). For example, *intelligent* and *genius* are highly similar. On the other hand, measuring relatedness (also called association) is to find out to what extent the given words are related or associated to each other. The related words are not necessarily similar words. For instance, *lemon* and *tea* are related but they are not similar as they mean very different things. We focus here on assessing how similar two words are.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 335–346, 2015.
DOI: 10.1007/978-3-319-18111-0_25

A considerable amount of effort has been put on calculating the semantic relatedness or association of words which is sometimes referred as similarity. Existing methods, especially those based on co-occurrence of words in a large collection of documents, have achieved significant results on measuring the relatedness of words [9]. However, explicitly quantifying the similarity of words fosters the development of applications that benefit from similarity than those which take into account a broader range of relations. To this end, we present a method that combines several diverse approaches that rely on corpus and knowledge bases. Our hypothesis is that different methods capture different aspects of semantic similarity and their meaningful combination produces better results.

The task of word-to-word similarity has many applications, such as automatic answer grading [7], [18], [24], plagiarism detection [22]. In general, word-to-word similarity can be combined to measure similarity of texts at various levels, thus, being useful in a wide range of applications. For example, word similarity is crucial to accurately measure the correctness of student answers in educational technologies such as intelligent tutoring systems. In such systems, a widely used approach is to assess how semantically similar a target student answer, e.g. to a Physics problem, is to a reference answer, i.e. an answer provided by an expert and which is deemed correct. For instance, if a student answer contains the word *velocity* and the expert answer includes the word *acceleration,* the question is whether we should deem the student response correct. We argue that semantic relatedness measures would lead to an incorrect assessment as the relatedness score will be high. While being related, *acceleration* and *velocity* are two different concepts. Semantic similarity methods would not assess *acceleration* and *velocity* as highly similar.

Based on the types of resources used, the methods that measure semantic relatedness or similarity are broadly of two types: those that rely on knowledge bases, such as WordNet [4], and those that infer word associations from bigger collections of texts based on word co-occurrence, called distributional methods. In the knowledge base category, WordNet based methods for calculating word similarity and relatedness are quite popular [14], [16]. On the other hand, distributional similarity methods include LSA [13], LDA [2], HAL [3], ESA [6], GloVe [21]. Recently, Deep Learning based methods [17] are also in use.

Previous methods, individually or as a combination of different methods, have yielded very good performance when it comes to measuring relatedness [28]. However, as [9] explored, distributional similarity methods are not capturing well the true similarity between words. They also published a dataset containing 999 word pairs (called Simlex-999) with human rated similarity scores. We combined various knowledge based and corpus based methods by applying Linear Regression and Support Vector regression to measure semantic similarity and achieved state-of-the-art results.

The rest of the paper is organized as follows. The next section provides an overview of related work. Then, we describe the approach for combining different methods. The Experiments and Results section describes our experimental setup and the results obtained. We conclude the paper with discussion and conclusions.

2 Related Work

There exist a large number of measures for computing word-to-word relations. As already mentioned, these techniques can be broadly classified into two main categories: knowledge-based, those relying on pre-existing knowledge resources (thesauri, semantic networks, taxonomies or encyclopedia), and corpus-based, those inducing distributional properties of words from corpora.

The knowledge-based techniques use the structure of semantic networks or ontologies (e.g. is-a hierarchy in Princeton WordNet [4]) and work on distance-based measures on the network's paths [14], [15], [30]. These can further be improved by using the Information Content of the lowest common subsumer in the hierarchy and corpus statistics [12], [16], [23]. Moreover, the WordNet gloss overlap measure can be used for inferring similarity [20]. Such methods are implemented in the WordNet::Similarity package [20] and also included the SEMILAR toolkit[1] (a semantic similarity toolkit, hereinafter referred to as SEMILAR) [25].

Another category of word-to-word similarity measures rely on corpus to compute a similarity score. For example, Latent Semantic Analysis (LSA; [13]), Explicit Semantic Analysis (ESA; [6]), Global Vector (GloVE; [21]), or Latent Dirichlet Allocation (LDA; [2]) exploit the distributions of words in large collections of documents. LSA and ESA work by generating semantic models or spaces in which words are represented as vectors, the values of which being, for instance, weighted frequencies of occurrences within given documents. On the other hand, LDA models documents as topic distributions and topics as distributions over words in the vocabulary. In this case, each word can be represented as a vector encoding its contribution to the LDA generated topics. The distributed representations, such as deep learning based models, are another type of methods in this category. In distributed representations, each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities [10]. One of the popular works on distributed representations is by Mikolov et al. [17] where they used probabilistic feed forward neural network language model to estimate word representations in vector space. As such, for all these methods, the similarity between words can be, and usually is, computed in terms of cosine similarity between corresponding vectors.

Datasets to assess the performance of word-to-word similarity and relatedness methods have been developed as well. One of the most widely used, the RG dataset, consists of 65 noun pairs of words collected by Rubenstein and Goodenough [28], who had them judged by 51 human subjects in a scale from 0.0 to 4.0 according to their similarity, but ignoring any other possible semantic relationships that might appear between the terms. However, this dataset contains only nouns and is quite small to build supervised models. Another dataset which has been quite popular is WordSim-353 [5] which contains 353 word pairs, each associated with an average of 13 to 16 human judgments. In WordSim-353, there were no distinctions made

[1] http://semanticsimilarity.org

between similarity and the relatedness during its annotation. Similarly, there are other datasets that do not distinguish similarity and relatedness during their annotation[2].

While there is a significant volume of work in this area in term of methods and datasets, there is not much work focusing on measuring similarity of words which is subtly different from measuring relatedness of words. This argument is in fact supported by the recent publication of the Simlex-999 corpus focusing exclusively on similarity. We focus in this paper on measuring similarity by combining knowledge-based and corpus-based measures. A closely related work is by Agirre et al. [1] where they took different approaches for measuring between similarity and relatedness. They proposed a WordNet based method and co-occurrence based methods. They conducted experiments on the RG dataset and annotated word pairs in WordSim-353 as similar or related. However, their dataset does not represent an important class of concept pair (associated but not similar entities) [9]. In our case, we combined various features including the similarity scores calculated with most recently published resources, such as Mikolov's word representations, GloVe vectors. Moreover, we did experiments with the recently published and larger dataset which Simlex-999 which consists of a set of 999 word pairs annotated with human judgments of similarity scores [9]. Each pair in Simlex-999 was explicitly judged for similarity by at least 36 people.

3 Approach for Combining Similarity Methods

Our approach is to combine different methods in a meaningful way. In order to combine methods, the overall performance of individual method should be relatively weak and at the same time capture different aspects of the data. This idea is similar to bagging where a set of weak classifiers can be shown to lead to a significantly stronger classifier by combining their outputs.

Knowledge based approaches are typically based on hand-coded relations among words e.g. synonymy, antonymy etc.; they utilize those relations which are important to define similarity but are hard to extract accurately using fully automated methods. A typical example of a lexical database that encodes explicit lexico-semantic relations among words is WordNet [4]. WordNet based methods quantify the similarity of words based on various relations among words, and heuristics and graph theories. However, their coverage is low. Furthermore, WordNet based methods require the mapping of words to concepts, i.e. a word sense disambiguation steps which could be extremely challenging to learn automatically. On the other hand, distributional representations capture various associations among words based on the principle that words that occur in similar context are related or similar.

There exist different approaches of representing the meaning of words and measuring their relationships e.g. LSA [13], ESA [6], LDA [2], [26], Neural Language Model (NLM) [17]. Even within a category, there exist diverse methods that are based on different premises. For example, Landauer et al. [13] claim that in

[2] A list of some datasets can be found at `http://www.cs.cmu.edu/~mfaruqui/suite.html`

LSA the meaning of words can be represented based on contextual-usage of the words through statistical computations applied to a large corpus of text. Deep Learning word embeddings are developed based on the idea that Neural Networks mimic the human mind and the connections among nodes are capable of representing complex irregularities [10], and so on.

To assess whether the combination would be helpful, we calculated similarity scores for each pair using different methods (described in Experiments and Results section) and chose the best score (i.e., score most close to the gold score) among them. By using the best scores, we obtained correlation $\rho = 0.959$ which is better than the correlation among any of the individual method's output. It indicates that each of them was performing well as compared to the other methods on particular subsets of instances and that their combination can achieve an impressive correlation with human judgments. This observation forms the basis of our approach. It is important to add that the individual methods we use are built around models developed from fairly large, albeit different, data sets. One can argue that comparing these individual methods is not fair because, as mentioned, they were trained on different data sets. However, it is not in the scope of this paper to compare individual methods but rather exploit the fact that they have different strengths and weaknesses and by combining them we hope to add up the strengths and smooth out the weakness, which is what our results indicate that we achieved.

One obvious way to combine methods would be linear regression. However, linear combination in higher dimension using kernel-based methods such as support vector machines can capture interesting relations between similarity scores obtained using individual methods. Therefore, we also experimented with support vector regression.

4 Experiments and Results

4.1 Data

We use a dataset which was recently released [9]. The dataset consists of 999 word pairs (called Simlex-999) with human generated similarity scores. The word pairs were annotated by human judges with similarity scores using a Likert-scale from 0 (no similarity) to 10 (exactly mean same thing). We were particularly interested in this corpus as the previously available benchmark datasets were not balanced (i.e., contained one category of words), contained a small number of instances, or annotated without making any distinction between relatedness and similarity. For example, RG [28] dataset contains 65 word pairs which is quite small, particularly for similarity model development. The other widely used dataset, WordSim-353 [5], contains 353 word pairs but their annotation does not differentiate similarity and relatedness. Moreover, Hill et al. [9] indicate that Simlex-999 is notably more challenging to model than the alternative datasets.

The Simlex-99 dataset contains 111 adjective pairs (A), 666 noun pairs (N), and 222 verb pairs (V). Each pair was rated by at least 36 native English speakers and the average score was assigned as final human judgment (i.e., gold score). The inter-rater agreement was calculated as the average of pairwise Spearman ρ correlations between

the ratings of all respondents. Overall agreement was ρ = 0.670. To make the scores consistent with the system generated scores, we normalized the human-rated scores (by dividing them by 10).

4.2 Features

As mentioned in the previous section, we used similarity scores of various methods as features in regression models. We describe the individual methods below.

WN$_{Combined}$: There are various similarity methods based on WordNet. We used *Lesk [20]*, *Jcn* [12], *Lin* [16], *Hso* [11], *Wup* [30], and *Path* [20]. Many of them work only on specific POS categories. For this reason, we combined their outputs and a single set of scores was generated as,

- Average of *Lesk*, *Jcn*, and *Lin* measures for verbs
- Average of *Lesk* and *Hso* measures for adjectives
- Average of *Lesk*, *Wup*, *Res*, *Jcn*, *Lin*, and *Path* measures for noun pairs

A word can have multiple senses. These methods were configured to use the first sense only.

WN$_{Syn}$, **WN$_{Ant}$**: indicates whether there is a synonymy (antonymy for WN$_{Ant}$) relation in WordNet between the given word pair. We only checked the synsets of given POS category.

LSA$_{Wiki}$, **LSA$_{Tasa}$**: The cosine similarity scores calculated using LSA models developed from the whole Wikipedia articles and TASA (Touchstone Applied Science Associates) corpus as described in Stefanescu et al. [27]. The Wiki LSA models were developed using an early spring 2013 Wikipedia version, containing 4,208,450 articles. TASA comprises 60,527 samples from 6,333 textbooks, works of literature, and popular works of fiction and nonfiction. These LSA models have word representations in 300 latent dimensions. Specifically, we used Wiki_NVAR_f7 and TASA_NVAR models that are available at SEMILAR website. The Wiki_NVAR_f7 model was developed considering only the lemmas of content words occurring at least 7 times. The TASA_NVAR is similar to Wiki_NVAR_f7, but with no frequency threshold.

CRDE: Similarity using word vectors generated by applying Deep Learning technique. We used 200-dimensional word representation model developed by Turian et al. [29][3]. Word embeddings were induced using neural language model. They used RCV1 corpus which has 37 million words in 1.3 million sentences after cleaning.

UMBC: Similarity calculated using UMBC system [8][4] without using POS information. This system calculates similarity using HAL (Hyperspace Analog to

[3] http://metaoptimize.com/projects/wordreprs/
[4] http://swoogle.umbc.edu/SimService/api.html

Language) [3] model developed using Wikipedia and the similarity score is boosted using WordNet knowledge.

ESA: Score calculated using Explicit Semantic Analysis [6]. We used web service of ESAlib[5] to calculate these scores. However, we got valid numeric scores for 916 word pairs only. Due to this reason, we did not use this feature for the regression but we present the correlation score calculated ignoring the others.

MK-NLM: Neuro probabilistic language model based word representations developed by Mikolov et al. [17] [6]. We used 300-dimensional word vectors developed by training distributed representations of words with the Skip-gram model on part of Google News dataset (about 100 billion words).

GloVe: Score calculated using word representation model proposed by Pennington et al. [21] [7] and trained on 42 billion Common Crawl words. We used 300-dimensional word representation model.

LDA_{wiki}: Score calculated using a Latent Dirichlet Allocation (LDA) model generated from whole Wikipedia articles (documents with less than 500 words, stopwords and words that occur in less than 500 documents were removed) [31]. Then a 300-topic LDA model was developed (using 270,290 documents and the vocabulary of 59,136 words). The word-topic association vector was used for similarity calculation. The model was developed using JGibbsLDA[8] in high performance computing machines.

4.3 Experiments

First, we calculated similarity scores using different methods and measured their correlations (r) with the human judgments (see Table 1). For WordNet based methods, similarity scores were calculated for adjectives, nouns, and verbs separately (scores were calculated only if the method supported that POS category). After that, a single set of scores (i.e., $WN_{Combined}$) was generated using similarity scores from all WordNet based methods as described before. We also checked whether the words were synonyms or antonyms. We used WordNet 3.0 for all of these operations. For vector based methods, we calculated cosine similarity scores using the word representation vectors. In this case, we did not use POS information of the word pairs as each of the models we used has single representation for each word. For missing words (10 words in the case of the LSA Wiki model, and 6 words in the LSA TASA model), we obtained synonyms from WordNet and replaced the original word by the vector of one of the synonyms that was found in the models.

[5] http://ticcky.github.io/esalib/
[6] http://code.google.com/p/word2vec/
[7] http://www-nlp.stanford.edu/projects/glove/
[8] http://jgibblda.sourceforge.net/

Second, we applied Linear Regression (LR) and Support Vector Regression (SVR) to combine the results obtained from different methods - all or subsets of methods (see Table 2 for the results). The Weka tool was used for both linear regression and support vector regression (using LibSVM[9]). For evaluation purpose, we applied 10-fold cross validation method which gives a very good estimate of the performance of the model.

4.4 Results

Table 1 presents correlations (Pearson and Spearman's rank correlation coefficients are separated by /; first one is Pearson correlation) of similarity scores produced using different methods with human judgments. The rows are numbered and the row number is used to refer to the result of that particular method.

Table 1. Correlation (Pearson and Spearman correlation coefficients are separated by /) of similarity scores generated using different methods and human judgment in Simlex-999 dataset.

ID	Method	All	Adjective	Noun	Verb
1	Lesk	0.347/0.404	0.418/0.422	0.373/0.448	0.301/0.315
2	Hso	0.324/0.330	0.264/0.236	0.421/0.460	0.223/0.204
3	Wup	-	-	0.471/0.489	0.246/0.180
4	Res	-	-	0.454/0.443	0.245/0.219
5	Jcn	-	-	0.462/0.451	0.279/0.121
6	Lin	-	-	0.462/0.452	0.289/0.252
7	Path	-	-	0.513/0.507	0.216/0.031
8	Lch	-	-	0.534/0.506	0.109/0.031
9	$WN_{Combined}$	0.362/0.322	0.418/0.422	0.535/0.507	0.327/0.285
10	LSA_{Tasa}	0.251/0.271	0.015/0.042	0.332/0.343	0.221/0.214
11	LSA_{Wiki}	0.277/0.273	0.250/0.285	0.325/0.318	0.153/0.154
12	CRDE	0.144/0.157	0.198/0.190	0.136/0.161	0.129/0.119
13	GloVe	0.400/0.373	0.550/0.574	0.433/0.404	0.194/0.177
14	Mk-NLM	0.453/0.442	0.597/0.592	0.459/0.452	0.348/0.321
15	LDA_{wiki}	0.228/0.288	0.321/0.334	0.240/0.325	0.181/0.173
16	UMBC	**0.557/0.558**	**0.624/0.613**	**0.599/0.591**	**0.522/0.490**
17	ESA	0.145/0.271	-	-	-
18	Avg (9-16)	0.488/0.491	0.536/0.556	0.522/0.511	0.427/0.393

[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

In addition to the results on overall data, the Table 1 presents the results grouped by adjective, noun, and verb. These results are plotted in a histogram as shown in Figure 2. The UMBC system performed the best in the overall category as well as in the individual category followed by Mk-NLM and GloVe based methods. The noun similarity using WordNet based method is also high. The similarity of adjectives and nouns are better correlated with human judgments than that of verbs. However, the performance of LSA_{Tasa} on adjectives is very low. It may be due to low presence of adjectives in TASA corpus which contains academic texts.

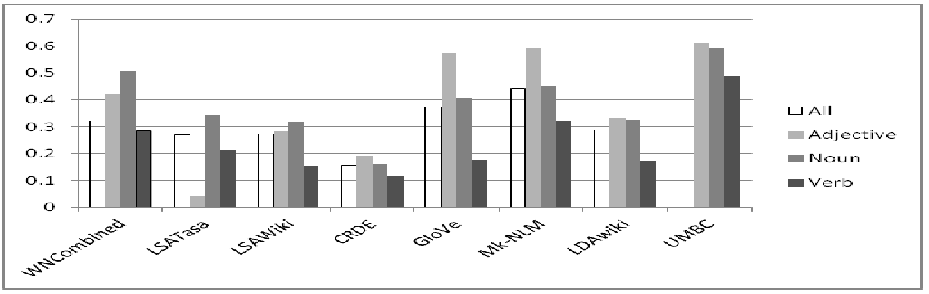


Fig. 1. Graph showing the performance (ρ) of different methods grouped by POS category

As discussed in Section 3, we chose the best score (i.e., score close to the gold score) for each word pair among the $WN_{Combined}$, WN_{syn}, WN_{ant}, LSA_{Tasa} , LSA_{Wiki}, CRDE, UMBC, GloVe, LDA_{Wiki} and Mk-NLM scores, and calculated the correlation with human judgments. The correlation (ρ) was 0.959. This indicated the huge potential in improving the similarity calculation by taping the power of individual method.

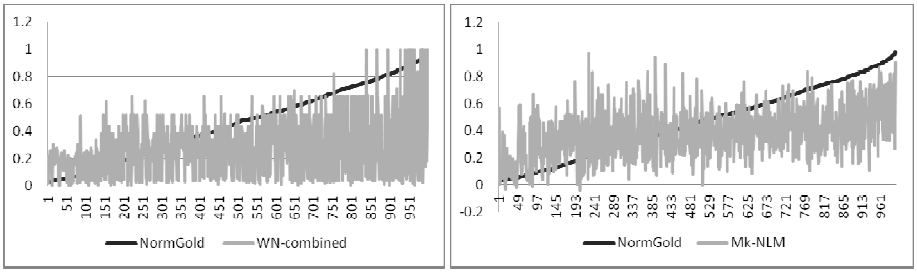


Fig. 2. The graphs showing the scores predicted by $WN_{combined}$ and Mk-NLM and the normalized gold score (instances sorted by the gold score)

Moreover, the graphs in Figure 2 show the output of representative methods $WN_{combined}$ and Mk-NLM. This also illustrates that combining the methods could improve performance overall, which is what we present next.

The results produced by individual methods were combined by applying linear regression (LR), and support vector regression implementation in LibSVM (SVR). The results are presented in Table 2. The average inter-annotator agreement of SimLex-999 is ρ = 0.670 and the best reported score by Hill et al. [9] is ρ = 0.446 where they used dependency based word embeddings.

Table 2. Correlation (Pearson correlation and Spearman's Rank correlation separated by /) and Root Mean Square Error (RMSE) obtained after combining different methods. The default kernel function used in support vector regression was Radial Basis Function (RBF).

Regression method: features	Correlation	RMSE
Inter-annotator agreement (Hill et. al., 2014)	-/0.670	
Hill et al. (2014)	-/0.446	
LR1: WN_{Ant},$WN_{Syn,}$ 9	0.473/0.429	0.192
LR2: 10-15	0.452/0.436	0.195
LR3: WN_{Ant},$WN_{Syn,}$ 9-15	0.598/0.587	0.175
LR4: WN_{Ant},WN_{Syn}, 9-16	**0.634/0.631**	0.167
SVR1: WN_{Ant},$WN_{Syn,}$ 9	0.495/0.422	0.186
SVR2: 10-15	0.480/0.440	0.189
SVR3: WN_{Ant},$WN_{Syn,}$ 9-15	0.623/0.599	0.167
SVR4: WN_{Ant},WN_{Syn}, 9-16	**0.658/0.642**	0.159
SVR5: (Linear kernel): WN_{Ant},WN_{Syn}, 9-16	0.634/0.626	0.157
SVR6: (Polynomial kernel): WN_{Ant},WN_{Syn}, 9-16	0.536/0.607	0.187
SVR7: (Sigmoid kernel): WN_{Ant},WN_{Syn}, 9-16	0.589/0.604	0.176

We run regressions with all possible combinations of features. However, instead of showing all combinations, we present results generated with four groups of features: WordNet based methods (WN_{Ant}, $WN_{Syn,}$ 9), corpus based methods (10-15), all features except UMBC, and all features. The best result ($\rho = 0.642$) was obtained when the all features were used in support vector regression with Radial Basis Function (RBF) kernel. Moreover, we changed the kernel function in support vector regression and run with the best performing feature set when RBF kernel was used. But it did not improve the result. The results show that the best performance is obtained in both linear and support vector regressions when features from both knowledge-based and corpus-based category were used. The best result obtained using support vector regression (SVR4; $\rho = 0.642$) is significantly better than the individual performance reported in Table 1 where maximum correlation was 0.558 ($P < 0.001$).

5 Conclusion

Assessing the similarity of text is a challenging task. One might argue that similarity between two words in isolation cannot be quantified and should be defined in context. However, when humans need to judge the similarity of two things, they consider various factors and make a holistic judgment which is what the combination of different similarity methods are probably capturing.

To conclude, we presented a way of measuring the similarity of words by combining different methods. Particularly, we applied regressions to combine WordNet and different vector based methods. We found that the results produced by regressions better aligned with the human judgment compared to the individual automated methods. The best result ($\rho = 0.642$) was obtained when features including

knowledge-based and corpus-based similarity scores were used in support vector regression with Radial Basis Function (RBF) kernel. This is significantly ($P < 0.001$) better than the individual performance reported in Table 1 where maximum correlation was 0.557 and the best result obtained using the linear regression. Our similarity model's performance on Simlex-999 has reached close to the average agreement of human annotators. In the future, we would like to work on measuring semantic similarity in context and apply to measure semantic similarity of bigger texts (e.g., sentence level similarity).

Acknowledgements. This research was partially sponsored by The University of Memphis and the Institute for Education Sciences (IES) under award R305A100875 to Dr. Vasile Rus.

References

1. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and WordNet-based approaches. In: Proceedings of HLT: The Annual Conference of NAACL, pp. 19–27. Association for Computational Linguistics (2009)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
3. Burgess, C., Lund, K.: Hyperspace analog to language (hal): A general model of semantic representation. In: Proceedings of the Annual Meeting of the Psychonomic Society, vol. 12, pp. 177–210 (1995)
4. Fellbaum, C.: WordNet. Blackwell Publishing Ltd. (1998)
5. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. In: Proceedings of the 10th International Conference on World Wide Web, pp. 406–414. ACM (2001)
6. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: IJCAI, vol. 7, pp. 1606–1611 (2007)
7. Graesser, A.C., Penumatsa, P., Ventura, M., Cai, Z., Hu, X.: Using LSA in AutoTutor: Learning through mixed initiative dialogue in natural language. In: Handbook of Latent Semantic Analysis, pp. 243–262 (2007)
8. Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J.: UMBC EBIQUITY-CORE: Semantic textual similarity systems. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics, vol. 1, pp. 44–52 (2013)
9. Hill, F., Reichart, R., Korhonen, A.: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. arXiv preprint arXiv:1408.3456 (2014)
10. Hinton, G.E.: Distributed representations (1984)
11. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An Electronic Lexical Database 305, 305–332 (1998)
12. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008
13. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes 25(2-3), 259–284 (1998)

14. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database 49(2), 265–283 (1998)
15. Lee, J.H., Kim, M.H., Lee, Y.J.: Information retrieval based on conceptual distance in is-a hierarchies. Journal of Documentation 42(2), 188–207 (1989)
16. Lin, D.: An information-theoretic definition of similarity. In: ICML, vol. 98, pp. 296–304 (1998)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
18. Mohler, M., Mihalcea, R.: Text-to-text semantic similarity for automatic short answer grading. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 567–575. Association for Computational Linguistics (March 2009)
19. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 241–257. Springer, Heidelberg (2003)
20. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:: Similarity: measuring the relatedness of concepts. In: Demonstration Papers at HLT-NAACL 2004, pp. 38–41. Association for Computational Linguistics (May 2004)
21. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation
22. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Stein, B.: Overview of the 4th International Competition on Plagiarism Detection. In: CLEF (Online Working Notes/Labs/Workshop) (September 2012)
23. Resnik, P.: Using information content to evaluate semantic similarity in taxonomy. arXiv preprint cmp-lg/9511007 (1995)
24. Rus, V., Lintean, M.: A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In: Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pp. 157–162. Association for Computational Linguistics (2012)
25. Rus, V., Lintean, M.C., Banjade, R., Niraula, N. B., Stefanescu, D.: SEMILAR: The Semantic Similarity Toolkit. In: ACL (Conference System Demonstrations), pp. 163–168 (August 2013)
26. Rus, V., Niraula, N., Banjade, R.: Similarity measures based on latent dirichlet allocation. In: Gelbukh, A. (ed.) CICLing 2013, Part I. LNCS, vol. 7816, pp. 459–470. Springer, Heidelberg (2013)
27. Ştefănescu, D., Banjade, R. Rus, V.: Latent Semantic Analysis Models on Wikipedia and TASA, LREC (2014)
28. Stefanescu, D., Rus, V., Niraula, N.B., Banjade, R.: Combining Knowledge and Corpus-based Measures for Word-to-Word Similarity. In: The Twenty-Seventh International Flairs Conference (March 2014)
29. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394. Association for Computational Linguistics (July 2010)
30. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: 32nd Annual Meeting of the Association for Computational Linguistics, pp.133–138 (1994)
31. Niraula, N.B., Gautam, D., Banjade, R., Maharjan, N., Rus, V.: Combining Word Representations for Measuring Word Relatedness and Similarity. In: The Proceedings of 28th International FLAIRS Conference (2015)

Domain-Specific Semantic Relatedness from Wikipedia Structure: A Case Study in Biomedical Text

Armin Sajadi, Evangelos E. Milios, Vlado Kešelj, and Jeannette C.M. Janssen

Dalhousie University, Faculty of Computer Science,
Halifax, NS, Canada B3H 4R2
{sajadi,eem,vlado}@cs.dal.ca, janssen@mathstat.dal.ca

Abstract. Wikipedia is becoming an important knowledge source in various domain specific applications based on concept representation. This introduces the need for concrete evaluation of Wikipedia as a foundation for computing semantic relatedness between concepts. While lexical resources like WordNet cover generic English well, they are weak in their coverage of domain specific terms and named entities, which is one of the strengths of Wikipedia. Furthermore, semantic relatedness methods that rely on the hierarchical structure of a lexical resource are not directly applicable to the Wikipedia link structure, which is not hierarchical and whose links do not capture well defined semantic relationships like hyponymy.

In this paper we (1) Evaluate Wikipedia in a domain specific semantic relatedness task and demonstrate that Wikipedia based methods can be competitive with state of the art ontology based methods and distributional methods in the biomedical domain (2) Adapt and evaluate the effectiveness of bibliometric methods of various degrees of sophistication on Wikipedia (3) Propose a new graph-based method for calculating semantic relatedness that outperforms existing methods by considering some specific features of Wikipedia structure.

1 Introduction

Semantic Relatedness is a relationship between a pair of concepts. This relation can be the well known taxonomic relation (i.e., *is-a*) or any non taxonomic relation such as antonymy, meronymy (*is-a-part-of*) or domain specific relations, such as *is-treated-by* and *is-caused-by* in the biomedical domain. We address the problem of quantifying the relatedness into a real value to be used in applications such as: query expansion, word sense disambiguation, and information retrieval. A detailed review of the applications is given in [4].

Most concept-based information retrieval systems in the biomedical domain rely on ontologies to calculate relatedness. Ontologies are labor intensive to create and do not exist for most domains. Where ontologies are unavailable, an alternative is using distributional (a.k.a corpus based) methods. However, distributional methods can only be competitive if they have access to sufficiently large domain specific corpora [1,11]. Building such corpora for many domains is not trivial.

This project assesses the suitability of Wikipedia in the biomedical domain as a potential knowledge resource for semantic relatedness computation and compares it to

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 347–360, 2015.
DOI: 10.1007/978-3-319-18111-0_26

three classes of methods: (1) methods using domain specific human authored biomedical ontologies (2) state of the art distributional methods and (3) a hybrid of ontology and distributional methods that we build by using the recent developments in deep learning for distributional representation; this method outperforms previously reported corpus based methods in the literature. We focus on biomedical domain because of the availability of high-quality ontologies (MeSH, SNOMED-CT, etc.), a rich literature for extracting semantic relatedness [29,11], successful distributional methods and corpora [29,19,33] and also reliable datasets [29,27,28].

To calculate relatedness, we present a novel method that takes advantage of the prepared concept *graph* structure in Wikipedia. By focusing only on the Wikipedia graph, we implicitly assume that the only relevant phrases in the text of the concept pages are those linking to other concepts (a.k.a anchor texts). We also make an explicit assumption, that the only relevant features for representing a concept c are its neighbors in the Wikipedia graph, or in other words, those mentioned in the page associated with c and/or those which mention c in their pages. These pages are human-curated and the relevance is always explained in the text, so any attempt to use concepts not mentioned in the text disregard the explanatory structure of Wikipedia and lacks the notion of *Literary Warrant* [14]. Based on these assumptions, the intuition behind the proposed algorithm is to use the concepts in the neighborhood of a concept and rank them using the structure of the graph. For example, *September 2008* and *Clozapine* are both connected to *Schizophrenia*, where the former is just a date when some new statistics about behavioral disorders were published and should be ranked lower than the latter that is a drug to treat *Schizophrenia*.

The contributions of this research are (1) comparing Wikipedia against ontologies and distributional methods in estimating relatedness, thereby demonstrating that Wikipedia may be a suitable knowledge resource for calculating relatedness in domains lacking such high quality resources (2) motivated by the non-hierarchical structure of Wikipedia, adapting and evaluating a group of structure based graph similarity methods of various degrees of sophistication on Wikipedia, and (3) proposing a new similarity method based on the idea of ranking the neighbors and evaluating its performance.

All evaluations are performed on datasets containing pairs of terms from biomedical domain and a gold standard semantic similarity value for each pair. The results are compared with the results of the ontology based methods with well known biomedical ontologies as their resources, as well as distributional methods on well known corpora.

2 Related Work

2.1 Relatedness in General Domain

Approaches for computing semantic relatedness are traditionally categorized as distributional (a.k.a corpus based), Lexical Knowledge Resource (LKR) based (LKR can refer to dictionary, taxonomy or ontology) or hybrid if they use both at the same time. However this categorization often obscures the fact that LKR can have content (other than structure) and therefore, can play the role of a corpus as well. We use the *structure-based* label to describe methods using only the structure of a knowledge base, typically via graph-representation. Regarding methods based on WordNet only, the state of the art

ones are the Context Vector method [29], which uses the glosses, and the Personalized PageRank (PPR) method [15,1], which is based on the structure. Corpus based methods can produce competitive results using large datasets and computational resources [1]. The best reported results are obtained using hybrid methods on a web corpus and WordNet [1].

2.2 Relatedness in the Biomedical Domain

The majority of studies in relatedness in the biomedical domain concentrate on ontology based methods, as such methods benefit from availability of high quality manually curated ontologies. Two well known ontologies are Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) and Medical Subject Headings (MeSH). These resources can be accessed directly or through a framework called Unified Medical Language System (UMLS), in which these ontologies and several other terminologies are integrated. Most of the methods applied on these ontologies are successful WordNet based methods [29,11], however there are a few methods developed specifically for the biomedical domain, (e.g., [26]).

As in the general domain, distributional methods can obtain competitive results in the biomedical domain, again depending on the quality of the corpus. The distributional approaches presented in [29,19,33] show promising results on small test datasets (although the last two approaches are hybrid in fact). Some studies suggest that on larger test datasets, ontology based methods outperform distributional methods by a wide margin [11].

2.3 Relatedness from Wikipedia

Wikipedia as a resource for Semantic Relatedness has been evaluated on well known domain independent datasets. Different methods are either adaptations of ontology based methods (WikiRelate [30]), distributional (Explicit Semantic Analysis (ESA) [10]), graph-based (Wikipedia Link Measure (WLM) [25], *Visiting probability* (VP) [35]) or hybrid (WikiWalk [36]). State of the art is ESA, which also happens to be the best single-resource based method [1]. An improvement on ESA is WikiWalk [36], which combines two state of the art methods, Personalized PageRank (PPR) which is graph-based and ESA which is corpus-based. This work is similar to our approach as it ranks all the nodes in the graph according to a target node. There are two differences between WikiWalk and our method: First, WikiWalk is run globally on the whole Wikipedia graph, which is both intractable and in contrast with our idea of locality of features and the importance of neighbors, second, we use a different ranking algorithm and a distance method designed for our ranking that performs better in our experiments (a comparison of different metrics is presented in section 5.5).

One problem common in the evaluation of the mentioned systems is ignoring the fact that Wikipedia is an encyclopedia, not a dictionary, hence general words are not covered as well as domain specific terms. The only domain specific evaluation [34] is focused on text similarity rather than concrete word similarity. Besides, this evaluation method is neither on a standard dataset nor against an ontology.

Our similarity method is also related to a method [21] proposed for a different task, namely for calculation of similarity between publications based on the citation graph. This method uses the authority scores assigned by HITS [18]. Aside from the difference between domains (citation analysis and concept relatedness), there are three main other differences: first, we use the neighborhood graph only (2) we use all scores returned by HITS and not only authority scores and (3) we use a different distance calculation (a comparison of different metrics is presented in section 5.5).

3 Wikipedia Graph

A Wikipedia *page* is associated with each *concept*, so a directed graph can be obtained with nodes representing concepts and edges representing out links from a page to another.

Definition 1. *The Basic Wikipedia graph is a digraph $G_b(V_b, E_b)$ where V_b is the set of Wikipedia concepts and $(u, v) \in E_b$ iff there is a link from the page associated with u pointing to the page associated with v.*

There is a specific type of edge, called *redirect*. Redirecting denotes synonymy (for example *UK* is redirected to *United Kingdom*). We derive another graph, called *Wikipedia graph* by defining the concept of *Synonym Ring* of a node, i.e, the set of nodes synonym to it. The idea is to find a way to group synonym nodes to form a *meta* node, and using it, merge the edges between nodes to become edges between meta nodes:

Definition 2. *E_r is defined to be the set of redirections, redirection denotes synonymy:*

$$(u, v) \in E_r \implies (u, v \in E_b) \wedge u \text{ is a synonym of } v \tag{1}$$

Definition 3. *Epsilon closure of a node is the set of the nodes accessible from it by traveling along redirect links:*

$$\varepsilon(v) = \{u | ((v, u) \in E_r) \vee (\exists u_i \in \varepsilon(v) \wedge ((u_i, u) \in E_r))\} \tag{2}$$

Definition 4. *Synonym Ring of a node v is the set of nodes synonym to v*

$$sr(v) = \{v\} \cup \{u | (u \in \varepsilon(v)) \vee (\exists u_i \in sr(v) \wedge u_i \in \varepsilon(u))\} \tag{3}$$

In this paper, by referring to a node associated with a concept, we always mean the synonym ring of the node. Finally the Wikipedia graph can be defined.

Definition 5. *A Wikipedia graph $G(V, E)$ is a graph on synonym rings of the nodes of the basic graph:*

$$\begin{aligned} V &= \{v \subset \mathcal{P}(V_b) | \exists (u \in V_b) : v = sr(u)\} \\ E &= \{(u, v) \in V \times V | \exists (u' \in u, v' \in v) : (u', v') \in E_b - E_r\} \end{aligned} \tag{4}$$

We use $I(v)$ to denote to the set of in-neighbors of node v and $O(v)$ to the set of its out-neighbors. For each node v, we define three graphs: (i) $N_G[v]$: closed neighborhood graph of v, is defined to be all vertices adjacent to v (including v) and all edges connecting two such vertices in both directions, in or out of v , (ii) $N_G^-[v]$: closed in-neighborhood graph of v, is the set of vertices in $I(v)$ and all edges connecting two such vertices and (iii) $N_G^+[v]$: closed out-neighborhood graph of v, is the set of vertices in $O(v)$ and all edges connecting two such vertices.

4 Methodology

The Wikipedia graph is not hierarchical, so well known taxonomy based methods cannot be applied directly. To compute the relatedness between concepts, we start from simple and well known graph-based methods.

4.1 Bibliometrics

A straightforward approach to compare two graphs is to calculate their overlap. In our case, both graphs are *vertex-induced subgraphs* of one graph, that is the Wikipedia graph, and hence, the graph overlap is simply the vertex overlap. Using bibliometrics terminology, we can count the proportion of common incoming neighbors ($co\text{-}citation$), common outgoing neighbors ($coupling$) or common neighbors ($amsler$). Equations 5 formulates these three similarities for two given concepts a and b [7].

$$co\text{-}citation(a,b) = \frac{|I(a) \bigcap I(b)|}{|I(a) \bigcup I(b)|} \quad , \quad coupling(a,b) = \frac{|O(a) \bigcap O(b)|}{|O(a) \bigcup O(b)|}$$
$$amsler(a,b) = \frac{|I(a) \bigcup O(a)| \bigcap |I(b) \bigcup O(b)|}{|I(a) \bigcup O(a)| \bigcup |I(b) \bigcup O(b)|} \tag{5}$$

4.2 SimRank

SimRank [17] is a structural similarity method extending bibliometrics. It can be considered as a generalized version of $co\text{-}citation$ that takes into account the similarities among the citing nodes as well. In each iteration, the summation of Equation 6 is calculated for all possible pairs of concepts a and b, until it converges:

$$s_0(a,b) = 1 \text{ if } a = b, \text{ else } 0$$
$$s_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s_k(I_i(a), I_j(b)) \tag{6}$$

where $I_i(v)$, for $1 \leq i \leq |I(v)|$ denotes individual incoming neighbors of node v and C is a decay factor between 0 and 1. Similar to basic bibliometrics, SimRank can be extended to outlinks or combination of in-neighbors and out-neighbors [37], but both performed very poorly in our experiments.

SimRank should run over the entire Wikipedia graph resulting in all pairwise similarities. Due to scalability limitations and the huge size of Wikipedia, we do the recursions

over the joint-neighborhood graph of the concepts u and v, denoted by $N_G[u, v]$. This graph is made by extracting the neighborhood graphs for each concept separately and then merging the two graphs by adding the links between the nodes. Another issue is that nodes with higher number of neighbors get higher similarity. To compensate for this effect, similar to [17], the final similarity computed by SimRank is multiplied by $|I(a)|^P \times |I(b)|^P$, where P is a parameter, $P \in [0, 1]$. Unlike [17], we use both $|I(a)|$ and $|I(b)|$ to to keep the similarity symmetric.

4.3 Our Proposed Method: HITS Based Similarity

In this section we propose our similarity method which can be considered as another form of extension to basic bibliometric methods. The intuition is: *similar nodes with similar rankings in the neighborhoods of the two concepts means high relatedness*. The problem with the basic graph overlap calculation is that nodes mostly have a high number of neighbors and not all of them have the same importance. Our idea is to rank the neighbors of a node based on the role they play in its neighborhood. We use Hyperlink-Induced Topic Search (HITS) [18] to do so. HITS is a well known concept in information retrieval. It was originally developed to rank a set of search results but we use it in a similarity calculation method, referred to as *HITS-Based method* in this project.

Algorithm 1. HITS Based Similarity Computation

1: **function** $HITS\text{-}sim_{st}(a,b,st)$
Input: : a,b, two concepts; $st \in \{\text{HUB}, \text{AUTHORITY}\}$, score type
Output: : Similarity between a and b
2: $N[a] \leftarrow$ Extract a neighborhood graph for a
3: $N[b] \leftarrow$ Extract a neighborhood graph for b
4: $L[a] \leftarrow HITS(N[a], st)$ ▷ $L(a)$ will contain neighbors of a sorted by HITS
5: $L[b] \leftarrow HITS(N[b], st)$ ▷ $L(b)$ will contain neighbors of b sorted by HITS
6: $L'[a] \leftarrow append(L[a], reverse(L[b] \setminus L[a]))$
7: $L'[b] \leftarrow append(L[b], reverse(L[a] \setminus L[b]))$
8: **return** $1 - Kendall\text{-}Distance(L'[a], L'[b])$
9: **end function**
10: **function** $HITS(N,st)$
Input: : N, An adjacency matrix representing a graph; st, score type
Output: : An ordered list of vertices
11: $S \leftarrow$ Using HITS calculation, get the required score (HUB or AUTHORITY) based on st for each node in N
12: $L \leftarrow$ sort vertices of N based on S in descending order
13: **return** L
14: **end function**

To compute similarity between two concepts using this idea, we propose Algorithm 1. In steps 2 and 3, neighborhood graphs can be any of the forms

introduced in section 3. In steps 4 and 5 we use HITS algorithm to get a representative list of vertices to use as the basis of relatedness between the two concepts. HITS gives every node two scores: *hub score* and *authority score* through a recursion on the graph. So if it is run over a graph consisting of pages related to a concept (*focused graph* in HITS terminology), the final product of the algorithm is two ranked lists: authoritative pages and those which are good hubs to the authoritative pages. For the similarity measure, we can use either the hub list :$HITS\text{-}sim_{hub}(\cdot,\cdot)$ = $HITS\text{-}sim_{st}(\cdot,\cdot,\text{HUB})$ or the authority list: $HITS\text{-}sim_{aut}(\cdot,\cdot)$ = $HITS\text{-}sim_{st}(\cdot,\cdot,\text{AUTHORITY})$. HITS assigns two initial scores to each node p, *authority score*, $x^{\langle p\rangle}$, and *hub score*, $y^{\langle p\rangle}$, and uses the mutual reinforcement relation between the two scores:

1. The x score of nodes pointed to by nodes with higher y should be higher.
2. The y score of nodes pointing to nodes with higher x should be higher.

Assuming that E is the set of the edges, HITS initializes every score with 1 and performs the following iterations for each node p:

$$x^{\langle p\rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q\rangle} \tag{7}$$

$$y^{\langle p\rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q\rangle} \tag{8}$$

By normalizing these scores after each step, assuming M be the adjacency matrix, it is provable that these equations converge and the final value of X, the vector of all x scores, will be the principal eigenvector of $M^T M$ and final value of Y, the vector of all y scores, will be the principal eigenvector of MM^T [18].

Extended HITS [32] is another approach that uses the same idea of mutual reinforcement to compute node similarity in a graph. Aside from the two *hub* and *authority* lists, it extracts a third scored list of nodes that can be considered as *intermediating* between *hubs* and *authorities*. We can treat the scores assigned by Extended-HITS the same way we do with HITS in Algorithm 1. We refer to this variation by $EHITS\text{-}sim$. Using either of these scores, we end up representing each concept by a list.

Having two ordered lists after step 5, we are facing a classic ordered list comparison, which can be done by *Kendall's tau* Distance [8]. Kendall's tau works on two lists with the same elements and increases the distance for each pair of elements with different orders in the lists. In steps 6-7, we append the concepts missing in one list and present in the other one, to the list that is missing them. Our motivation in reversing the order is to penalize the similarity for any pair that one or both of them are missing in either of the lists.

Kendall's tau distance calculates the number of pairwise disagreements between the two lists. If σ_1 and σ_2 are two lists, with the same elements (in different orders) and length n, it is defined as:

$$K(\sigma_1,\sigma_2) = \frac{2}{n(n-1)} \sum_{\{i,j\}\in\mathcal{P}} \bar{K}_{i,j}(\sigma_1,\sigma_2) \tag{9}$$

where

- $\mathcal{P}$ is the set of unordered pairs of distinct elements of the lists.
- $K_{i,j}(\sigma_1, \sigma_2)$ is 0 if i and j are in the same order in both of the lists, otherwise it is 1

Hub and Authority capture two different aspects of similarity, so our final score (and our proposed method referred to by $HITS\text{-}sim$) will be a weighted average of both scores to combine them in one similarity score. To avoid parameter tuning, we always use simple average with $\lambda = 0.5$.

$$\begin{aligned} HITS\text{-}sim(a,b) = \lambda \times HITS\text{-}sim_{hub}(a,b) \\ + (1-\lambda) \times HITS\text{-}sim_{aut}(a,b) \\ \lambda \in [0,1] \end{aligned} \tag{10}$$

5 Evaluation

Evaluation of Semantic Relatedness methods consists of comparing the relatedness scores given to several pairs of concepts with human judgments. Comparison is done by calculating Spearman's rank correlations between the the two scores. For $SimRank$, the parameters are taken from the original experiments [17] ($C = 0.8$ and $P = 0.5$). We have not reported all variants of SimRank and HITS-based methods; they can work on either of the neighborhood graphs. We got better results with the following settings: $SimRank$ on $N_G^-[\cdot]$ and $HITS\text{-}sim_{aut}$, $HITS\text{-}sim_{hub}$ and $EHITS\text{-}sim$ on $N_G^-[\cdot,\cdot]$, $N_G^+[\cdot,\cdot]$ and $N_G[\cdot,\cdot]$ respectively. All experiments are based on the 20140102 dump version of Wikipedia. We use a one-tailed test on the Fisher's $z\text{-}score$ to calculate the significance of correlations [2] when comparing our results to published results of other methods, but for hybrid *word2vec* and wikipedia based methods we use the more accurate method for calculating significance, known as Zou's method for *dependent overlapping correlations* [38] (to apply this method one needs the actual scores between all pairs and knowing merely the final correlation is not enough).

5.1 Datasets

For the biomedical domain, there exist higher quality and reliable datasets of bigger sizes compared to general domain; the increased size of the datasets leads to more statistically reliable results. The datasets being used in this experiment are: (1) *Pedersen benchmark* [29], a set of 29 concepts and the most reliable dataset that biomedical comparisons are usually based on. It is the most statistically reliable subset of a set of 120 pairs, scored by three physicians and nine medical coding experts (sometimes they are reported as two separated datasets, referred to by Ped. Phys. and Ped. Coders). (2) *Mayo benchmark* [28], a set of 101 concept pairs ranked by 13 Mayo Medical Index experts. Pakhomov et al. [28] proposes a general framework to compile and evaluate semantic relatedness benchmarks; Mayo dataset is the result of that study. (3) *UMN benchmarks* [27], this is the biggest dataset scored by medical residents. It introduces two different overlapping sets of 587 and 566 concepts pairs, focusing on *similarity* (referred to by UMN Sim.) and *relatedness* (referred to by UMN Rel.).

All concepts are identified by their Concept Unique Identifier (CUI) in UMLS, we mapped them to Wikipedia pages manually, for example CUI C0038454 is mapped to *Stroke* in Wikipedia. We refer to MeSH and SNOMED-CT (through UMLS) by *sct-umls* and *mesh-umls* in the tables. Also by *umls* we refer to MeSH, SNOMED-CT and 60 other lexicons, all integrated in UMLS. We also report the results on two additional datasets with general terms, Miller and Charles (MC) [24] and WordSimilarity-353 collection [9] to make a comparison with other Wikipedia based methods possible. We used the disambiguated WordSimilarity353 for Wikipedia [25], which also covers MC dataset. The Wikipedia mapped datasets as well the software to reproduce the results can be found at: `http://www.CICLing.org/2015/data/33`

5.2 Comparison with the Relatedness Methods Based on Biomedical Ontologies

For the ontology based methods we base our comparisons on Garla et al. [11] which provides open source software and performs the experiments on publicly available datasets (another similar research providing the results for ontology based methods is McInnes et al. [22], but Garla et al. provides better results, probably due to using different versions of the incorporated ontologies).

The best results belong to three methods: LCH [20], which is a path-based method, Intrinsic Information Content (IC) based LCH (IIC-LCH) which is the same as LCH but replaces the path with IC difference between the two concepts [31], and Personalized Page Rank Based Algorithms (PPR) [1]. Garla et al. [11] includes these state of the art methods experimented on three ontologies, SNOMED-CT (o_1), MeSH (o_2), and finally, all ontologies integrated in *umls* (o_3), forming a graph with around two million nodes and 7 million relations. We only include the highest Wikipedia results in this section (which is achieved by $HITS\text{-}sim$) in Table 1. It is noticeable that our Wikipedia based method gives more improvements on the bigger datasets. This table supports our initial claims, especially the results for the largest dataset (UMN relatedness), where our Wikipedia-based method outperforms all ontology-based methods by a wide and statistically significant margin ($p\text{-}value < .001$).

Table 1. Comparison with Ontology based methods [11]: Correlation across measures and ground truth. Ontologies used are: $\mathbf{o_1}$: *sct-umls*; $\mathbf{o_2}$: *mesh-umls*; $\mathbf{o_3}$:*umls*. * significant difference with all MeSH-based and snomed-ct based methods ($p\text{-}value < .05$), † significant difference with all methods ($p\text{-}value < .001$).

Method	**Pedersen. N=29**			**Mayo N=101**			**UMN sim. N=566**			**UMN rel N=587**		
	$\mathbf{o_1}$	$\mathbf{o_2}$	$\mathbf{o_3}$	$\mathbf{o_1}$	$\mathbf{o_2}$	$\mathbf{o_3}$	$\mathbf{o_1}$	$\mathbf{o_2}$	$\mathbf{o_3}$	$\mathbf{o_1}$	$\mathbf{o_2}$	$\mathbf{o_3}$
LCH	.44	.42	.61	.03	.26	.3	.23	.25	.4	.17	.34	.34
IIC-LCH	.38	.43	.7	.3	.25	.44	.36	.29	.46	.3	.35	.39
PPR	.63	.31	.69	.17	.05	.46	.23	.18	.41	.17	.18	.33
HITS-sim	**.71**			***.52**			†**.58**			†**.51**		

5.3 Comparison with Distributional Methods

The comparison with distributional methods is given in Table 2. The state of the art corpus based methods are Context Vector [29] and Tensor Encoding [33]. However, these methods are to some extent hybrid as they both use meta thesauri to unify the text and map it to biomedical terms. Symonds et al. [33] reports Tensor encoding results for the smaller test datasets only, while Context Vector is evaluated on larger datasets as well in Garla et al. [11], from which we report the results. To have a more thorough evaluation we trained the state of the art and popular *Skip-gram* vector representation (a.k.a Word2Vec) [23] on the OHSUMED dataset [13], a collection of 348,566 references from medical journals over a five-year period (1987-1991). Tensor encoding and other distributional methods (such as [19]) are using the same corpus. We also implemented a hybrid version which first uses MetaMap [3] to map the phrases to UMLS concepts and then learns the Skip-gram model. This model gave us the best reported results on the smaller datasets which outperformed both Context Vector and Tensor Encoding methods. The Wikipedia results are still competitive on smaller datasets and significantly better on the larger ones.

Table 2. Comparison with distributional methods: Correlation across measures and ground truth. * Mayo Corpus of Clinical Notes. † Difference with $HITS\text{-}sim$ is significant ($p\text{-}value < .05$).

Method	Resources	Pedersen N=29	Mayo N=101	UMN sim. N=566	UMN rel. N=587
Vector	Mayo Corpus*+UMLS	.76	†.02	†.02	†-.13
Tensor	OHSUMED+UMLS	.76			
Word2Vec	OHSUMED	†.34	†.26	†.36	†.29
Word2Vec	OHSUMED+UMLS	**.80**	**.63**	†.39	†.39
HITS-sim	Wikipedia	.71	.52	**.58**	**.51**

5.4 Evaluating $HITS\text{-}sim$: The Effect of Ordering

A comparison of our proposed method with other Wikipedia based methods is shown in Table 3. We compare our method with WLM [25] which is the most popular structural method based on Normalized Google Distance [6] and with bibliometric graph similarity methods. From Distributional methods, we compared with CPRel[16] and ESA [10] (we report the results for both methods from [16]). Abstracting from the details, both methods generate a term-document matrix based on tfidf. Given two terms, CPRel uses the Wikipedia pages associated with the terms and calculates the cosine similarity between the two document vectors from the term-document matrix, while ESA finds the correspondent rows for the terms in the term-document matrix and calculates the cosine between the two vectors. General terms are not well covered in Wikipedia and the associated pages have a low quality. This leads to inferior results with the structure based methods. This will not affect ESA when dealing with general words (as ESA uses the text of Wikipedia as a corpus only), but on the other hand, ESA is not directly

applicable to multi-word phrases (which is the case with most Wikipedia concepts). We used the same subset ($size = 318$) of WordSim353 used with CPRel.

It is observed that $HITS\text{-}sim$ is the only method that outperforms other methods on most of the test datasets. Relatedness test datasets are limited in size and this affects the significance of the differences. In Table 3, the significant differences with $HITS\text{-}sim$ are marked. Also following [2], we calculated the weighted average of correlations on WordSimilarity-353, Pedersen and UMN-relatedness (three largest datasets with no overlap) and observed a significant difference between $HITS\text{-}sim$ and all structure based methods (*WLM*, *co-citation*, *coupling*, *amsler*, SimRank and EHITS-sim) under $p\text{-}value < .05$

Table 3. Comparison between Wikipedia based methods: Correlation across measures and ground truth. * Difference with $HITS\text{-}sim$ is significant under $p\text{-}value < .05$ † Difference with $HITS\text{-}sim$ is significant on the weighted average of WordSim353, Ped. All and UMN Rel (three largest datasets that do not share any pair) under $p\text{-}value < .05$.

Method	MC	WordSim353	Ped. Phys.	Ped. Coders	Ped. All	Mayo	UMN Sim.	UMN Rel.
ESA	.73	**.75**						
CPRel	.83	.64						
WLM†	.86	.67	.63	.69	.67	.49	**.58**	.49
Co-Citation†	.86	.67	.62	.68	.66	.47	.57	.49
Coupling†	**.90**	*.65	.61	.66	.64	*.44	*.49	*.4
Amsler†	.86	.68	.58	.66	.64	*.45	*.53	*.43
SimRank†	.79	*.51	*.56	*.55	*.55	*.39	*.45	*.39
EHITS-sim†	.84	*.62	.6	.67	.64	*.46	*.54	*.45
HITS-sim	.88	.70	**.67**	**.72**	**.71**	**.52**	**.58**	**.51**

5.5 The Effect of Distance Method

Comparison of our proposed way of incorporating Kendall's tau distance with *cosine* metric as proposed in [21], is given in Table 4. Another measure that can take into account both importance and the ratio scale of the scores given by HITS, is *Pearson* correlation. The lower performance of both *cosine* and *Pearson* is because the compared scores are the results of calculations performed on different graphs, in other words, the compared scores are in two different spaces.

6 Complexity Analysis

Regarding our proposed HITS-Based algorithms, it requires only the principal component of the neighbourhood matrix, and hence, the Power Method can be used which is very efficient with sparse matrices (linear convergence) [12]. It should be noted that calculating HITS for each concept is a one-time task; we run HITS offline and precompute the ranks of the neighbours for each node. Therefore, the complexity of Algorithm 1 depends on Kendall-tau, which can be calculated efficiently with $O(nlog(n))$

Table 4. The effect of the distance method used in Algorithm 1 for three distances: Kendall's tau (τ), Pearson (r) and $cosine$ distance (cos). Values are spearman correlation (ρ) with the gold standards.

	Pedersen			**MayoSRS**			**UMN Rel.**			**UMN Sim.**		
	τ	r	cos	τ	r	cos	τ	r	cos	τ	r	cos
ρ	.71	.57	.64	.52	.42	.52	.58	.35	.55	.51	.36	.49

operations [5]. Therefore, our algorithm has the same asymptotic complexity as basic bibliometrics.

7 Conclusion

We gave a new comparison between different algorithms for Semantic Relatedness in the biomedical domain. Our experiments demonstrates that: (1) distributional and ontology based methods can be quite competitive, and a hybrid of them improves the results. (2) using Wikipedia as a resource is comparable with the available specialized resources and often even significantly improves upon them (Tables 1 and 2). (3) our new proposed graph-based relatedness computing approach based on the HITS algorithm achieves the best correlations with human judgement as illustrated in Table 3. We chose the biomedical domain because of the availability of different ontologies and methods, which is significantly higher than any other domain.

Acknowledgments. This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Boeing Company, and Mitacs.

References

1. Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A.: A study on similarity and relatedness using distributional and wordnet-based approaches. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL 2009, Association for Computational Linguistics, Stroudsburg (2009), http://dl.acm.org/citation.cfm?id=1620754.1620758
2. Agirre, E., Cer, D., Diab, M., Gonzalez-agirre, A., Guo, W.: SEM 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In: *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics (2013)
3. Aronson, A.R., Lang, F.M.: An overview of metamap: historical perspective and recent advances. JAMIA 17(3), 229–236 (2010), http://dblp.uni-trier.de/db/journals/jamia/jamia17.html#AronsonL10
4. Budanitsky, A.: Lexical Semantic Relatedness and its Application in Natural Language Processing. Ph.D. thesis, University of Toronto, Toronto, Ontario (1999)
5. Christensen, D.: Fast algorithms for the calculation of Kendall's τ. Computational Statistics 20(1), 51–62 (2005), http://dx.doi.org/10.1007/BF02736122
6. Cilibrasi, R.L., Vitanyi, P.M.B.: The google similarity distance. IEEE Trans. on Knowl. and Data Eng. 19(3), 370–383 (2007), http://dx.doi.org/10.1109/TKDE.2007.48

7. Couto, T., Cristo, M., Gonçalves, M.A., Calado, P., Ziviani, N., Moura, E., Ribeiro-Neto, B.: A comparative study of citations and links in document classification. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2006, pp. 75–84. ACM, New York (2006), http://doi.acm.org/10.1145/1141753.1141766
8. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2003, pp. 28–36. Society for Industrial and Applied Mathematics, Philadelphia (2003)
9. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 406–414. ACM, New York (2001), http://doi.acm.org/10.1145/371920.372094
10. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, pp. 1606–1611. Morgan Kaufmann Publishers Inc., San Francisco (2007), http://dl.acm.org/citation.cfm?id=1625275.1625535
11. Garla, V., Brandt, C.: Semantic similarity in the biomedical domain: an evaluation across knowledge sources. BMC Bioinformatics 13(1), 1–13 (2012)
12. Golub, G.H., van der Vorst, H.A.: Eigenvalue computation in the 20th century. Journal of Computational and Applied Mathematics 123(1-2), 35–65 (2000); numerical Analysis 2000. Vol. III: Linear Algebra, http://www.sciencedirect.com/science/article/pii/S0377042700004131
13. Hersh, W., Buckley, C., Leone, T.J., Hickam, D.: Ohsumed: An interactive retrieval evaluation and new large test collection for research. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994, pp. 192–201. Springer-Verlag New York, Inc., New York (1994), http://dl.acm.org/citation.cfm?id=188490.188557
14. Hjrland, B.: Citation analysis: A social and dynamic approach to knowledge organization. Information Processing & Management 49(6), 1313–1325 (2013), http://linkinghub.elsevier.com/retrieve/pii/S0306457313000733
15. Hughes, T., Ramage, D.: Lexical semantic relatedness with random graph walks. In: EMNLP-CoNLL, pp. 581–589 (2007)
16. Jabeen, S., Gao, X., Andreae, P.: CPRel: Semantic relatedness computation using wikipedia based context profiles. In: Research in Computing Science, vol. 70, pp. 55–66 (2013)
17. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 538–543. ACM, New York (2002)
18. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (1999)
19. Koopman, B., Zuccon, G., Bruza, P., Sitbon, L., Lawley, M.: An evaluation of corpus-driven measures of medical concept similarity for information retrieval. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 2012, pp. 2439–2442. ACM, New York (2012), http://doi.acm.org/10.1145/2396761.2398661
20. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) pp. 305–332. MIT Press (1998)
21. Lu, W., Janssen, J., Milios, E., Japkowicz, N., Zhang, Y.: Node similarity in the citation graph. Knowledge and Information Systems 11(1), 105–129 (2007), http://dx.doi.org/10.1007/s10115-006-0023-9
22. McInnes, B.T., Pedersen, T., Pakhomov, S.V.: UMLS-Interface and UMLS-Similarity: open source software for measuring paths and semantic similarity. In: AMIA Annual Symposium Proc. 2009, pp. 431–435 (2009)

23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013), http://arxiv.org/abs/1301.3781
24. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and Cognitive Processes 6(1), 1–28 (1991)
25. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceedings of AAAI 2008 (2008)
26. Nguyen, H., Al-Mubaid, H.: New ontology-based semantic similarity measure for the biomedical domain. In: 2006 IEEE International Conference on Granular Computing, pp. 623–628 (2006)
27. Pakhomov, S., McInnes, B., Adam, T., Liu, Y., Pedersen, T., Melton, G.B.: Semantic Similarity and Relatedness between Clinical Terms: An Experimental Study. In: AMIA Annu. Symp. Proc. 2010, pp. 572–576 (2010)
28. Pakhomov, S.V.S., Pedersen, T., McInnes, B., Melton, G.B., Ruggieri, A., Chute, C.G.: Towards a framework for developing semantic relatedness reference standards. J. of Biomedical Informatics 44(2), 251–265 (2011)
29. Pedersen, T., Pakhomov, S.V., Patwardhan, S., Chute, C.G.: Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics 40(3), 288–299 (2007)
30. Ponzetto, S.P., Strube, M.: Knowledge derived from wikipedia for computing semantic relatedness. J. Artif. Intell. Res (JAIR) 30, 181–212 (2007)
31. Sánchez, D., Batet, M.: Semantic similarity estimation in the biomedical domain: An ontology-based information-theoretic perspective. J. of Biomedical Informatics 44(5), 749–759 (2011), http://dx.doi.org/10.1016/j.jbi.2011.03.013
32. Senellart, P., Blondel, V.D.: Automatic discovery of similar words. In: Berry, M.W., Castellanos, M. (eds.) Survey of Text Mining II: Clustering, Classification and Retrieval, pp. 25–44. Springer-Verlag (January 2008)
33. Symonds, M., Zuccon, G., Koopman, B., Bruza, P.D., Nguyen, A.: Semantic judgement of medical concepts: combining syntagmatic and paradigmatic information with the tensor encoding model. In: Australasian Language Technology Association Workshop (ALTA 2012). University of Otago, Dunedin (December 2012), http://eprints.qut.edu.au/54722/
34. Yang, B., Heines, J.M.: Domain-specific semantic relatedness from Wikipedia: can a course be transferred? In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, NAACL HLT 2012, pp. 35–40. Association for Computational Linguistics, Stroudsburg (2012), http://dl.acm.org/citation.cfm?id=2385736.2385744
35. Yazdani, M., Popescu-Belis, A.: Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. Artif. Intell. 194, 176–202 (2013), http://dx.doi.org/10.1016/j.artint.2012.06.004
36. Yeh, E., Ramage, D., Manning, C.D.: Wikiwalk: random walks on Wikipedia for semantic relatedness. In: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, TextGraphs-4, pp. 41–49. Association for Computational Linguistics, Stroudsburg (2009)
37. Zhao, P., Han, J., Sun, Y.: P-rank: a comprehensive structural similarity measure over information networks. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 553–562. ACM, New York (2009)
38. Zou, G.Y.: Toward using confidence intervals to compare correlations. Psychological Methods 12(4), 399–413 (2007), http://dx.doi.org/10.1037/1082-989x.12.4.399

Unsupervised Induction of Meaningful Semantic Classes through Selectional Preferences

Henry Anaya-Sánchez and Anselmo Peñas

NLP & IR Grroup, UNED,
Juan del Rosal, 16
28040 Madrid, Spain
{henry.anaya,anselmo}@lsi.uned.es

Abstract. This paper addresses the general task of semantic class learning by introducing a methodology to induce semantic classes for labeling instances of predicate arguments in an input text. The proposed methodology takes a Proposition Store as Background Knowledge Base to firstly identify a set of classes capable of representing the arguments of predicates in the store; where the classes corresponds to common nouns from the store to support interpretability. Then, it learns a selectional preference model for predicates based on tuples of classes to set up a generative model of propositions from which to perform the induction of classes. The proposed method is completely unsupervised and rely on a reference collection of unlabeled text documents used as the source of background knowledge to build the proposition store. We demonstrate our proposal on a collection of news stories. Specifically, we evaluate the learned model in the task of predicting tuples of argument instances for predicates from held-aside data.

Keywords: Semantic class learning, semantic class induction, generative model of selectional preferences.

1 Introduction

The problem of identifying semantic classes for words in Natural Language Processing (NLP) has been shown useful to address many text processing tasks, mainly in the development of systems that suffers from data scarcity or sparseness.

Although some semantic dictionaries and ontologies do exist such as WordNet [11] or DBPedia [10], their coverage is rarely complete, especially for large open classes (e.g., very specialized classes of people and objects), and they fail to integrate new knowledge. Thus, it often helps a lot to firstly learn word categories or classes from a large amount of (unlabeled) training data and then to use these categories as features in the text processing tasks.

The general task of semantic class learning, which can be broadly defined as the task of learning classes of words and their instances from text corpora, has been addressed in a variety of forms that correspond to different application

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 361–371, 2015.
DOI: 10.1007/978-3-319-18111-0_27

scenarios. Among these forms, we can find two that have been termed as *semantic class mining* [16,6,9] and *semantic class induction* [7,4]. These have to do respectively with (i) the expansion of (seed) sets of instances labeled with class information (Knowledge Base population), and with (ii) automatic annotation of individual instances with their semantic classes in the context of a particular text.

In our research, we are focused on the later. Specifically, we center on the task of providing a collection of instances in a text (namely, instances of predicate arguments) with class information about what each entity is regarding the context it appears. Eventually, our goal will be to enrich the context with properties inherited from the semantic class.

Thus, an important issue addressed in our work is that of learning an interpretable, class-based meaningful model of tuples of argument instances for predicates. By *meaningful*, we refer to a class-based model satisfying the following two properties:

- be a general enough class-based model so that it can represent any tuple of instances for the predicate, but also
- be a specific enough model so that it directly reflects the most important properties of instances that can be inherited from the textual context in which the instances occurs.

For example, in the context "*x1* throws a touchdown pass", entity *x1* should be assigned to a class entailing football *players* rather than just a generic class *person*, and more likely, *x1* should receive the class *quarterback*.

In this way, this paper proposes a new methodology to learn a (stochastic) class-based selectional preference model for predicates enabled to be used for semantic class induction. The methodology takes a Proposition Store [2,13] (i.e., a collection of propositions built from a reference collection of unlabeled text documents) as the Background Knowledge Base from which we learn the models to classify the instances in an input text.

The preference model is aimed to map each predicate to a discrete distribution of class tuples that model the stochastic generation of individual propositions; where, to support interpretability, we consider the classes to be represented by means of common nouns (specifically, nominal phrases). We assume that the classes come from a a global set of classes such that there is a subset of the global set capable of representing the collection of instances of each predicate argument.

A strong point of our proposal is that it can deal with predicates of arbitrary arity, and not only with binary predicates as usual in literature of Open Information Extraction.

Unlike existing approaches that models the generation of predicate arguments based on Probabilistic Topic Modeling [14,15](most of them modeling individual predicate arguments), our approach is based on non-latent classes represented by common nouns. By doing this, we argue for class interpretability since the latent topics inferred by traditional topic modeling approaches are hard to interpret.

On the other hand, our proposal also differs from the approach of learning interpretable entity types presented in [5], since our proposal actually models tuples of instances and classes (of arbitrary arity) and scales to broad collections of reference.

We achieve scalability by considering the functional domain of each argument to be represented by a strict subset (a small subset in practice) of all of the possible classes.

We evaluate our proposal from a collection of news. Specifically, we model the tuples of semantic classes underlying the predicate arguments in a Proposition Store built from the news texts. Then, we evaluate the obtained model on the prediction of tuples of instances predicate arguments from new input texts.

The experiments carried out show significant improvements over a (baseline) generative model of tuples of argument instances based on latent classes that is defined by means of Hierarchical Dirichlet Processes (HDP) [18].

2 The Methodology

To classify the instances of predicate arguments in an input text, our approach relies on statistical models learned for each predicate in a proposition store used as Background Knowledge Base. The proposition store can be broadly represented as a collection of propositions gathered from a reference collection of unlabeled text documents; where each proposition s in the store is an element of the form $r(a_1, \ldots, a_{arity(r)})$ such that r is a predicate, and $\forall j \in \{1, \ldots, arity(r)\}$, a_j denotes the instance of the jth predicate argument in proposition s. Function $arity$ associates each predicate to its number of arguments.

Specifically, our idea to classify instances assumes that there is finite, non-empty set of semantic classes $C = \{c_1, \ldots, c_{|C|}\}$ such that instances of each predicate argument $A_{r,i}$ (r is the predicate, and $i \in [1, arity(r)]$) can be represented by means of a (non-empty) set of classes $C_{r,i} \subseteq C$.

Then, a class-based selectional preference model Γ that maps each predicate r to a discrete probability distribution of class tuples from $C_{r,1} \times \cdots \times C_{r,arity(r)}$ is learned as the statistical model from which we preform the (inductive) classification of instances of predicate arguments.

For each predicate r, $\Gamma(r)$ is expected to globally models the stochastic generation of individual propositions with the form $s = r(a_1, \ldots, a_{arity(r)})$ as follows:

$$\begin{aligned} &\mathrm{p}(s = r(a_1, \ldots, a_{arity(r)})) = \\ &\quad = \sum_{k=1}^{|\Gamma(r)|} \mathrm{p}(z = \Gamma(r)_k) \prod_{i=1}^{arity(r)} \mathrm{p}(a_i | z[i]) \end{aligned} \tag{1}$$

where $|\Gamma(r)|$ denotes the dimension (i.e., the number of class tuples) of the probability distribution $\Gamma(r)$, $\mathrm{p}(z = \Gamma(r)_k)$ is the probability associated to kth class tuple in $\Gamma(r)$ (actually, the probability of selecting the kth class tuple to generate a proposition with predicate r), and $\mathrm{p}(a_i|z[i])$ is the conditional

probability of instance a_i given the ith class, $z[i]$, in the tuple of classes in z ($z = \Gamma(r)_k$).

For example, by considering $C_{\text{throw},2}$ = {<*pass*>, <*interception*>, <*ball*>, <*touchdown*>}, a possible definition for Γ(throw) could be:

$$\begin{cases} (\langle \textit{quarterback} \rangle, \langle \textit{pass} \rangle) & \vdash 0.54 \\ (\langle \textit{quarterback} \rangle, \langle \textit{interception} \rangle) & \vdash 0.21 \\ (\langle \textit{quarterback} \rangle, \langle \textit{ball} \rangle) & \vdash 0.07 \\ (\langle \textit{quarterback} \rangle, \langle \textit{touchdown pss} \rangle) & \vdash 0.06 \\ (\langle \textit{person} \rangle, \langle \textit{ball} \rangle) & \vdash 0.05 \\ (\langle \textit{group} \rangle, \langle \textit{ball} \rangle) & \vdash 0.04 \\ (\langle \textit{person} \rangle, \langle \textit{pass} \rangle) & \vdash 0.016 \\ (\langle \textit{group} \rangle, \langle \textit{pass} \rangle) & \vdash 0.008 \\ (\langle \textit{person} \rangle, \langle \textit{touchdown} \rangle) & \vdash 0.005 \\ (\langle \textit{person} \rangle, \langle \textit{interception} \rangle) & \vdash 0.002 \end{cases} \tag{2}$$

and in this way the probability of generating a proposition s = throw(Young, ball) would be:

$$\begin{aligned} &\text{p}(s = \text{throw(Young, ball)}) = \\ &\quad 0.54\text{p}(\text{Young}|\langle \textit{quarterback} \rangle)\text{p}(\text{ball}|\langle \textit{pass} \rangle) + \\ &\quad 0.21\text{p}(\text{Young}|\langle \textit{quarterback} \rangle)\text{p}(\text{ball}|\langle \textit{interception} \rangle) + \\ &\quad \vdots \\ &\quad 0.002\text{p}(\text{Young}|\langle \textit{person} \rangle)\text{p}(\text{ball}|\langle \textit{interception} \rangle) \end{aligned} \tag{3}$$

given a specific definition of conditional probabilities between classes and argument instances.

Then, based on Γ, we propose to classify the instances of predicate arguments in an input text, that is represented by a collection of propositions $S = s_1, \ldots, s_{|S|}$ gathered from the text, as follows. Each proposition s_i in S is represented by the form $s_i = r_i(a_{i,1}, \ldots, a_{i,arity(r_i)})$.

2.1 Inducing Classes for Instances of Predicate Arguments

We classify each argument instance $a_{i,j}$ of proposition s_i with the class $z_i[j]$ that results from labeling s_i with the tuple of classes $z_i = (z_i[1], \ldots, z_i[arity(r_i)])$, where $\langle z_i[1], z_i[2], \ldots, z_i[arity(r_i)] \rangle$ belongs to $\{\Gamma(r_i)_1, \ldots, \Gamma(r_i)_{|\Gamma(r_i)|}\}$.

Specifically, we consider the posterior distribution of class tuples defined as:

$$\begin{aligned} &\text{p}(z = \Gamma(r_i)_k | s_i) \propto \\ &\qquad \text{p}(z = \Gamma(r_i)_k) \prod_{j=1}^{arity(r_i)} \text{p}(a_{i,j} | z[j]) \end{aligned} \tag{4}$$

to label s_i with:

$$z_i = \underset{z}{\text{argmax}} \text{p}(z = \Gamma(r_i)_k | s_i) \tag{5}$$

3 Identifying Interpretable Semantic Classes for Predicate Arguments

Since our aim is to obtain an interpretable model to classify instances, we consider the set of classes C to be defined as the set of all common nouns (namely, nominal phrases) used as argument instances in the reference proposition store.

Then, we rely on [1] to identify the set of semantic classes that can represent the kth argument of predicate r, namely, $A(r,k)$ by regarding the Pointwise Mutual Information value between each class and a class-based model of the predicate argument that we estimate as follows:

$$\mathrm{p}(c|A(r,k)) \propto \sum_{c' \in C} \sum_{a \in A^*} \mathrm{p}^*(c|c')\mathrm{p}(c'|a)\mathrm{p}(a|A(r,k)) \tag{6}$$

where, $\mathrm{p}(a|A(r,k))$ is the probability of observing instance a as instance of the predicate argument $A(r,k)$, $\mathrm{p}(c'|a)$ is the conditional probability of class c' given a, and $\mathrm{p}^*(c|c')$ is a parameter of a mapping model between classes that we learn using infinite Markov chains over the initial mapping given by $\mathrm{p}(c|c') \propto \sum_{a \in A^*} \mathrm{p}(c|a)\mathrm{p}(a|c')$. A^* is the set of all instances of predicate arguments in the store.

Thus, we define the set of classes $C_{r,k}$ as:

$$C_{r,k} = \{c \in C | \mathrm{p}(c|A(r,k)) > \theta_0 \wedge \log(\mathrm{p}(c|A(r,k)/\mathrm{p}(c)) > \gamma_0\} \tag{7}$$

The factor $\mathrm{p}^*(c|c')$ in Equation 6 is aimed at obtaining a "transitive" semantic smoothing of the class posteriors given by $\{\sum_{a \in A^*} \mathrm{p}(c'|a)\mathrm{p}(a|A(r,k))\}$.

In this paper, we estimate the conditional probabilities of individual argument instances given a class using MLE by counting the associations of instances and classes in the appositions found in the reference text collection from which the proposition store is built. These statistics are complemented with counts from the association between an observed instance represented by a nominal phrase with its head noun.

In our experiments, we use $\theta_0 = 0.001$ and $\gamma_0 = 1.0$.

4 Learning the Class-Based Selectional Preference Model

To learn the class-based selectional preference model $\Gamma(r)$ for predicate r from the proposition store, we just need to infer the prior probabilities for the tuples of classes in $C_{r,1} \times \cdots \times C_{r,arity(r)}$ used in Equation 1 to encode the stochastic generation of individual propositions with the form $s = r(a_1, \ldots, a_{arity(r)})$.

To perform the inference of priors, we consider a Gibbs sampling procedure that randomly assigns a tuple of classes z to each proposition in the store with the form $s = r(a_1, \ldots, a_{arity(r)})$ according to the discrete posteriors:

$$\mathrm{p}(z = \Gamma(r)_k|s) \propto \frac{n_k + \alpha}{N + \alpha K} \prod_{i=1}^{arity(r)} \mathrm{p}(a_{j,i}|z[i]) \tag{8}$$

where n_k is the number of propositions in the store (with predicate r) labeled with the class tuple based in $\Gamma(r)_k$, N is the total number of propositions in the store with predicate r, α is a smoothing term, and K is the total number of possible class tuples in $C_{r,1} \times \cdots \times C_{r,arity(r)}$.

After performing the Gibbs sampling, the priors are defined as:

$$\mathrm{p}(z = \Gamma(r)_k) \propto n_k + \alpha \tag{9}$$

We keep only those tuples of classes that have been assigned to at least 3 propositions.

5 Experiments

In order to evaluate our proposal, we consider a collection of 30,826 New York Times articles about US football, from which we build two proposition stores: one for training (based on the first 80% of the published articles) and the other one for testing (based on the remainder articles).

The aim was to identify the semantic classes underlying each predicate argument and the to build the class-based selectional preference model for predicates.

Specifically, documents in the training set were parsed using a standard dependency parser [3,8] together with TARSQI [19], and after collapsing some syntactic dependencies following [2,13], we select the collection of 1,646,583 propositions corresponding to the top 1500 more frequent verb-based predicates (i.e., about the 90 percent of the total number of propositions in the training) to set up the proposition store of reference.

The same procedure was applied to gather propositions from the test set, but they were held-aside for testing purposes.

We applied our approach to firstly identify the classes behind each predicate argument and then obtain the models from the proposition store used for training. The obtained models were evaluated by conducting two experiments. In each experiment, we choose to compare the results obtained by our proposal to a (baseline) version of our approach produced by applying HDP [18] to learn the classes underlying each argument using latent distributions.[1]

5.1 Evaluating the Coherence of the Classes

Thus, the first experiment was aimed at measuring the coherence or degree of interpretability of identified classes. To be fair, we use in this experiment the distributions of instances obtained by modeling each predicate argument (namely, the collection of argument instances of each predicate in the training)

[1] HDP is a fully bayesian, unsupervised PTM approach that different from LDA and related (traditional) PTM approaches does not need to known the number of topics (in our case, instance distributions) to be discovered beforehand. Besides, HDP has been shown to optimize the generative approach of LDA in terms of the likelihood of predicting data.

as a mixture of the classes. For this purpose, we rely on the UMass measure of coherence as defined in [17], that in this case regards co-occurrence frequencies of instances across the predicate arguments and a positive real value ϵ to define the coherence of each distribution induced as follows:

$$coherence(c_{A_i}; n) = \sum_{u=1}^{n} \sum_{\substack{v=1 \\ l \neq u}}^{n} \log \frac{S(c_{A_i}^{(u)}, c_{A_i}^{(v)}) + \epsilon}{S(c_{A_i}^{(v)})} \tag{10}$$

where in our case $(c_{A_i}^{(1)}, .., c_{A_i}^{(n)})$ is the list of the n most frequent instances labeled with class c_{A_i}, and $S(c_{A_i}^{(u)}, c_{A_i}^{(v)})$ is the number of predicate arguments in the corpus containing with instances $c_{A_i}^{(u)}$ and $c_{A_i}^{(v)}$. Similarly, $S(c_{A_i}^{(v)})$ is the number of predicate arguments that have been instanced at least one time with instance $c_{A_i}^{(v)}$. The larger the values of this measure the better the coherence of the class.

The parameter ϵ is employed to penalize the labeling of instances that do not co-occur as argument instances. Thus, values of $\epsilon \in (0, 1)$ are used to help distinguishing between distributions (underlying the learned classes) that are semantically interpretable and distributions that are artifacts of statistical inference.

The UMass measure of coherence is intrinsic in nature. Significantly, it compute its counts from the training corpus used to train the models rather than a test corpus [17]. So that, it attempts to confirm that the models learned data known to be in the corpus. This measure has been shown to be in agreement with coherence judgments by experts [12] in PTM.

In Table 1, we show the averaged values of UMass coherence obtained by each approach. As can be seen, the greatest values of the coherence measure correspond to the distributions of instances underlying the classes learned by our approach. This directly corroborates the good performance of the proposed model to learn coherent classes of entities to semantically label the aggregates of instances. In all cases, HDP significantly performs the worst in this experiment.

Table 1. Averaged values of UMass coherence for the clustering-based distributions of instances induced by the generative models (using ϵ=1.0e-50)

Method	n=5	n=10	n=15	n=20
HDP	-217.051	-1271.62	-3665.42	-8073.6
Our proposal	-1.9693	-8.3945	-18.8997	-33.4624

To illustrate how the obtained values of UMass coherence are representative enough of actual coherent distributions of instances, we show in Table 2 the classes learned for some predicate arguments.

As can be seen, different from the approach based on HDP, our approach accurately capture the more likely meaning of each predicate argument.

Table 2. Examples of the classes identified by our approach for some predicate arguments compared to the more probable distributions of instances learned using HDP (top 10 terms are shown)

Arg.	Noun-based classes identified	HDP distributions
x win -	team, group, no., person, champion, [football,team], host, giants, defeat, 49er, opponent, defend,champion], victory, [super bowl,champion], [only,team], [other,team]	{team, group, jets, giants, defense, new york giants, offense, 49er, miami, new england patriots, ...} {person,group,player,that,team, people,coach,myself,bill parcells,it, ...} {pass, ball, yard, touchdown, goal, [field,goal],play,lead,interception,victory, ...}
- win y	game,championship,title, [national,championship], [first,game], [last,game],[football,game],bowl,victory job,[playoff,game],[straight,game],award, [consecutive,game], one,[division,title], division,[final,game],[home,game],[big,game], [championship,game],[run,game],[super,bowl], [regular-season,game],[national,title],battle	{game, season, football, drive, championship, title, career, super bowl, time, that, ...}
x pass -	touchdown, group, person, [first,touchdown]	{quarterback, vinny testaverde, receiver, kerry collins,phil simms, chad pennington, curtis martin, tiki barber, jeff hostetler, ...} {person,group,player,that,team,people,coach, myself,bill parcells,it, ...} {team, group, jets, giants, defense, new york giants,offense, 49er, miami, new england patriots, ...}
- pass y	yard, season, test, play, touchdown, record, interception, ball, situation, examination, physical, [big,play], attempt, efficiency, rush, mark, downs, [last,season], yardage, offense, completion, [total,yard], more, rusher, person, protection, one	{pass, ball, yard, touchdown, goal,[field,goal], play, lead, interception, victory, ...} {what,that,game,team,way,chance,thing,job, time,lot, ...} {game, season, football, drive, championship, title, career, super bowl, time, that, ...}
x catch -	receiver,[wide,receiver],endrookie,[draft,pick], [tight,end], person, tailback, fullback, group, [rookie,receiver],[star,receiver],[lead,receiver], camera	{quarterback, vinny testaverde, receiver, kerry collins,phil simms,chad pennington,curtis martin, tiki barber,jeff hostetler, ken o'brien} {person,group,player,that,team,people,coach, myself,bill parcells,it, ...} {pass, ball, yard, touchdown, goal,[field,goal], play, lead, interception, victory, ...}
- catch y	pass, [touchdown,pass], ball, [short,pass], [first,pass],[scoring,pass],[incomplete,pass], [long,pass],[score,pass],touchdown,[screen, pass],[more,pass],[deep,pass],that,[9-yard, pass],[third-down,pass],[8-yard,pass],[game, ball],eye, person, one, group, punt, attention	{pass, ball, yard, touchdown, goal,[field,goal], play, lead, interception, victory, ...} {person,group,player,that,team,people,coach, myself,bill parcells,it, ...}
x make -	team, group, person, that, kicker	{person,group,player,that,team,people,coach, myself,bill parcells,it, ...} {team, group, jets, giants, defense, new york giants,offense,49er,miami new england patriots, ...} {what,that,game,team,way,chance,thing,job, time,lot, ...}
- make y	play, decision, mistake,[big,play],catch, playoff,start,change,move,difference, appearance,call,offer, deal, choice, money, debut,sense,statement,score, progress,trip, one, interception	{pass, ball, yard, touchdown, goal,[field,goal], play, lead, interception, victory, ...} {what,that,game,team,way,chance,thing,job, time,lot, ...} {one, able, good, ready, all, over, sure, better, out, old, ...}

5.2 Evaluating the Generalization Performance

The second experiment was focused on evaluating the meaningfulness of the proposed model by means of predicting correct tuples of instances for the predicates. Thus, we consider measuring the difference between (i) the averaged log-likelihood of generating the propositions in the test proposition store (i.e., the held-aside data) and (ii) the averaged log-likelihood values of generating a collection of pseudo-negative samples. The aim was to measure how well our model generalized data at the same time that it keeps precision; i.e., it generates well positive samples and it is less likely to generate "negative samples".

Pseudo-negative samples were obtained for each predicate r by randomly sampling a number of tuples of instances in the training proposition store (regardless their predicates) from those ones not observed as as argument of r. The number of samples was equal to the number of propositions with predicate r in the test store.

Table 3 summarizes the results obtained in this experiment. As it is shown, the value of the difference in the case of the version based on HDP is close to 0. This suggests that this version is likely to generate both positive and random tuples of instances for each predicate, and so the model can be hardly employed to induce correct semantic classes for the instances observed in an input text.

On the other hand, the value obtained by our approach is significantly superior and is far from 0. This validate our proposal to predict correct tuples of instances and corroborates the idea of using our model for semantic class induction.

Table 3. Difference between averaged log likelihood of generating (positive) samples in the test and pseudo negative samples obtained at random. The standard deviation of log likelihood of positive samples is shown in column 'Std. dev.'.

Method	Difference	Std. dev.
HDP	0.32	0.14
Our approach	4.79	0.16

6 Conclusions

In this paper, a new methodology to induce semantic classes for labeling tuples of instances of predicate arguments in an input text has been proposed. The proposed methodology takes a Proposition Store as Background Knowledge Base to firstly identify a set of classes capable of representing the arguments of predicates in the store; where the classes corresponds to common nouns from the store to support interpretability. The set of identified classes was then used to devise a generative model of selectional preferences (based on tuples of classes) to be used as the base for the induction of classes. The proposed methodology is completely unsupervised. We demonstrate our proposal on a collection of news stories. Specifically, we evaluate our approach in the task of predicting "correct"

tuples of argument instances for predicates. Significant improvements were obtained over a (baseline) generative model of tuples of instances based on latent classes. Future work includes the application of our proposal to enrich the input texts with properties inherited from the semantic classes.

Acknowledgments. This work was partially funded by MINECO (PCIN-2013-002-C02-01) and EPSRC (EP/K017845/1) in the framework of CHIST-ERA READERS project.

References

1. Anaya-Sánchezand, H., Peñas, A.: Unsupervised learning of meaningful semantic classes for entity aggregates. In: Proceedings of IWCS 2015 (to appear, 2015)
2. Clark, P., Harrison, P.: Large-scale extraction and use of knowledge from text. In: Proceedings of the Fifth International Conference on Knowledge Capture, pp. 153–160. ACM (2009)
3. De Marneffe, M.-C., Manning, C.D.: The stanford typed dependencies representation. In: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp. 1–8 (2008)
4. Grave, E., Obozinski, G., Bach, F., et al.: Hidden markov tree models for semantic class induction. In: CoNLL-Seventeenth Conference on Computational Natural Language Learning (2013)
5. Hovy, D.: How well can we learn interpretable entity types from text? In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA, June 22-27, vol. 2: Short Papers, pp. 482–487 (2014)
6. Huang, R., Riloff, E.: Inducing domain-specific semantic class taggers from (almost) nothing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 275–285. Association for Computational Linguistics (2010)
7. Iosif, E., Tegos, A., Pangos, A., Fosler-Lussier, E., Potamianos, A.: Unsupervised combination of metrics for semantic class induction. In: IEEE Spoken Language Technology Workshop, pp. 86–89. IEEE (2006)
8. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 423–430 (2003)
9. Kozareva, Z., Riloff, E., Hovy, E.H.: Semantic class learning from the web with hyponym pattern linkage graphs. In: Proceeding of the ACL, vol. 8, pp. 1048–1056 (2008)
10. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8. ACM (2011)
11. Miller, G.: Wordnet: A lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
12. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 262–272 (2011)

13. Peñas, A., Hovy, E.: Filling knowledge gaps in text for machine reading. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 979–987. Association for Computational Linguistics (2010)
14. Ritter, A., Etzioni, O., et al.: A latent dirichlet allocation method for selectional preferences. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 424–434 (2010)
15. Séaghdha, D.O.: Latent variable models of selectional preference. In: Proceedings of the 48th Annual Meeting oF the Association for Computational Linguistics, pp. 435–444 (2010)
16. Shi, S., Zhang, H., Yuan, X., Wen, J.-R.: Corpus-based semantic class mining: distributional vs. pattern-based approaches. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 993–1001. Association for Computational Linguistics (2010)
17. Stevens, K., Kegelmeyer, P., Andrzejewski, D., Buttler, D.: 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, pp. 952–961. Association for Computational Linguistics (2012)
18. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. Journal of the American Statistical Association 101(476), 1566–1581 (2006)
19. Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S.B., Littman, J., Rumshisky, A., Phillips, J., Pustejovsky, J.: Automating temporal annotation with tarsqi. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, pp. 81–84 (2005)

Hypernym Extraction: Combining Machine-Learning and Dependency Grammar

Luis Espinosa-Anke, Francesco Ronzano, and Horacio Saggion

TALN - Universitat Pompeu Fabra
C/Tànger, 122-134, 08018 Barcelona
{luis.espinosa,francesco.ronzano,horacio.saggion}@upf.edu

Abstract. Hypernym extraction is a crucial task for semantically motivated NLP tasks such as taxonomy and ontology learning, textual entailment or paraphrase identification. In this paper, we describe an approach to hypernym extraction from textual definitions, where machine-learning and post-classification refinement rules are combined. Our best-performing configuration shows competitive results compared to state-of-the-art systems in a well-known benchmarking dataset. The quality of our features is measured by combining them in different feature sets and by ranking them by their Information Gain score. Our experiments confirm that both syntactic and definitional information play a crucial role in the hypernym extraction task.

1 Introduction

Hypernym Extraction is the task to identify (hyponym, hypernym) relations in naturally-occurring text. For example, given the sentence "A mosque is a place of worship for followers of Islam", the objective is formalize an *is-a* relation between "mosque" and "place of worship". Such task is important for structuring knowledge hierarchically [1]. It is an appealing task in NLP applications such as Named Entity Recognition [2], Query Refinement [3], Image Classification [4], Taxonomy Learning [5], Question Answering [6], Automatic Glossary Construction [7], Ontology Learning [5] or Textual Entailment [8]. Two clear examples of its importance are: (1) The WordNet hierarchy [9], where senses are organized according to "is-a" relations, and (2) The Wikipedia BiTaxonomy Project [10], which produced a *taxonomized* version of Wikipedia, and which is based on a first step on Definition Parsing and Hypernym Extraction.

In this paper we present a set of experiments for hypernym extraction and report results that outperform state-of-the art systems in the WCL (Word-Class Lattices) dataset, a well-known benchmarking dataset of textual definitions from Wikipedia where term and hypernym are manually annotated [11]. We cast our approach as a sequential classification task where, for each word in a definition, the goal is to predict whether it is at the beginning, outside or inside a hypernym (which can be a single or a multiword phrase).

The main contribution of our paper is a set of experiments over a standard benchmarking dataset for hypernym extraction achieving state-of-the-art performance, by combining linguistic, definitional and graph-based information.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 372–383, 2015.
DOI: 10.1007/978-3-319-18111-0_28

The remainder of this paper is structured as follows: Section 2 reviews prominent work carried out in this area; Section 3 describes the linguistic motivation behind this work; Section 3.2 details the features and feature sets used in our experiments; Section 4 shows (1) a comparative evaluation across feature sets, (2) a comparative evaluation with results reported in previous work and (3) a feature relevance discussion; and Section 5 summarizes this article and outlines directions for future work.

2 Background

Textual patterns constitute the backbone of the earliest works in inducing semantic relations between words [8]. Examples widely referred to in the literature include Hearst's lexical patterns (such as "NP and other NP") [12]. Moreover, [13] propose to automatically acquire a vast large number of lexico-syntactic patterns and apply them to the newswire domain. Another well-known example is the use of Robust Minimal Recursion Semantics for semantic pattern matching [14].

In general, the literature agrees on the fact that semantic relations like hypernymy show enough variability to make pure pattern-based approaches inefficient since these patterns are either noisy by nature, as the case of *is a*, or too domain-specific and therefore impossible to generalize across domains or genres.

For this reason, machine-learning and more recently purely distributional approaches have contributed to the task of hypernym discovery. Among the former, the system described by [11] learns generalized lexico-syntactic patterns which are used to maximize the score of candidate definition sentences and, within definitions, hypernymic phrases. Moreover, [15] explored the role of syntactic dependencies as features for an SVM-based classifier. This last method is conceptually similar to ours since raw text is modelled in terms of linguistic dependencies. We extend their approach by exploiting definitional and graph-based information, which contribute to improving the performance of the system.

Distributional approaches are also becoming increasingly popular. For example, [1] describe a hypernym-discovery system for Chinese based on the notion of word-embeddings, i.e. the observation that semantically related words have common contexts at different window sizes. They propose to train a *Skip-gram* and a *CBOW* model following [16], where they take into account the embedding offsets between hyponym-hypernym pairs, and from there a projection training is designed in order to find the best hypernym for a given hyponym.

On the other hand, [8] describe a set of experiments in which they explore the veracity of the *Distributional Inclusion Hypothesis*, which states that specific terms appear in distributional contexts that are a subset of more general but related distributional contexts of more general words.

3 Modelling the Data

In the linguistic theory of Dependency Grammar, a syntactic structure is described by the distribution of lexical elements linked by asymmetrical relations called dependencies [17]. One of the main characteristics is that, unlike constituent structures, a dependency tree has no phrasal nodes. Moreover, the dependency representations provide

a direct encoding of predicate-argument structures, and the relations between units in a dependency tree are bilexical, i.e. they constitute binary (head, argument) relations [18]. Finally, in a dependency parse tree, most informative nodes (like the subject or the direct object of the sentence) are likely to be closer to the root node (main verb of the sentence). This means that (1) long-distance relations can be safely captured in a parse tree regardless of the number of modifiers that precede a target node (e.g. (subject, verb, object) relations), and (2) in definitions, tree-traversal algorithms can be easily implemented for skipping over-generalizing hypernyms (e.g. "class", "kind" or "type") as they are likely to appear near the main verb of the sentence, e.g. "X is a type of Y".

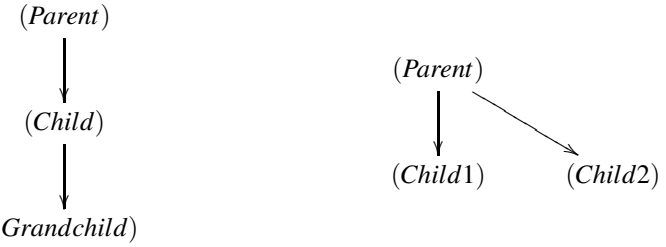
As mentioned before, and building up on previous work that exploits dependency parsing for Hypernym Extraction [10,15], we design a set of features that represent a sentence in terms of dependency relations among its lexical units.

3.1 Syntactic Motivation

We perform our experiments on the WCL dataset. This dataset is a subset of Wikipedia, where textual definitions and additional information are manually annotated. Such information, as described in [19], refers to: (1) The *definiendum*, i.e. concept that is being defined; (2) The *definitor*, i.e. the verb phrase to introduce the definition; (3) *definiens*, i.e. genus or phrase that contains the hypernym; and (4) *rest*, i.e. the rest of the sentence containing a definition. For simplicity, henceforth we refer to *definiens* as the union between *genus* and *rest*. A sample definition is illustrated below (see parse tree in Figure 1):

Sample Definition: "An <*term*> abbreviation </*term*> is a shortened form of a <*hyp*> word </*hyp*> or <*hyp*> phrase </*hyp*>."

Firstly, we apply a dependency parser [20] to the WCL dataset and extract, for each sentence, all its subtrees with the following shapes:



Each node can either include surface form information, part of speech, the dependency relation of such node with its head, or a combination of any of the former[1]. We hypothesize that the encyclopedic genre is consistent enough as to be able to draw syntactic generalizations by firstly looking at its most recurrent patterns.

The representativeness of the two shapes described above in terms of encyclopedic language is very high. For example, the *is(Verb, Root)→in(Prep, Loc)→(Noun, PMOD)* amounts to almost 20% of the whole corpus[2]. In addition, over 98% of the definitions in

[1] For the remainder of the paper, we denote *s* as surface form, *p* as part of speech, and *d* as dependency relation.

[2] We denote syntactic dependencies as arrows (*head→governor*).

such dataset have one word with *PRD* syntactic function, and we found over 850 cases where the PRD token was a direct dependent of the Root verb, and was the first word of a manually tagged hypernym: this means that 46% of the (*term*,*hypernym*) relations in this dataset would be extracted applying a simple mapping rule. While this would introduce an undesirable amount of noise, it suggests that the common assumption that *textual definitions show a high syntactic variability* [6,11] depends on what we actually consider to be language variability, and the genre and domain to which the document or corpus belongs to. For this specific case (i.e. Wikipedia), there seems to be a fairly high syntactic consistence.

Having justified our data modelling choice, the next section describes the features we designed for informing our classifier.

3.2 Experimental Setup

What follows is a description of the features used to train our model. We can cluster them in three main groups, namely: Linguistic features (1-3); definitional features (4) and graph-based featuers (6-8). Our motivation for introducing graph-based features over the parse tree is the following: We hypothesize that a hypernym might be described in terms of the popularity of its word or phrase in the syntactic tree (computed in terms of adjacent edges), its children at several levels of depth, or its salience with regard to its frequency in informative subtrees like $SBJ \leftarrow ROOT \rightarrow PRD$. However, as our experiments reveal, while these features might be effectively used for Definition Extraction [21], only one out of four seems to contribute to the hypernym extraction task when a model already includes linguistic and definitional information.

1. **Surface form (*surface*) and lemma (*lemma*)**: Normalized (lower-case) surface form and lemma. Note that unlike the experiments shown in [22,23,15], we do not generalize the definiendum to a wildcard (TARGET or TERM). We argue that in a real-world scenario one does not necessarily know which is the definiendum term, and thus removing this information also contributes to a less biased classifier. Rather, we use this information as a feature in order to assess its contribution to the learning process.

2. **Part of Speech (*pos*)**: The part of speech of the current word

3. **Head Id (*headID*) and Dependency Relation (*depen*)**: These two features refer to the syntactic function of the current word and the unique identifier of its governor or head. For example, subject (SBJ), object (OBJ), predicative (PRD) or nominal modifier (NMOD).

4. **Definiendum (*term*) and definiens (*def-ndef*)**: Whether the word is a definiendum term (i.e. it matches exactly the Wikipiedia page title to which the text snippet belongs to), and whether such word is part of the definiens. We apply a simple heuristic rule that tags all words after the first verb of the sentence as definiens.

5. **PageRank (*p-rank*)**: We compute the popularity of a node in a sentence with the PageRank algorithm. To attain this, we use an off-the-shelf Python library: NetworkX [24].

6. **Node Outdegree (*outdegree*)**: The out-degree of a node in a syntactic dependency tree is equal to the number of dependents.

7. **Morphosyntactic chains (*chains*)**: We extract all children of a node recursively until we reach the tree leaves in breadth-first fashion. For each node, we extract part-of-speech and dependency relation. This feature is a string that represents such path. While this approach is inspired by previous work on Semantic Role Labelling [25], ours differs in that we also include the dependency information.

8. **Syntactic Salience (*syntS*)**: In addition to the above features, we are interested in a more general metric to assess the extent to which a word and its associated linguistic information describes a textual genre. Motivated by the fact that in textual definitions not only are hypernyms likely to appear, but they show syntactic regularities, we count how many times a word is part of the most frequent subtrees in the dataset taking into consideration different ranges of linguistic information (from only the word's surface form to subtrees including the word's surface form, part-of-speech and syntactic funtion).

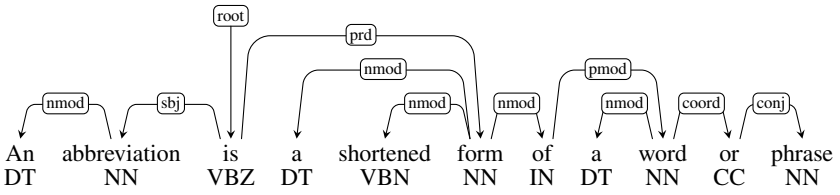


Fig. 1. Dependency parse tree of a textual definition

Numeric features such as node degree, pagerank or syntactic salience are discretized, i.e. within a range between the smallest and highest score, each value is assigned a discrete type between 1 and 10. This coarse-grained set of attributes allows us to understand better each feature's effect in the learning process and perform more sensible error analysis.

Having prepared our sets of features, these are used for training and testing a Conditional Random Fields (CRF) [26] classifier using CRF++[3]. Given the inherent ability of CRF for learning prior and posterior contextual information in a sequential classification task, we design three experiments where three context windows are considered: [-1,1], [-2,2] and [-3,3]. For each window, we design feature sets incrementally adding one feature at a time (see in Table 1 a matrix outlining all the feature sets used

[3] `https://code.google.com/p/crfpp/`

Table 1. Different feature sets adding one feature at a time

	surface	lemma	pos	headID	depen	def-ndef	term	p-rank	outdegree	chains	syntS
FeatSet1	x										
FeatSet2	x	x									
FeatSet3	x	x	x								
FeatSet4	x	x	x	x							
FeatSet5	x	x	x	x	x						
FeatSet6	x	x	x	x	x	x					
FeatSet7	x	x	x	x	x	x	x				
FeatSet8	x	x	x	x	x	x	x	x			
FeatSet9	x	x	x	x	x	x	x	x	x		
FeatSet10	x	x	x	x	x	x	x	x	x	x	
FeatSet11	x	x	x	x	x	x	x	x	x	x	x

in our experiments). The scores reported in this paper are derived from 10-fold cross validation.

3.3 Recall-Boosting Heuristics

After manually inspecting the output of the classifier, we observe that there are cases in which the discrepancy between the predicted label and the gold standard can be at questioned. In fact, [15] mention issues derived from the complexity of what actually constitutes a valid hypernym in a textual definition and its effect on the quality of the annotation of the WCL dataset. Among others, they refer to incorrect relationships, e.g. incorrectly annotating a meronym as a hypernym, or inconsistent modifier attachment, e.g. cases where the same modifier attached to two semantically-related concepts is sometimes included as part of a multiword hypernymic phrase, and others not.

This motivated a post-classification heuristic inspired by [27] consisting in a set of rules for label-switching. Let token_i be a word classified as not being part of a hypernymic phrase (O), we perform the label-switching step replacing its current label with either B, i.e. at the beginning of a hypernym phrase, or I, i.e. inside a hypernym phrase, yielding token_i^{update}. The following conditions are considered:

$$\text{token}_i^{update} = \begin{cases} \text{B} & \text{if } P(\text{token}_i) = \text{B} > \theta \wedge P(\text{token}_i) = \text{B} > P(\text{token}_i) = \text{I} \\ \text{I} & \text{if } P(\text{token}_i) = \text{I} > \theta \wedge P(\text{token}_i) = \text{I} > P(\text{token}_i) = \text{B} \\ \text{B} & \text{if } P(\text{token}_i) = \text{O} < \lambda \wedge \text{token}_i^{Synt} = \text{PRD} \end{cases}$$

Where token_i^{Synt} refers to the syntactic function of the word token_i, and where θ and λ are constants empirically set to .35 and .8 respectively after experimenting with several thresholds and inspecting manually the resulting classification.

These heuristics contribute to increase F-Score in feature sets 1 and 2 when considering [-1,1] contexts. Likewise, F-Score also improves after this step in feature sets 1, 2 and 3 when considering [-2,2] and [-3,3] contexts. In many configurations, recall improves almost 10 points, and while in strict comparison against gold standard the

drop in precision affects negatively the overall F-Score in the majority of feature sets considered, we found that in some cases our greedier approach detected a better hypernym than the one manually annotated in the gold standard. Let us look at the following sample definition:

> "An abzyme (from antibody and enzyme), also called catmab (from catalytic monoclonal antibody), is a monoclonal antibody with catalytic activiy"

In the manually annotated dataset, the hypernym is "antibody", and in the majority of our experiments our algorithm identifies "monoclonal antibody", thus producing a false positive in our word-level evaluation. However, it is not clear that "antibody" is a better hypernym for "abzyme" than "monoclonal antibody". In fact, there is a Wikipedia entry for "monoclonal antibody"[4], but not for "important antibody", for instance, which suggests that the prediction of our algorithm is correct since "monoclonal" is not a property of "antibody" but rather defines a monosemic type of antibody.

4 Evaluation

4.1 Results and Discussion

We evaluated at token-level in terms of Precision, Recall and F-Measure by adding one feature at a time to the CRF-trained model. These results are shown in Table 2. Four main conclusions can be drawn: (1) Word-level morphosyntactic features are highly informative in the encyclopedic genre (see the boost in performance after these features are added to the model), which reinforces our intuition that syntactic structures do follow certain patterns and show regularities that can be exploited; (2) The best-performing model (highest F-Score) is *FeatSet8*, which includes all linguistic features, definitional information, and page-rank; (3) Unsurprisingly, the best performing models for each feature set are those including the largest context window ([-3,3]); and (4) Recall-Boosting post-classification rules increase F-Score only in the most basic feature sets. We provide further discussion on feature relevance in Section 4.2.

Finally, we compared our best-performing model with existing state-of-the-art systems reported in the literature. Firstly, the Word-Class Lattices algorithm [22], and secondly an approach conceptually similar to ours that also modelled the problem in terms of syntactic dependencies [15] (Table 3).

4.2 Information Gain

Information Gain measures the decrease in entropy when the feature is present vs. absent [28]. We rank our features according to their $score(f, ctx, i)$, where f_i is a token-level feature, ctx refers to the context window to which it is applied, and i is the index of the current token (i.e. its current iteration). We use the machine-learning toolkit Weka

[4] `http://en.wikipedia.org/wiki/Monoclonal_antibody`

Table 2. Performance of our CRF-trained model at three different context windows ([1:1], [2:2] and [3:3]). We include results before applying the post-classification-heuristic (DefConf) and after (Boosted). We observe the best performance when only linguistic and definitional information is considered.

		DefConf-1:1	DefConf-2:2	DefConf-3:3	Boosted-1:1	Boosted-2:2	Boosted-3:3
	P	48.51	65.22	70.33	30.35	40.22	46.46
*FeatSet*1	R	31.96	41.45	48.34	65.44	72.06	75.23
	F	38.49	50.64	57.25	41.43	51.6	57.41
	P	49.36	61.87	66.55	32.12	41.77	47.84
*FeatSet*2	R	33.92	44.33	51.13	64.52	71.26	74.27
	F	40.17	51.58	57.79	42.85	52.66	58.18
	P	64.93	67.58	72.65	41.98	49.38	55.32
*FeatSet*3	R	33.17	47.23	56.62	64.68	71.34	75.36
	F	43.85	55.54	63.31	50.86	58.34	63.79
	P	70.32	72.41	74.32	48.05	53.2	58.47
*FeatSet*4	R	44.98	55.37	60.87	70.07	74.63	76.37
	F	54.8	62.71	66.89	56.99	62.1	66.22
	P	76.04	75.85	76.17	56.03	58.67	62.05
*FeatSet*5	R	54.33	61.52	64.73	74.68	76.86	78.49
	F	63.34	67.88	69.94	64.01	66.51	69.31
	P	80.19	82.99	**84.22**	62.44	68.14	73.08
*FeatSet*6	R	63.26	72.04	75.69	79.85	82.42	**84.99**
	F	70.68	77.12	79.71	70.04	74.59	78.58
	P	80.08	83.05	84.15	62	68.43	73.25
*FeatSet*7	R	63.15	72.04	75.51	79.57	82.47	84.96
	F	70.57	77.13	79.58	69.66	74.77	78.67
	P	80.11	82.56	84.01	62.67	68.34	72.59
*FeatSet*8	R	63.47	72.02	76.12	79.68	82.27	84.82
	F	70.79	76.91	**79.85**	70.13	74.64	78.22
	P	79.94	82.31	83.82	62.01	68.04	72.44
*FeatSet*9	R	63.68	72.06	75.94	79.58	82.26	84.64
	F	70.86	76.82	79.66	69.67	74.46	78.06
	P	79.6	81.86	83.6	62.4	68.64	72.71
*FeatSet*10	R	63.86	71.35	75.74	79.02	81.69	84.51
	F	70.85	76.23	79.47	69.7	74.59	78.15
	P	79.72	81.87	83.43	62.69	68.7	73.1
*FeatSet*11	R	64.48	71.62	75.36	79.22	82.13	84.16
	F	71.28	76.03	79.17	69.94	74.81	78.22

Table 3. Comparative Evaluation between our best performing model (FeatureSet8 with no post-classification heuristics) and the results reported in [22] and [15]

	Precision	Recall	F-Score
N&V WCL-1	77	42.09	54.42
N&V WCL-3	78.58	60.74	68.56
B&DiC	83.05	68.64	75.16
Our Approach	**84.01**	**76.12**	**79.85**

Table 4. Selected best features for Hypernym Extraction. Each feature reads as follows: $featureName$Position=value, where Position refers to the context in which appears at the current iteration. For instance, Position=-1 refers to one word before the word at the current iteration.

Rank	**Feature**	**InfoGain**
1	deprelPosition0=PRD	0.0682345
2	posPosition0=nn	0.0538957
3	deprelPosition-1=NMOD	0.0517277
4	defnodefPositiond0=def	0.0349189
5	defnodefPosition0=nodef	0.0349189
6	defnodefPosition1=def	0.0349189
7	headIDPosition-1	0.0320474
8	deprelPosition-2=ROOT	0.0315236
9	defnodefPosition+1=nodef	0.0300525
10	defnodefPosition-3=nodef	0.0300255
24	chainsPosition0=dt_NMOD&nnp_SBJ	0.0182301

[29]. Looking at the best features in our model (Table 4), we can conclude the following[5]: (1) Hypernym extraction algorithms improve by a huge margin if provided with syntactic information; (2) Previous work has demonstrated improvement in the task of Definition Extraction by informing the classifier with terminological information [23]. This seems to hold the other way round as well; (3) We also observe an interesting set of features clumped together with the same value and the same Information Gain score. These are *no_value* feature scores, which means that the context specified (e.g. $i = -1$)

[5] The full set of features and their Information Gain rank can be accessed at: `https://www.dropbox.com/s/d8er9jvgjz2dqo8/infogain_syntsal.txt?dl=0`. There are 2111 features with non-zero Information Gain score.

is null due to the current iteration being at the beginning or end of the sentence. This might point to hypernyms being consistently mentioned at a certain position in a sentence; (4) the discretization of our numeric values might have been too coarse-grained for being discriminative enough in a classification task. Finally, (5) After looking at the last row in Table 4, we observe the highest graph-based ranking feature (in position 24) referring to the fact that a word has a child with NNP part-of-speech and dependency relation SBJ.

5 Conclusions and Future Work

We have described a set of experiments on hypernym extraction from textual definitions in the WCL dataset. We experimented with linguistic, definitional and graph-based features which operated over the sentence parse tree. Our best model achieves competitive results in comparison with existing approaches on the same dataset. The experiments carried out also showed that linguistic and definitional information are by far the most important features in our configuration, and only few exceptions among the graph-based features can be considered informative.

Our main conclusions can be summarized as follows: (1) Hypernym extraction from textual definitions benefits significantly from syntactic and definitional information; (2) Recall-boosting heuristics contribute to increase the overall F-Score in configurations that considered smaller context windows; and (3) Graph-based features have limited discriminative power for this task.

The approach presented in this paper to hypernym extraction in textual definitions opens several avenues for future work. For example, we would like to draw statistics to measure accurately how many of the false positives in which our approach incurred after applying the Recall-Boosting heuristics could be correct hypernyms by looking at generic encyclopedias or domain-specific knowledge bases. Also, since the contribution of graph-based features was very limited, we would like to explore with finer-grained discretization heuristics as well as with the raw numeric values. Finally, it would be interesting to test our approach on other large datasets, such as WiBi [10] or the Linked Hypernyms Dataset [30].

Acknowledgments. We would like to express our gratitude to the anonymous reviewers for their helpful comments. This work is partially funded by the SKATER project, TIN2012-38584-C06-03, Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, España; and Dr. Inventor (FP7-ICT-2013.8.1 611383), programa Ramón y Cajal 2009 (RYC-2009-04291).

References

1. Fu, R., Guo, J., Qin, B., Che, W., Wang, H., Liu, T.: Learning semantic hierarchies via word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol 1: Long Papers. Association for Computational Linguistics, pp. 1199–1209 (2014)

2. Kazama, J., Torisawa, K.: Exploiting wikipedia as external knowledge for named entity recognition. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 698–707 (2007)
3. Chandramouli, K., Kliegr, T., Nemrava, J., Svátek, V., Izquierdo, E.: Query refinement and user relevance feedback for contextualized image retrieval. In: Proceedings of the 5th International Conference on Visual Information Engineering (2008)
4. Kliegr, T., Chandramouli, K., Nemrava, J., Svatek, V., Izquierdo, E.: Combining image captions and visual analysis for image concept classification. In: Proceedings of the 9th International Workshop on Multimedia Data Mining: Held in Conjunction with the ACM SIGKDD, pp. 8–17. ACM (2008)
5. Navigli, R., Velardi, P., Faralli, S.: A graph-based algorithm for inducing lexical taxonomies from scratch. In: IJCAI 2011, pp. 1872–1877 (2011)
6. Saggion, H., Gaizauskas, R.: Mining on-line sources for definition knowledge. In: 17th FLAIRS, Miami Bearch, Florida, pp. 45–52 (2004)
7. Muresan, A., Klavans, J.: A method for automatically building and evaluating dictionary resources. In: Proceedings of the Language Resources and Evaluation Conference, LREC. European Language Resources Association (2002)
8. Roller, S., Erk, K., Boleda, G.: Inclusive yet selective: Supervised distributional hypernymy detection. In: Proceedings of the Twenty Fifth International Conference on Computational Linguistics, COLING 2014, Dublin, Ireland, pp. 1025–1036 (2014)
9. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38, 39–41 (1995)
10. Flati, T., Vannella, D., Pasini, T., Navigli, R.: Two is bigger (and better) than one: the wikipedia bitaxonomy project. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol 1: Long Papers. Association for Computational Linguistics, pp. 945–955 (2014)
11. Navigli, R., Velardi, P., Ruiz-Martínez, J.M.: An annotated dataset for extracting definitions and hypernyms from the web. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC 2010. Language Resources Association (ELRA), Valletta (2010)
12. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics, vol. 2, pp. 539–545. Association for Computational Linguistics (1992)
13. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. Advances in Neural Information Processing Systems 17 (2004)
14. Herbelot, A., Copestake, A.: Acquiring ontological relationships from wikipedia using rmrs. In: Proceedings of Workshop on Web Content Mining with Human Language Technologies, ISWC 2006. Citeseer (2006)
15. Boella, G., Di Caro, L., Ruggeri, A., Robaldo, L.: Learning from syntax generalizations for automatic semantic annotation. Journal of Intelligent Information Systems, 1–16 (2014)
16. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL, pp. 746–751. Citeseer (2013)
17. Nivre, J.: Dependency grammar and dependency parsing. Technical reporut, Växjö University (2005)
18. Ivanova, A., Oepen, S., Dridan, R., Flickinger, D., Øvrelid, L.: On different approaches to syntactic analysis into bi-lexical dependencies an empirical comparison of direct, pcfg-based, and hpsg-based parsers. In: Proceedings of the 13th International Conference on Parsing Technologies, pp. 63–72 (2013)
19. Storrer, A., Wellinghoff, S.: Automated detection and annotation of term definitions in German text corpora. In: Conference on Language Resources and Evaluation, LREC (2006)

20. Bohnet, B.: Very high accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 89–97. Association for Computational Linguistics, Stroudsburg (2010)
21. Espinosa-Anke, L., Saggion, H.: Applying dependency relations to definition extraction. In: Métais, E., Roche, M., Teisseire, M. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 8455, pp. 63–74. Springer, Heidelberg (2014)
22. Navigli, R., Velardi, P.: Learning word-class lattices for definition and hypernym extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 1318–1327. Association for Computational Linguistics, Stroudsburg (2010)
23. Jin, Y., Kan, M.Y., Ng, J.P., He, X.: Mining scientific terms and their definitions: A study of the ACL anthology. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 780–790. Association for Computational Linguistics, Seattle (2013)
24. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy 2008), Pasadena, CA, USA, pp. 11–15 (2008)
25. Hacioglu, K.: Semantic role labeling using dependency trees. In: International Conference on Computional Linguistics (COLING). Association for Computational Linguistics, Stroudsburg (2004)
26. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
27. Cai, P., Luo, H., Zhou, A.: Named entity recognition in italian using crf. In: Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy (2009)
28. Forman, G.: An extensive empirical study of feature selection metrics for text classification. The Journal of Machine Learning Research 3, 1289–1305 (2003)
29. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)
30. Kliegr, T.: Linked hypernyms: Enriching dbpedia with targeted hypernym discovery. Web Semantics: Science, Services and Agents on the World Wide Web (2014)

Arabic Event Detection in Social Media

Nasser Alsaedi and Pete Burnap

Cardiff School of Computer Science and Informatics, Cardiff University, UK
{N.M.Alsaedi,P.Burnap}@cs.cardiff.ac.uk

Abstract. Event detection is a concept that is crucial to the assurance of public safety surrounding real-world events. Decision makers use information from a range of terrestrial and online sources to help inform decisions that enable them to develop policies and react appropriately to events as they unfold. One such source of online information is social media. Twitter, as a form of social media, is a popular micro-blogging web application serving hundreds of millions of users. User-generated content can be utilized as a rich source of information to identify real-world events. In this paper, we present a novel detection framework for identifying such events, with a focus on 'disruptive' events using Twitter data. The approach is based on five steps; data collection, pre-processing, classification, clustering and summarization. We use a Naïve Bayes classification model and an Online Clustering method to validate our model over multiple real-world data sets. To the best of our knowledge, this study is the first effort to identify real-world events in Arabic from social media.

Keywords: Text mining, Information Extraction, Classification, Online-Clustering, Machine Learning, Event detection.

1 Introduction

In the recent years, microblogging, as a form of social media, has rapidly grown in popularity as a mechanism for expressing opinions, broadcasting news and supporting interaction between people. One of the most representative examples is Twitter, which allows users to publish short tweets (messages within a 140-character limit) about any subject, including commentary on real-world events. Events can be community-specific, such as local gatherings, or can be wide-reaching national or even international level events. At an international level, people use social media to comment on events such as presidential elections, health pandemics, natural and man-made disasters, and major sport events as they are happening, and even before mainstream media release information about the event [8, 11, 14].

Wenwen-Dou defined an event on social media as:

"*An occurrence causing change in the volume of text data that discusses the associated topic at a specific time*." [20].

Here, we use the same definition where events have different degrees of importance causing the different "volume change" when discussed in social media platforms. Thus, an event can be characterized by a 'bursty' increase in particular

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 384–401, 2015.
DOI: 10.1007/978-3-319-18111-0_29

terms or words at some point in time. In this paper we are particularly interested in whether we can identify *disruptive* events using social media, and distinguish between these and other events. Examples of such events include protests, terrorist attacks, transport loss and crimes. In [1] disruptive events in the context of social media are defined as:

"*An event that interferes the achieving of the objective of an event or interrupts ordinary event routine. It may occur over the course of one or several days, causing disorder, destabilizing securities and may results in a displacement or discontinuity*."

Our objective is therefore to identify these events so that disruption, security issues, and disorder, can be managed and minimized. As events are typically 'bursty' topics of interest, they can lead to an instant and voluminous social reaction. Identifying events using the public reaction published openly via social media presents a number of benefits for planning and response purposes, but also many challenges. These challenges include: First, the speed and volume at which data arrives, where tweets arrive continuously in chronological order. Second, the nature of "live" events produces a continuously changing dynamic corpus. Third, the significant amount of "noise" presented in the stream constitutes around 40% of all tweets, which have been reported as pointless "babbles" [3] or spam. Finally, each tweet is short (140 characters), which means they often lack the context that would assist text analysis.

The main task that we tackle in this paper is the ability to develop an algorithm to detect disruptive events and test the applicability of our algorithm to Arabic content posted to Twitter. Arabic is a rich Semitic language which is highly productive, both derivationally and inflectionally [2, 4]. The number of Arabic words is estimated to be 60 billion, derived from approximately 10,000 roots. Arabic poses many challenges for data mining tasks [2]. Most of these challenges are due to orthography and morphology. It is true that some of these challenges are shared with other languages but it exhibits considerable complexity from theoretical to computational linguistics. Furthermore, the language processing becomes even more challenging when considering the language used in social networking and microblogging sites, where dialects are heavily used. These dialects may differ in vocabulary, morphology, and spelling from the standard Arabic and most do not have standard spellings.

To overcome these challenges, we propose a novel event detection model that is language-independent. This model is based on frequency or co-occurrence of terms over time. Arabic event detection is enriched using automatically Named Entity Recognition, dictionaries, and Twitter features such as Retweet ratio and Hashtags.

Many researchers have proposed models and techniques for the purpose of identifying real-world events using social media data. In this paper, we propose an online classification-clustering framework, which is able to handle a constant stream of new documents with threshold parameters that can be modified in an experimental manner during training phase. The high volume of tweets from Twitter is the input of the system, which produces a table of the events in a particular region, associated sub-events (details) and *disruptive events* (as defined above) for a particular time (daily or hourly fashion). Social media data are very noisy; hence the first step in our framework after collecting data is preprocessing, which aims to reduce the amount of

noise before classification. The next step is to separate event-related tweets and non-event content. We implement a Naive Bayes machine classifier to achieve this. Then, we compute tweet features in order to extract similar characteristics and apply an incremental online clustering algorithm to assign each message in turn to a suitable event-based cluster by calculating each tweet's similarity to existing clusters, ultimately enabling us to detect a range of events. We focus in this work on real-world event identification for both large scale and rare (disruptive) events such as car accidents in a given location. Our contributions can be summarized as follows:

- Using our framework, we identify the relationship between Twitter activity and real-world events by detecting key events throughout the day;
- Using temporal, spatial and textual features, our framework is able to detect disruptive events at a given place for a particular time.
- Our framework is language independent as we address the challenging task of detecting events in Arabic.
- We validate our model on multiple real-world data sets to show the effectiveness of the framework.

The rest of the paper is organized as follows: Section 2 reviews related work on event detection in social media. In section 3, we discuss the main elements of our proposed framework. In section 4 we discuss several features; temporal, spatial and textual features. Section 5 presents our experiments and discusses the results. Finally, we conclude and highlight the future work of research in section 6.

2 Related Work

In the recent years, many researchers have shown interest in online event detection in social media. For instance, Petrovic et al. [11] presented an approach to detect breaking stories from a stream of tweets. The proposed approach, which is based on the locality-sensitive hashing (LSH), automatically organizes every incoming tweet in an existing story or labels it as a new story. In order to reduce the search space and improve the performance of the LSH, they added a secondary search, which indeed improves the results by19%. Using a different approach, Cordeiro [12] proposed a continuous wavelet transformation based on hashtag occurrences combined with a topic model inference using Latent Dirichlet Allocation (LDA). Instead of using individual words, hashtags are used to build wavelet signals. Wavelet peak and local maxima detection techniques are used to detect peaks in the hashtag signal. Then, LDA is applied to all tweets from the hashtag signal when an event is detected. However, these approaches do not differentiate whether topic detected is event-related or celebrity update. Non-event content such as personal or celebrity updates are not important to the decision-making process and may introduce noise.

Sakaki et al. [14] developed a probabilistic spatio-temporal model to monitor tweets and detect disastrous events such as earthquakes. Their method is based on features such as the keywords “Earthquake!” where they assumed that each user is regarded as a sensor with a function of detecting a target event and reporting it via

Twitter. One requirement of the approach is that to monitor an event we need to know the event in advance to provide representative keyword queries to be detected. This is an issue for detecting dynamic or unexpected events.

Becker et al. [21] proposed an online clustering framework, suitable for large-scale social media sites such as Twitter, to identify different types of real-world events. The online clustering technique groups together topically similar tweets and implements four features (Temporal features, Social features, Topical Features and Twitter-Centric Features) to distinguish between real-world events and non-events. Another study that stresses the importance of proper nouns identification to enhance the similarity comparison between tweets was presented by Phuvipadawat and Murata in [15]. Their method collected, grouped, ranked, and tracked breaking news from Twitter. Nevertheless, these two approaches are limited to widely discussed events and fail to report rare and potentially disruptive events. In addition, none of the aforementioned approaches have been shown to perform well with Arabic content.

The amount of research reported on Arabic information retrieval is considerably limited and immature compared to what is done in other less inflected languages. Most attention is focused on text classification, techniques used for language pre-processing like (stemmers and index tools), filtering and translation [2, 4]. Previous work on Arabic IR has used distance-based algorithms, Learning algorithms, Bayesian classification methods and N-grams for searching Arabic text documents [4].

3 Framework for Event Detection

Figure 1 illustrates our novel framework, which supports the automatic identification of events from social media. The five steps in the framework include; data collection, pre-processing, classification, on-line clustering and summarization. In this section we will explain each step in more detail.

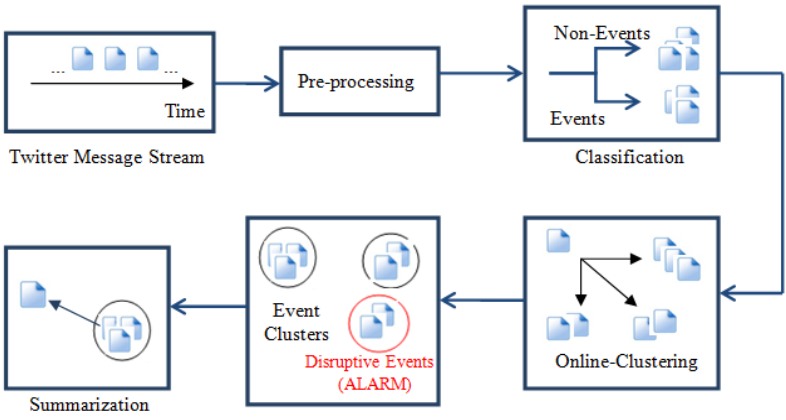


Fig. 1. Twitter Stream Event Detection Framework

3.1 Data Collection

We use Twitter's Streaming API to collect user-generated posts because it allows subscription to a continuous live stream of data. Our goal is to monitor and detect events (including disruptive events) in a given location without prior knowledge of these events. Thus, we collect tweets based on a set of keywords that generally describes a region (for example: Abu Dhabi) using different languages – Arabic and English. We also collect tweets from users who selectively add the required region as their location. In addition, we also make use of geographic Hashtags in the data collection process.

Data is stored using MongoDB [19], an open-source document database, which is easy to use and provides high availability speed and memory. MongoDB has been shown to be suitable for storing tweets, and supports different indices with straightforward queries [19].

3.2 Pre-processing

The goal of the pre-processing step is to represent data in a form that can be analyzed efficiently and to improve the data quality by reducing the amount of trivial noise (i.e. deleting tweets that are irrelevant to events). We perform text processing techniques such as stop-word elimination (Term frequency and TF-IDF are the criterions used for classifying stop words) and stemming (Khoja stemmer for Arabic tweets [22] and Porter Stemming [25] for English and other Latin tweets). In addition to the Arabic stop word list included in the Khoja stemmer [22], we added to it more stop words which are determined using Term frequencies and TF-IDF of the training corpus. Moreover, posts that were less than 3 words long were removed and tweets with one word accounting for over half of the words are also removed, as these posts are less likely to have useful information.

3.3 Classification

This step aims to distinguish events from noise or irrelevant tweets. Words from each tweet are considered as features and a Naive Bayes classifier was chosen for the classification task over a number of leading methods such as support vector machines (SVMs) or Logistic Regression, due to its performance in previous extensive experiments as demonstrated in [1]. The main reasons for using Naïve Bayes model are; it is relatively fast to compute, easy to construct with no need for any complex iterative parameter estimation schemes. Unlike SVMs or Logistic Regression, Naïve Bayes classifier treats each feature independently. Naïve Bayes also tends to do less overfitting compared to Logistic Regression [1, 14].

We used the R statistical software package (http://www.R-project.org), specifically the e1071 R package, to build and train the Naïve Bayes Classifier on a training corpus of 1500 tweets that have been annotated as "event" or "non-event". Event instances outnumber the non-event ones as the training set consisted of 600 Non-Event tweets and 900 Event-related tweets.

The features and their corresponding category (event or non-event) are provided to the classifier and these constitute the training set. From the training data the likelihood of each tweet belonging to either class is derived based on the occurrence of the tweet's features in the training data. When a new example is presented, the class likelihood for the unseen data is predicted based on the training instances.

Algorithmic steps:

i. Input tweets.
ii. Extract features from tweets.
iii. These features and their corresponding labels are used to train the learning algorithm (Naive Bayes classifier).
iv. New tweets are presented to the trained classifier to predict their label using their extracted features.

3.4 Online-Clustering

The classification step separates event-related documents from non-event posts (such as chats, personal updates, spam, incomprehensible messages). Consequently, non-event posts are filtered. To identify the topic of an event, including determining those that are disruptive events, we define a range of features including temporal, spatial and textual features, which are detailed in the next section. We then apply an online clustering algorithm, which is outlined in Algorithm 1.

Input:
n set of documents $(D_1,\ldots,D_n)$
Threshold τ
Output:
k clusters $(C_1,\ldots,C_k)$
Step 1: **For** a given τ, compute the centroid similarity function $E(D_i, c_j)$ of each cluster c_j
Step 2: **If** centroid similarity $E(D_i, c_j) \geq \tau$ **do**:
1) A new cluster is formed containing D_i
2) The new centroid value = D_i
Step 3: **If** centroid similarity $E(D_i, c_j) < \tau$ **do**:
1) Assign it to cluster which gives maximum value of $E(D_i, c_j)$
2) Add D_i to cluster j and recalculate the new centroid value c_j.

Algorithm 1. Online Clustering Algorithm

Using set of features $(F_1,\ldots,F_k)$ for each document (tweet) $(D_1,\ldots,D_n)$ we compute a similarity measure $E(D_i, c_j)$ between the document and each cluster $(C_1,\ldots,C_k)$ where similarity function is computed in turn against each cluster c_j for $j=1,\ldots,m$ and m is the number of clusters (initially $m=0$). In this paper, we use **the average** weight of each term across all documents in the cluster to calculate the centroid similarity function $E(D_i, c_j)$ of a cluster. The threshold parameters are determined empirically in the training phase.

The decision to use online clustering algorithm was taken for three main reasons: (i) it supports high dimensional data as it effectively handles the large volume of social media data produced around events; (ii) many clustering algorithms such as K-means require the prior knowledge of the number of clusters. As we do not know the number of events and sub-events *a priori* the online clustering is suitable as it does not require such input; (iii) partitioning algorithms are ineffective in this case because of the high and constant sheer scale of tweets [21].

3.5 Summarization

After clustering tweets into clusters, the next natural step would be to automatically summarize or represent topics being discussed within clusters. Each cluster may contain hundreds of tweets, and the task of finding most representative tweets or extracting top terms (topics) is essential to support the identification of events, especially disruptive events, so any potential security and safety issues can be managed. Summarization task is a very challenging task in its own and takes various forms [23]. The simplest approach is to consider each tweet as a document, and then apply a summarization method on this corpus to capture its key features [17, 21, 23]. Voting algorithms [17] are utilized in applications where in the context of microblogging sites take into account the following:

- The average length of a tweet;
- The total frequency of features in a tweet;
- Number of retweets, favorites and mentions;
- The inclusion of multimedia contents such as images.

In this paper, we implement a voting approach where the highest number of retweets in a cluster is used as a criterion for the summarization task. However, we leave the improvement of multilingual summarization of microblogs for future work.

4 Feature Selection

Many researchers have proposed enhancements to models or developed new approaches to optimize the capturing of patterns in the input signals. Here, we introduce several features related to the Twitter in order to reveal characteristics of clusters that are associated with rare real-world events particularly disruptive events.

4.1 Temporal Features

Temporal features are important factors that have been overlooked in many event detection studies using in social media. The volume of tweets and the continually updated commentary around an event suggests that informative tweets from several hours ago may not be as important as new tweets [21]. For this reason we retain the most frequently occurring terms a cluster in hourly time frames and compare the number of tweets posted during an hour that contain term t to the total number of

tweets posted during that hour. This helps identify terms that enable event clustering and also helps ordering events [8, 11, 14].

4.2 Spatial Features (Geospatial, Regional)

Events are characterized by rich set of spatial and demographic features [1]. In this paper, we make use of three statistical location approaches to extract geographic content from clusters. The first one is from Twitter where the source latitude and longitude coordinates are provided by the user. The second method depends on the shared media (photos and videos) by using the GPS coordination of the capture device (if supported). Third, Open NLP (http://opennlp.sourceforge.net) and Named-Entity Recognition (NER) were implemented for geotagging the tweet content (text) to identify places,, organization, street names, landmarks etc. These approaches rely purely on Twitter with no need for user IP, private login information, or external knowledge bases which give the maximum advantage [5, 24].

Once the geographic content is extracted from each tweet in a cluster, we aggregate them to determine the cluster's overall geographic focus. The higher the volume of tweets from nearly near coordinates, the higher the level of confidence in the location of the event will be. Table 1 presents a disruptive event (loss of communication) happening in the F1 event (see dataset in section 5.1) where spatial features are used to determine the cluster (event) overall location (Yas Marina).

Table 1. Spatial features are extracted (bold) from user's tweet to determine the cluster's overall location

Date	Time	User	Original tweet	Translated tweet	RT
04/11/2013	20:13:04	PJoc31		Having problem calling my friends using du in **Yas Island Rotana** hotel **#AbuDhab**i #F1 Grand Prix: The **Yas Marina** Circuit	5
04/11/2013	20:16:41	M7mdAS96	**ياس مارينا** مكان خيالي لكن ما عرف شو مشكلة الاتصال والاشارة دووووم ضعيفة. بليز ساعدوني #ضروري#F1 **#AbuDhabi**	The **Yas Marina Circuit** is an awesome venue however I am having trouble with communication and coverage signal. please help #F1 **#AbuDhabi**	2
04/11/2013	20:23:12	BintZayed91	كان الاتصال ممتاز في فترة الظهر ما عرف شو ياها دو من ربع ساعة أحاول اتصل او ارسل رسالة ماشي فايدة **شارع ياس بلازا** قريب **فندق روتانا #ياس** #F1	Connection was excellent at noon Don't know what happened with Du signal as I am trying to make a call or send sms from quarter of an hour with no success **Plaza st** near **Yas Rotana hotel #Yas** #F1	9

We assume that all locations provided by users are correct however [6] found that 34% of Twitter users had entered fake locations in their profile. Some users may intentionally misrepresent their home location either to cover for their actual location, or for privacy-security issues. On the other hand, some users provided location may differ from their actual location because their locations change frequently due to travel. The virtual sense of community should also be taken into consideration.

4.3 Textual Features

Textual or content features have been identified as contributing to the spread of a post in social media [13]. For example, hashtags are used to generate content features [7, 8], and identify topics affecting retweet likelihood [5, 8, 13, 26]. Here, we introduce the features we derived from tweet text.

Near-Duplicate Measure
The average content similarity over all pairs of tweets posted in a cluster (1-hour) is calculated using:

$$\sum_{a,b \in set\ of\ pairs\ in\ tweets} \frac{similarity\ (a,b)}{|set\ of\ pairs\ in\ tweets|}$$

where the content similarity is computed using the standard cosine similarity over words from tweet a, b vector representation $\overrightarrow{V}(a), \overrightarrow{V}(b)$ of the tweet content:

$$similarity\ (a,b) = \frac{\overrightarrow{V}(a).\overrightarrow{V}(b)}{|\overrightarrow{V}(a)||\overrightarrow{V}(b)|}$$

If the two tweets have a very high similarity, we assume that one of them is a near-duplicate of the other. The original tweet is considered as the first tweet in a particular time frame and/or the shortest tweet in length. Even though, duplicates are less likely to provide additional information about an event, several users independently witnessing an event and tweeting about it would effectively increase the confidence level of an event. An example of tweets with high near-duplicate measure is presented in Table 2 (from the 1st dataset in section 5.1).

Table 2. Severe weather alarm from tweets on the 2nd of November 2013

Date	Time	User	Original tweet	Translated tweet	RT
02/11/2013	6:09:52	hazza_saiff	صباح الخير ..**ضباب** علي **خط** #**ابوظبي العين**	Good morning.. **fog** on #**AbuDhabi alain highway**	2
02/11/2013	6:11:24	BuHazae	**ضباب كثيف** على **خط العين ابوظبي Net_AD@**	**Net_AD@ Heavy fog** on **abu dhabi alain highway**	**3**
02/11/2013	6:12:53	Rose_alduwaila	ارجوا الانتباه **ضباب خط العين ابوظبي** http://t.co/z0siijm WLC	Attention please **fog** on **AbuDhabi alain highway** http://t.co/z0siijm WLC	7
02/11/2013	6:12:58	mzinelsawari	**#برق الامارات ضباب كثيف في خط ابوظبي** قبل الخزنه	**#Uaebarq heavy fog** on **abu dhabi highway** before Alkhazna	**4**
02/11/2013	6:14:11	GroupStorms	**ابوظبي ضباب كثيف على خط #ابوظبي_العين**	**Abu Dhabi heavy fog** on **#abudhabi_alain highway**	3
02/11/2013	6:19:23	WALEED625	تنبيه: **ضباب كثيف** على مختلف طرق الخارجية إمارة #**ابوظبي** وبالذات **خط ابوظبي العين** نتمنى من الأخوة أخذ الحيطة	Attention: **heavy fog** on various external ways of #**Abu Dhabi** and **Abu Dhabi alain highway** in particular please brothers take extra caution	0

Retweet Ratio

Retweet represent the influence of a tweet beyond one-to-one interaction domain. Popular tweets could propagate multiple hops away from the source as they are retweeted throughout the network [7]. Hence, the number of retweets is an indication of popularity. Furthermore, retweeting in a social network can serve as a powerful tool to reinforce a message when not only one but a group of users repeat the same message [7, 8]. Therefore, retweet ratio indicates tweets surrounding an event where users agree with the message or wish to spread the information (warning, advice, evidence…) with other users. Retweet ratio has been implemented to detect events and to estimate rumors in social media stream [18]. We calculate this attribute by normalizing number of times a tweet appears in a timeframe to the total number of tweets in that timeframe.

Mention Ratio

A mention is a mechanism used in Twitter to reply to users, engage others or to join a conversation in a form of (@username). A user can mention one or more users anywhere in the body of the post. Hence, we calculate the number of mentions (@) relative to the number of tweets in a cluster. Ordinary users show a great passion for celebrities and as a result the most mentioned users are celebrities where sometimes users mention them without necessarily reading their posts [7, 13]. Regarding events reporting, users tend to mention journalists, politicians and official accounts such as news agencies or government official accounts to drive their attention about an event or to add more credibility to their event-related posts.

Hashtag Ratio

Hashtags are an important feature of social networking sites and can be inserted anywhere within a message. Some Hashtags indicate their posted messages (#bbcF1) and some others are dedicated originally to events such as (#abudhabigp). In addition, topic related hashtags are used as an information seeking index on Twitter to search Twitter for more tweets belonging to a topic. The use of hashtags became a coordinating mechanism for disruptive-related activity on Twitter [14, 20]. The Hashtag ratio is the ratio of tweets containing hashtag over the total number of tweets in that timeframe.

Link or Url Ratio

As Twitter is limited to 140 characters per message it is common in the Twitter community to include links when tweeting to share additional information or for referencing. Clusters that have tweets with links from popular websites (news agencies or government sites) may boost level of confidence of that information and hence more adoption to such tweets and clusters. Not all links refer to officials but mostly they are images or videos uploaded by users. Additionally, the co-occurrence of URLs in a cluster confirms that these tweets refer to the same event and improves the level of confidence of an event. This attribute is calculated by the fraction of tweets with URL to the total number of tweets in a timeframe.

Tweet Sentiment

Users express their opinions on a variety of topics in Twitter. They might discuss news, complain about services and express positive or negative sentiment about products [9, 10]. In fact, companies manufacturing such products have developed techniques to analyze these posts to get a sense of sentiment about their products [10].

In prior work, we found that negative sentiment is usually associated when reporting disruptive events (Negative overall cluster).The sudden change of tweets' sentiment is another observed characteristic of a disruptive event cluster. Here we focus on negative sentiment regarding identifying disruptive events, given that negative sentiment tweets are more likely to be retweeted as shown in [6, 8, 9]. We use a semantic classifier based on the SentiStrength model in [9].The SentiStrength algorithm is suitable because it is designed for short informal text with abbreviations and slang. Furthermore, it combines a lexicon-based model with a set of additional linguistic rules for spelling correction, negations, booster words (e.g., very), emoticons, and other factors. Most importantly, SentiStrength support multiple languages including Arabic.

Dictionary-Based Feature

One of the main objectives of our framework is the ability to automatically detect messages that contain precise information about disruptive events such as labor strike or fire incidences. To enrich such rare event identification, present tense verbs, popular event nouns and adjectives that describe events as they take place are considered as a feature. This bag of words model uses a dictionary of trigger words to detect and characterize events which are manually labeled by experts from several management departments such as traffic control department, crises departments, emergencies and others.

Examples of present verbs are: witness, notice, observe, participate, engage, listen etc.

Examples of event nouns are; breaking news, update, situation, delay etc.

Examples of event adjectives are; urgent, live, latest, severe, horrifying etc.

5 Experimental Evaluation

5.1 Experimental Setup

Data: Our first dataset, which consists of around 1.7 Million tweets (1698517), was collected from 15 October 2013 to 05 November 2013 using Twitter's Streaming API. Our initial aim was to monitor and analyze disruptive events associated with major events in a particular region. We chose the Formula 1 Motor Racing, which was hosted in Abu Dhabi (our input location) between 1st and 4th November 2013. The number of Arabic tweets is 890658 where English tweets are 39191. Around 24% of tweets were published in other Latin script and other languages. Figure 2 shows the language distribution in our first dataset. As our task focuses on Arabic event detection, we restrict our dataset to Arabic tweets and eliminate all non-Arabic tweets.

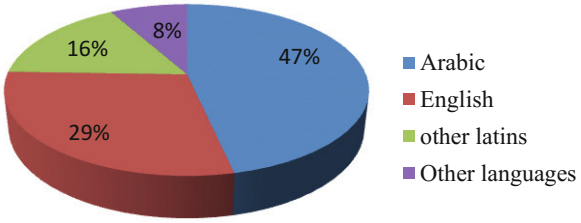


Fig. 2. The distribution of languages used in our dataset

Since then we focused our attention on collecting tweets for the purpose of analyzing disruptive events in the capital Abu Dhabi. In this work, we restrict our search to Arabic tweets. A considerable change of tweets volume was noticed from 2nd to 5th December 2014 due to the famous double-crime (considered as a terrorist attack) on the 2nd December 2014 which was unprecedented in the peaceful Abu Dhabi history. An American woman was murdered in a shopping mall. The second crime was held by the same suspect when she planted a primitive bomb on the doorstep of an American citizen in a different location. The second dataset consists of 1161854 Arabic tweets. Figure 3 shows the tweets volume in Abu Dhabi which clearly indicates the rise of posts' volume and discussions during the terrorist attack.

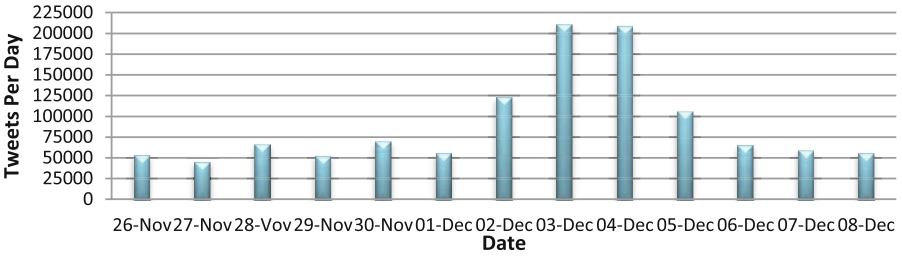


Fig. 3. The volume of tweets in the second data set from (26^{th} Nov to 8^{th} Dec) in Abu Dhabi

Annotation: To evaluate the framework, we evaluate the two main stages: classification and clustering. For classification, three human annotators manually labeled 1200 tweets in to two classes "Event" and "Non-Event" to train our classifiers (500 Non-Event tweets and 700 Event-related tweets). The agreement between our three annotators, measured using Cohen's kappa, was substantial (kappa = 0.807).

The resulting dataset after classification contained approximately 62,000 event-related tweets which we used to train, test and evaluate the clustering algorithm. We used the first 15 days of data (from 15/Oct until 29/Oct from the first dataset) to train the clustering algorithm and to tune the thresholds using the validation set. Then we tested the clustering algorithm on unseen data of the last 6 days from the 30th of Oct until the 4th of Nov. Threshold values were varied from 0.10 to 0.90 at graded increments of 0.05% with a total of 17 tests in order to find the best cut-off of $\tau = 0.55$ (77 character difference). Figure 4 illustrates the F-measure for different thresholds

where the best performing threshold τ =0.55 seems to be reasonable because it allows some similarity between posts but does not allow them to be nearly identical.

In order to evaluate the clustering performance, we employed three human annotators to manually label 637 clusters based on the highest number of retweets a post gets to represent that cluster. The task of the annotators was to choose one of the eight different categories: politics, finance, sport, entertainment, technology, culture, disruptive event and others. The agreement between annotators was calculated using Cohen's kappa (K=0.772) which indicates an acceptable level of agreement. We used only **492 clusters** on which all annotators agreed as the **gold standard**.

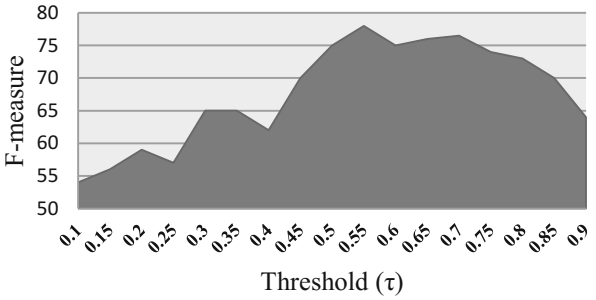


Fig. 4. F-measure of online clustering over different thresholds

5.2 Evaluation Matrices

To measure the effectiveness of classifiers based on our proposed features, we used a set of well-known classification metrics: precision, recall, accuracy, and F1 measure. Precision is how often are our predictions of a class are correct —a measure of false positives. Recall is how often tweets are classified correctly as the correct class — a measure of false negatives. F-measure is a harmonic mean of precision and recall. Accuracy is the proportion of the correctly classified tweets to the total number of tweets. A false positive is when the outcome is incorrectly predicted as X class when it is actually Y class. A true positive is when actual X class events are correctly predicted as X class events.

$$\text{Precision(P)} = \frac{\text{tp}}{\text{tp+fp}} \qquad \text{Recall(R)} = \text{True positive rate} = \frac{\text{tp}}{\text{tp+fn}}$$

$$\text{F} - \text{measure} = \frac{2\times\text{P}\times\text{R}}{\text{P+R}} \qquad \text{False positive rate} = \frac{\text{fp}}{\text{tn+fp}}$$

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{fn} + \text{tn}}$$

To evaluate the quality of clusters we compute average cluster precision (AP) [16] on the gold standard. The average precision measures how many of the identified clusters are correct averaged over hours per day and calculated based on the precision of each cluster per hour per day. Average precision is a common evaluation metric in

tasks like ad-hoc information retrieval where only the set of returned documents and their relevance judgments are available [1, 16, 20, 21].

5.3 Experimental Results

To evaluate the overall framework, we have to evaluate the two main elements. Starting with Classification: we found in [1] that the Naive Bayes classifier outperformed other machine learning algorithm (SVMs classifier and Logistic Regression) in classifying events using the English language. Furthermore, Naive Bayes classifier achieves better results using combination of attributes (Unigrams+ Bigrams+ part-of-speech (POS) + Named Entity Recognition (NER)) with F-measure value of 85.43%. Here we repeat the same experiment comparing the same machine learning algorithms but with only Arabic input and the new annotation. A ten-fold cross validation approach is adopted to train and test the methods using the WEKA machine learning toolkit for the classification task. Table 3 gives the F-measure results of the three machine learning algorithms using combination of attributes.

Table 3. F-scores of different classification algorithms

	Naive Bayes classifier	SVMs classifier	Logistic Regression classifier
F-measure	80.24	78.53	76.85

We obtain similar results to [1] as the Naïve Bayes classification method outperforms others. There is an overall drop in the performance of all three methods, which we expected due to the limitation of the used attributes. For example, Part-of-speech (POS) and Named Entity Recognition (NER) are very limited for Arabic language.

In order to evaluate the clustering performance, we used similar techniques to [1, 16, 21]. Average precision is calculated with respect to eight categories: politics, finance, sport, entertainment, technology, culture, disruptive event and other-event. Table 4 shows the average precision percentages of clusters in the test set.

Table 4. Average precision of the online clustering algorithm, in percent

Date	Politics	Finance	Sport	Entertainment	Technology	Culture	Disruption Events	Average Per Day
30-Oct	83.26	82.19	79.50	78.64	73.20	75.93	82.35	79.30
31-Oct	81.34	82.47	85.33	69.91	72.37	77.43	80.58	78.49
⋮								
4-Nov	79.75	81.86	81.93	79.38	80.46	81.51	83.02	81.13
Average Per Topic	81.39	80.62	79.57	73.23	76.13	77.54	82.26	78.68

While the online clustering algorithm achieves a good performance, the results are sometimes inconsistent with respect to topics. Not surprisingly, the average precision of identifying political events is greater than the average precision of identifying entertainment related events by about 9%. Since it is easier to extract and categorize

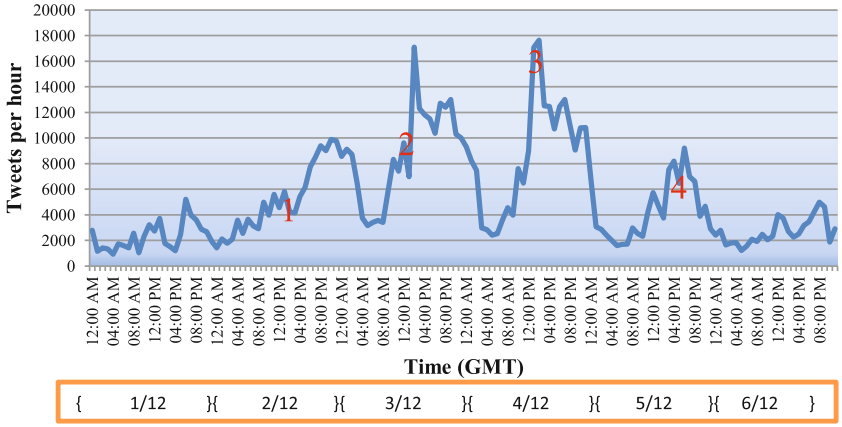
events like politics, finance, sport and disruptive events than events like entertainment, technology or cultural events even for humans which cause the main disagreement between annotators in the annotation task. Finally, it is important to notice that the framework is able to automatically identify disruptive events with the best performance of 82.26%.

One of the frameworks' objectives is to identify disruptive events and send a notification to the administrators. Table 5 shows the top 3 emerging disruptive events identified by the framework based on the number of retweet counts for the second dataset. For space limitation, we only present results of the disruptive incidents on the 2nd of Dec as an example of the system's output. The system can produce results with different level of time granularity (per hour, 3 hours, …, per day).

Table 5. Top 3 emerging disruptive events identified by the system on the 2nd of Decmber 2014

Date	User	Tweet	Translation	RT
Dec 2	AbuDhabiPolice	مشاجرة في دورة مياه تسفر عن مصرع سيدة بجزيرة الريم http://www.securitymedia.ae/ar/media.center/News/4202109.aspx	Woman Dies after Public Toilet Fight on Reem Island http://www.securitymedia.ae/ar/media.center/News/4202109.aspx	76
	Mona_Alraesi	حريق ضخم في محطة لتوزيع الكهرباء في ابوظبي بالقرب من مصفح الصناعيه ونسال الله السلامة للجميع pic.twitter.com/kLLc4L0hoJ	A huge fire in an electricity distribution station in Abu Dhabi near musaffah industrial area we ask God for everyone's safety	49
	NET_AD	أبوظبي الان : حادث تدهور على خط دبي_بوظبي بعد محطة السمحه مع وجود اصابات... نرجو أخذ الحيطه والحذر	Abu Dhabi now: there is a multiple car crashes on the Abu Dhabi_Dubai highway after Alsamha petrol station with several injuries ... please take caution	22

To provide further validation for our system, we evaluated it using the second dataset which contains more disruptive events than the first dataset. We were able to compare our disruptive event identification results with the official record of events, as the authorities released 2 YouTube videos with the exact time of these events (shown in Figure 5). All of these events were detected successfully by the framework. Figure 6 shows the clustering output of two time-frames (2-3PM on the 2nd of Dec and the same time of the next day 3/12/2014). The results suggest that the number of disruptive events (clusters in the red) increased dramatically over the same period from previous day as more people discussed the murder and its consequences.



1. An American woman was murdered in a Shopping mall. (Based on the CCTV which was released by the officials on YouTube, the time of the crime was between 1:12pm-2:45pm on the 2nd of Dec)
2. The Ministry of Interior released CCTV (On the 3rd of Dec at 12pm) footage of the suspect "Reem Island Ghost" and ask public for information.
3. Abu Dhabi Police revealed the second video (on YouTube on the 4th pf Dec at 1pm) which contains the double-crime, search, inspection procedures and the arrest of the suspect.
4. Minister of Interior made the announcement at a press conference about Reem Island Crime that the suspect has been arrested (5 Dec at 3pm).

Fig. 5. The volume of tweets in the second dataset from (1st Dec to 6th Dec) in Abu Dhabi with the main events detection

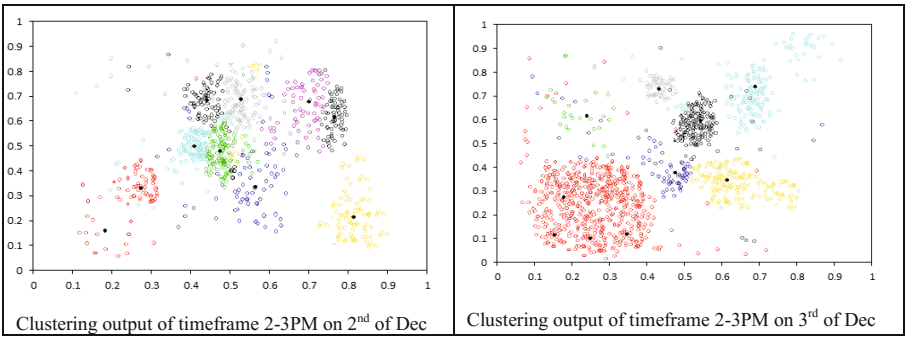


Fig. 6. The clustering output of two time-frames (2nd - 3rd/Dec/2014)

6 Conclusions and Future Work

In this paper we have presented an integrated framework to detect real-world events in Arabic from social media platform (Twitter). The event identification was performed through several stages; data collection, preprocessing, classification,

clustering and summarization. We have also shown that our approach is able to reveal disruptive events for a certain location using rich set of features. Extensive experiments were conducted to evaluate the effectiveness of the proposed framework using two real-world datasets.

This framework can be generalized to develop a social awareness system or for the purposes of decision making enrichment which can be implemented in many fields such as crises management or information intelligence. Our results support the claim that the use of social media for the purposes of information gathering could be utilized as a complementary to traditional intelligence and not to be used independently. In future we aim to compare our results with other works in the area of event detection on Twitter. This is a challenge due to the differences between datasets as each dataset has different size, time and characteristics. We also aim to validate our results against real-time complete official reports or official news streams.

There are many directions for future work. One of the main directions is to compare and validate the performance of the proposed framework against other well-known algorithms such as the state-of-the-art Labeled Dirichlet Allocation (LDA) method. Another direction is to study the contributions and limitations of various feature types to event detection in social media. Finally, detection of rumors in social media with deep analysis of the distinctive characteristics of rumors and the way they propagate in the microblogging communities will be carried out in the near future.

References

1. Alsaedi, N., Burnap, P., Rana, O.: A Combined Classification-Clustering Framework for Identifying Disruptive Events. In: Proceedings of 7th ASE International Conference on Social Computing (SocialCom 2014), pp. 1–10 (2014), http://ase360.org/handle/123456789/71
2. Darwish, K., Magdy, W.: Arabic Information Retrieval. Foundations and Trends® in Information Retrieval 7, 239–342 (2014), http://www.nowpublishers.com/articles/foundations-and-trends-in-information-retrieval/INR-031
3. PearAnalytics. Twitter study (August 2009), http://www.pearanalytics.com/wpcontent/uploads/2009/08/Twitter-Study-August-2009.pdf
4. Larkey, L., Ballesteros, L., Connell, M.: Light stemming for Arabic information retrieval. Arabic Computational Morphology, 221–243 (2007)
5. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: a content-based approach to geo-locating twitter users. In: Proceeding CIKM 2010, pp. 759–768 (2010), http://dl.acm.org/citation.cfm?id=1871535
6. Hecht, B., Hong, L., Suh, B., Chi, E.: Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 237–246 (2011)
7. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, P.: Measuring User Influence in Twitter: The Million Follower Fallacy. In: ICWSM 2010 (2010)
8. Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in twitter. Journal of the American Society for Information Science and Technology 64(7), 1399–1410 (2013)

9. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in Twitter events. Journal of the American Society for Information Science and Technology 62(2), 406–418 (2011)
10. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the ACL 2011 Workshop on Languages in Social Media, pp. 30–38 (2011)
11. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 181–189 (2010)
12. Cordeiro, M.: Twitter event detection: combining wavelet analysis and topic inference summarization. In: Doctoral Symposium on Informatics Engineering, DSIE 2012 (2012)
13. Cheng, J., Adamic, L., Dow, P., Jon, K., Jure, L. (2014), Can cascades be predicted? In: WWW 2014 (2014), http://dl.acm.org/citation.cfm?id=2567997
14. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: 19th International World Wide Web Conference, WWW 2010 (2010)
15. Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in Twitter. In: Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2010, pp. 120–123 (2010)
16. Bollmann, P.: A comparison of evaluation measures for document retrieval systems. Journal of Informatics, 97–116 (1977)
17. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 38 (1998)
18. Takahashi, T., Igata, N.: Rumor detection on twitter. In: SCIS ‘6 and ISIS ‘13, pp. 452–457 (2012)
19. Kumar, S., Morstatter, F., Liu, H.: Twitter Data Analytics. Springer (2014)
20. Dou, W., Wang, X., Skau, D., Ribarsky, W., Zhou, M.X.: LeadLine: Interactive visual analysis of text data through event identification. In: VAST 2012, pp. 93–102 (2012)
21. Becker, H., Naaman, M., Gravano, L.: Beyond Trending Topics: Real- Event Identification on Twitter. In: ICWSM, pp. 1–17 (2011)
22. Khoja, S., Garside, R., Knowles, G.: Stemming arabic text. In: NAACL 2001 (2001)
23. Chua, F., Asur, S.: Automatic Summarization of Events from Social Media. In: ICWSM 2013 (2012)
24. Mahmud, J., Nichols, J., Drews, C.: Where Is This Tweet From? Inferring Home Locations of Twitter Users. In: ICWSM, pp. 511–514 (2012), http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/viewFile/4605/5045
25. Porter, M.: An algorithm for suffix stripping. Program: Electronic Library & Information Systems 40(3), 211 – 218
26. Burnap, P., Williams, M.L., Sloan, L., Rana, O., Housley, W., Edwards, A., Knight, V., Procter, R., Voss, A.: Tweeting the Terror: Modelling the Social Media Reaction to the Woolwich Terrorist Attack. Social Network Analysis and Mining 4, 1 (2014)

Learning Semantically Rich Event Inference Rules Using Definition of Verbs

Nasrin Mostafazadeh[1] and James F. Allen[1,2]

[1] Computer Science Department, University of Rochester, Rochester, New York
[2] Institute for Human and Machine Cognition, Pensacola, Florida
{nasrinm,james}@cs.rochester.edu

Abstract. Natural language understanding is a key requirement for many NLP tasks. Deep language understanding, which enables inference, requires systems that have large amounts of knowledge enabling them to connect natural language to the concepts of the world. We present a novel attempt to automatically acquire conceptual knowledge about events in the form of inference rules by reading verb definitions. We learn semantically rich inference rules which can be actively chained together in order to provide deeper understanding of conceptual events. We show that the acquired knowledge is precise and informative which can be potentially employed in different NLP tasks which require language understanding.

1 Introduction

Systems performing NLP tasks such as Question Answering (QA), Recognizing Textual Entailment (RTE) and reading comprehension depend on extensive language understanding techniques to function. Deep language understanding enables an intelligent agent to construct a coherent representation of the scene intended to be conveyed through natural language utterances, connecting natural language to the concepts of the world. Developing a deep understanding system requires large amounts of conceptual and common-sense understanding of the world. As an example, consider a QA system which is given the question in Figure 1. One pre-requisite for answering this question is to semantically understand and interpret both query and the snippet. Figure 1 shows a generic semantic interpretation of the question and the snippet with grey labels. Throughout this paper we use the verbal semantic roles[1] as distinguished by TRIPS system (Allen et al., 2005).

After the semantic interpretation, the system understands that it should look for a *kill* event with Einstein as the *affected* person. However, the system does not see any explicit connection between the event in the question and the event presented in the snippet. Now let us provide the system with the following piece of knowledge in the form of an inference rule about the event *kill*:

$$(X_{agent}\ \textbf{kills}\ Y_{affected}) \xrightarrow{entails} (X_{agent}\ \textbf{causes}\ Y_{affected}\ to\ die) \qquad (1)$$

[1] http://trips.ihmc.us/parser/LFDocumentation.pdf

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 402–416, 2015.
DOI: 10.1007/978-3-319-18111-0_30

By having access to such an inference rule, the system will know that 'killing' entails 'cause to die', where explicitly the 'killer' causes the 'affected' to die. One can imagine many more complex pieces of knowledge presented in the form of inference rules, each of which can provide a new clue for a system which requires language understanding. It is obvious that a system should have various natural language processing capabilities in order to successfully answer questions, however, here we focus on the bottleneck of conceptual knowledge on events.

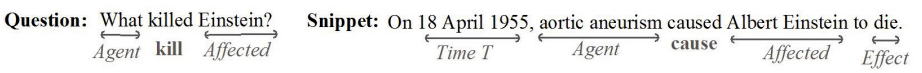


Fig. 1. Example question and its corresponding relevant information posed to a question answering system

As the earlier example shows, having conceptual knowledge about the events in the form of semantically rich inference rules – such as knowing what happens to the participants before and after it occurs or the consequences of the event – can play a major role in language understanding in different NLP applications. We believe that an effective conceptual knowledge should provide semantic reasoning capabilities, with semantic roles and sense disambiguation. In this paper, we introduce a novel attempt to automatically learn a semantically rich knowledge base for events, which provides high precision inference rules aligned by their semantic roles. We propose to learn the knowledge base by automatically processing large amounts of definitional knowledge about verbs, using their WordNet (Miller, 1995) word sense definitions (glosses). We accomplish this by deep semantic parsing of glosses, automatic extraction of inference rules, unsupervised alignment of semantic role labels, and chaining inference rules together until hitting a 'core' concept.

The phases of our approach are shown in Figure 2. We will provide details about each of these phases in Sections 2-4. The main outcome of our approach is the Inference Rules corpus which could be used for different language understanding tasks. In Section 5 we show that our semantic role alignment methodology is a promising way for acquiring precise and semantically rich inference rules. Moreover, we show that the inference rules acquired by our approach have higher precision than any other related work. Although we use WordNet here, our approach is applicable to any other definitional resources.

2 Deep Semantic Parsing of Definitions

As the first phase of our approach, we need to have deep semantic understanding of the verb definitions. Here we use the TRIPS broad-coverage semantic parser[2] which produces state-of-the-art logical form (LF) from natural language text

[2] `http://trips.ihmc.us/parser/cgi/parse`

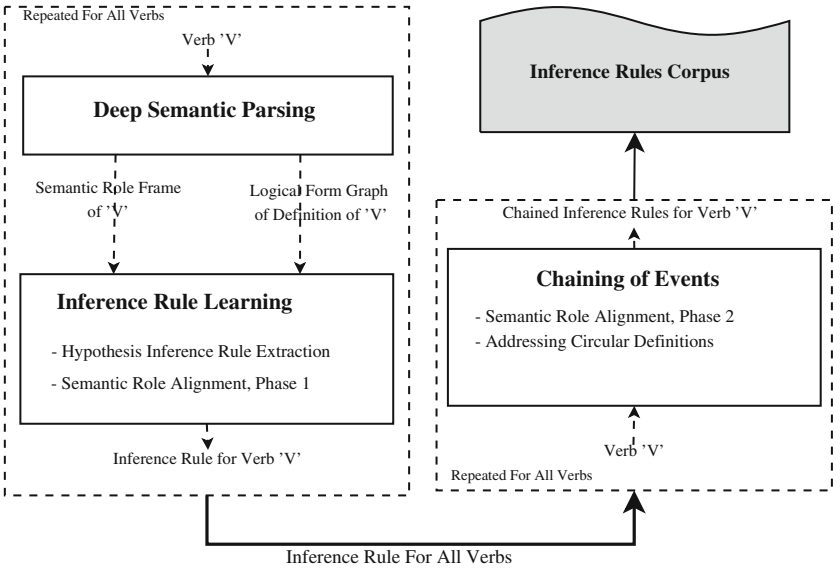


Fig. 2. The phases of our approach

(Allen et al., 2005). TRIPS provides an essential processing boost beyond other off-the-shelf applications, mainly sense disambiguated and semantically rich deep structures. The approaches presented in this paper can be applied to any other wide-coverage semantic parsers, such as Boxer system (Bos, 2008).

Many glosses are complex, often highly elliptical and hard to parse. For example, 'kill.v.1'[3] is defined as 'cause to die' which does not explicitly mention the subject or the object of the sentence. Another example is 'love.v.1' which has the gloss 'have a great affection or liking for', where the object of the sentence is missing. TRIPS recovers such missing information, producing a parse such as 'something causes something to die' (Allen et al., 2013) for the gloss of 'kill.v.1'. The output of this phase is the semantic role frame[4] (semframe) for each verb synset together with LF graph of its gloss. For instance, the semframe of the verb *kill.v.1* is given as $\{agent_{ont:person.n.1},\ affected_{ont:organism.n.1}\}$. Figure 3 shows the simplified LF graph of the gloss of 'kill.v.1'.

3 Inference Rule Learning

In the second phase of our approach we aim to extract semantically rich inference rules for all verb synsets.

[3] We represent WordNet words sense disambiguated using their part of speech and sense number. So 'kill.v.1' is the first sense of the verb 'kill'.

[4] Semantic role frame is called to the set of semantic roles associated with a verb.

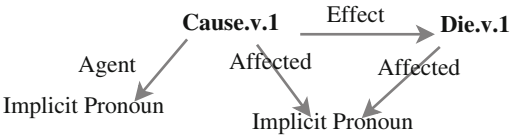


Fig. 3. Logical form produced by TRIPS for 'kill.v.1'

3.1 Hypothesis Inference Rule Extraction

Hypothesis inference rules are preliminary rules which are extracted using the two outputs of the deep semantic parsing phase. A hypothesis rule is an axiom with Left Hand Side (LHS) and Right Hand Side (RHS), each consisted of predicates where LHS logically entails RHS. There is always one predicate on the LHS, but there could be more than one predicates on the RHS (one of which is the root predicate, marked with '*'). LHS predicate comes from the semframe of the verb and the RHS predicates come from the LF graph of the verb's definition. We define a predicate to be either a verb or verb nominalization, as they inherently have the potential to occur at some time point as events. Here we stick to a very simple logical representation of axioms in the form of inference rules, which enables easy incorporation of our knowledge base in various systems. For instance, the following is the hypothesis rule that we deterministically extract for the verb 'kill.v.1':

$$(\textbf{kill.v.1}\ X_{agent}\ Y_{affected}) \Rightarrow (\textbf{cause.v.1}\ A_{agent}\ B_{affected}\ C_{effect})^* \wedge(\textbf{die.v.1}_C\ B_{affected}) \quad (2)$$

where each predicate is enclosed within parenthesis which has some arguments (semantic roles) realized with either variables or constants. As you can see, a predicate itself can be an argument of some other predicate, e.g., in the earlier example 'die.v.1' (reified with variable *C*) is the *effect* role of 'cause.v.1'. We call the set of hypothesis inference rules for all the WordNet verb sysnsets $corpus_{hypothesis}$. Now the question is whether a hypothesis rule is usable as an inference rule. The answer is that we often do not get a LF graph with all of the roles recognized correctly; and even if we do, more importantly we still do not know which role on the LHS corresponds to a role on RHS. This issue motivates 'semantic role alignment'.

3.2 Semantic Role Alignment, Phase 1

We want to know whether or not it is always the case that the *agent* role in the LHS of a rule maps to the *agent* role in the RHS and they should have the same realization. What happens to the *agent* role of LHS in case there is no *agent* role on the RHS? We call the problem of mapping the roles of the LHS to the roles of RHS 'Semantic Role Alignment' (SRA). The machine translation (MT) community has established an extensive literature on word alignment (Brown et al., 1993; Och and Ney, 2003), where translating 'she came' into French sentence 'elle est venue' requires an alignment between 'she' and 'elle', and between 'came' and

'est venue'. We believe that MT alignment approaches are suitable for the SRA task because of the following reasons:

- The semantic roles on the LHS and RHS tend to have semantic equivalence. So it is intrinsically the case that there is a (partial) mapping from roles on the RHS to the roles on the LHS.
- As opposed to the kind of inference rules learned based on distributional similarity (Harris, 1985) (to be discussed in Section 6), here the semantic content of LHS should not diverge substantially from RHS given the fact that RHS is basically defining LHS.
- MT alignment models are typically trained in an unsupervised manner, depending on sentence-aligned parallel corpora. For our task large volumes of training data are lacking, so an unsupervised training (to be explained in this section) is the most suitable approach.
- As it will be discussed in Section5, unsupervised aligners (which find hidden structures in data) can actually account for some frequent parsing errors in our system, which is very promising.

We model the SRA problem as a maximum bipartite matching problem: for each inference rule, we define n_{lhs} as the set of nodes such as l_i, each of which corresponds to a role in LHS; n_{rhs} is another set of nodes such as r_j, each of which corresponds to a role in RHS. Each pair of nodes (l_i, r_j) has an edge connecting them, which is weighted with the plausibility of the alignment of that pair. An alignment function a is defined as follows:

$$a : n_{lhs} \rightarrow n_{rhs} \cup \{null\}$$

which is a function mapping each role $\in n_{lhs}$ to a role $\in n_{rhs}$ or a null symbol, similar to IBM-style machine translation model (Brown et al., 1993). Here, mapping a LHS role to *null* means that the role should be 'inserted' in the RHS. Then the SRA problem is considered as a maximum weighted matching problem where the best alignment for the inference rule is the highest scoring a^*, under the constraint of 'one-to-one' matching, which is defined as follows:

$$a^* = \arg\max_a \{score(n_{lhs}, a, n_{rhs})\}$$

$$score(n_{lhs}, a, n_{rhs}) = \sum_{\substack{l_i \in n_{lhs} \\ r_j \in n_{rhs}}} score(l_i, a, r_j)$$

$$score(l_i, a, r_j) = log(Pr(l_i, a|r_j))$$

The training of the probability of aligning a role on LHS to a role on RHS, $Pr(l_i, a|r_j)$, is accomplished using the Expectation Maximization (EM) algorithm (Brown et al., 1993). In the E-step the expected counts for each role pair (l_i, r_j) are calculated and in M-step we normalize and maximize. We mainly estimate the so-called translation probability parameter $t(r|l)$ (Brown et al., 1993).

In order to prepare the data for performing the alignment explained above, we should firstly build an appropriate parallel corpus. Our idea is to build a corpus of LHS roles parallel with RHS roles from the set of hypothesis inference rules for all verbs in $corpus_{hypothesis}$. One issue to consider is that the rules with multi-predicate RHS cannot have a two-sided mapping. Among all 13,249 hypothesis inference rules that we generate, 649 one of them have two RHS predicates and only 10 of them have three RHS predicates. As the first step, we remove all the rules with multi-predicate RHS – about 0.4% of all the rules. With the remaining rules, we build a corpus of all LHS roles parallel with the RHS roles. We call this $corpus_{unary}$. Then we apply the alignment algorithm explained earlier to this corpus for learning the model parameters. Using the learned parameters, for each hypothesis inference rule we find the maximum weighted alignment. As an example, consider the verb *digest.v.3* which is defined as 'to tolerate something or somebody unpleasant'. The hypothesis inference rule produced for this verb is as follows:

$$(\textbf{digest.v.3}\ X_{pivot}\ Y_{affected}) \Rightarrow (\textbf{tolerate.v.4}\ A_{pivot}\ B_{theme})^* \qquad (3)$$

Figure 4 shows the bipartite matching graph for this inference rule. The maximum weighted matching is shown by the dark edges. As a result of maximum weighted matching, the aligned inference rule for 'digest.v.3' is the following:

$$(\textbf{digest.v.3}\ X_{pivot}\ Y_{affected}) \Rightarrow (\textbf{tolerate.v.4}\ X_{pivot}\ Y_{affected})^* \qquad (4)$$

We evaluate the outcome of this experiment, called '$phase1_{unary}$', in Section 5.

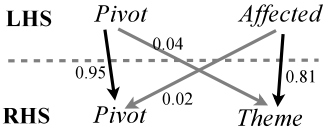


Fig. 4. The bipartite matching graph for alignment of inference rule 3

Our approach for SRA of the inference rules with multi-predicate RHS is linguistically motivated by the fact that the root predicate captures the core semantic meaning of the LHS. In short, our approach is as follows:

- Step 1: Discard the non-root RHS predicate and find the maximum weighted matching between LHS and the root RHS[5].
- Step 2: Make a set of nodes from the LHS roles which are matched to NULL in Step 1. Use this set as a new LHS, then find the maximum weighted matching to the non-root predicate.

[5] It is evident that a roles which is realized with the reification of another predicate, as with the *effect* role in (2), does not take part in the alignment problem.

This approach can generalized as a recursive SRA for rules which have more than two RHS predicates. The results of this experiment, named '$phase1_{bin}$', can be reviewed in Section 5. Applying this alignment approach the inference rule (2) results in the following aligned rule:

$$(\textbf{kill.v.1}\ X_{agent}\ Y_{affected}) \Rightarrow (\textbf{cause.v.1}\ X_{agent}\ Y_{affected}\ C_{effect})^{*} \wedge(\textbf{die.v.1}_{C}\ Y_{affected}) \quad (5)$$

At the end of this phase, we will have an aligned and ready to use level-1 inference rule generated for each WordNet verb synset. We call this collection $corpus_{level-1-rules}$. We associate a score with each inference rule, which is its normalized weighted matching score.

4 Chaining of Events

Given the inference rule for each verb from the previous phase, we want to expand our understanding of each event by chaining verbs together. For example, consider a QA system that has encountered the sentence "the boy skinned his knee when he fell" and wants to know more about the concept of 'skinning' by looking up the verb 'skin.v.2' in our knowledge base. The ideal information that we would like to be able to get by forward chaining of the level-1 inference rules is as follows:

$\mathcal{X}$ **skin.v.2** $\mathcal{Y}$:
$\xrightarrow{means}$ $\mathcal{X}$ **bruise.v.1** the [skin] of $\mathcal{Y}$
$\xrightarrow{means}$ $\mathcal{X}$ **injure.v.1** [the underlying soft tissue] of the [skin] of $\mathcal{Y}$
$\xrightarrow{means}$ $\mathcal{X}$ **cause.v.1** [harm] to the [underlying soft tissue] of the [skin] of $\mathcal{Y}$

Obtaining the above chaining requires yet another phase of role alignment, going from each level to the next one and expanding each predicate on the RHS.

4.1 Semantic Role Alignment, Phase 2

Consider the inference rule (5) which we obtained in the previous section. For the expansion of 'die.v.1' on the RHS, we will use its inference rule which is as follows:

$$(\textbf{die.v.1}\ X_{agent}) \Rightarrow (\textbf{lose.v.1}\ X_{agent}\ bodily\_attributes_{theme}) \quad (6)$$

As you can see the semframe of 'die.v.1' in inference rule (6) does not match its semframe in inference rule (5). There are many cases similar to this one and the reason is that semantic parsing and sense disambiguation are not perfect and are error prone. Moreover, verbs can have different semframes in different contexts. Here we perform semantic role alignment phase 2, using a similar method to '$phase1_{unary}$'. This time we build a corpus of all LHS definitions parallel with any of their usages in the entire $corpus_{level-1-rules}$. We call this new corpus $corpus_{def-use}$. EM can find hidden error patterns here as well as actual semantic

alignment patterns. The results of this experiment named '$phase2_{EM}$' can be reviewed in Section 5.

After finding the maximum weighted matching using the trained parameters, we get a new inference rule proper for continuing the forward chaining on 'kill.v.1' which is as follows:

$$(\textbf{die.v.1}\ X_{affected}) \Rightarrow (\textbf{lose.v.1}\ X_{affected}\ bodily\_attributes_{theme}) \qquad (7)$$

We have also obtained a probabilistic distribution on semframes for each synset given context, using the parsed glosses. We used this statistics together with EM alignment for favoring a specific semframe over another, resulting in a higher precision alignment in phase 2. The results of this experiment named '$phase2_{EM+}$' is reported in section 5.

The second phase of semantic role alignment results in high precision chaining and on average we get 10 new inference rules with high matching score after three levels of chaining – which increases the size of our inference rules corpus by an order of magnitude. For instance consider the verb 'kill.v.14', which is defined as 'to cause to cease operating'. This verb has three RHS predicates: cause, cease and operate and could have 2^3 different expansions just for the first level.

4.2 Addressing Circular Definitions

Usually there are some circular definitions for words in definitional resources including WordNet (Allen et al., 2011; Ide and Vronis, 1994). For example, the synset 'cause.v.1' is defined as 'cause.v.1 to happen.v.1' which is an immediate circulation. There have been some preliminary strategies (Allen et al., 2011) for breaking the definition cycles. Those findings show that some cycles can be resolved by selecting an alternative sense for the cyclical definition or simplifying the definitions. However, there are some key cycles which cannot be broken in this manner because there is essentially no specific simpler definition for some concepts, e.g., 'cause'. This is an essential problem with machine understanding, because machines have no direct experience with the world, which could have enabled them understand what a natural concept means.

This issue brings up an important psycholinguistic research, where it is believed that human lexicon is a complicated web of semantically related nodes instead of a one-to-one mapping of concepts to the words (Levary et al., 2012). According to earlier work, dictionaries have a set of highly interconnected nodes from which all other words can be defined (Picard et al., 2009). To our knowledge, there has not been any research on finding core concepts on WordNet verbs, using the graph theory experiments. Continuing the work on building dictionary graphs (Levary et al., 2012), we built a graph where directed links are drawn from a word to the words in its definition. In this graph, we found the strongly connected components using Tarjan's algorithm (Tarjan, 1972), which resulted in a set of 56 strongly connected components with size bigger than 1, which included 158 verb synsets. Other definitional paths of WordNet verbs converge to this set quickly, which we call the core verbs. Our idea is to stop forward chaining of definitions (avoiding circulation

trap) when we hit a core verb. The main core concepts that we have identified are as follows: cause, make, be, do, stop, start, begin, end, have, prevent, enable, disable.

After chaining of events and SRA phase 2, we obtain our final corpus of Inference Rules, containing new rules derived from chaining started from level-1 rules and going up to higher levels. We assign a score to each inference rule in different levels of the final corpus, which is the sum of normalized weighted matching scores divided to the number of levels.

5 Evaluation and Results

We have conducted two focused experiments for evaluating the two major contributions of our approach.

Semantic Role Alignment: We attempted to build a gold-standard corpus on semantic role alignment. For annotators, we used seven linguistics experts who had no relation to the work and three researchers who were involved. For each individual annotator, we randomly sampled 100 hypothesis inference rules from the $corpus_{hypothesis}$, and asked them to perform the role alignment for the given hypothesis rule[6]. The role alignment task was either of the following actions towards each RHS role:

- *Substitute*: substitute the role with one of the LHS roles (also decide about the realization value). This action corresponds to a role matching from LHS to RHS.
- *Delete:* Completely remove the role.

Moreover, they had the option of performing *Add* action, which involves adding a new role on RHS. This action corresponds to matching a LHS role with NULL.

The annotators were also asked to assign a confidence score (out of three) to the resultant aligned inference rule. This score takes into account the cases in which there is really no good alignment, and the annotator feels that his/her best possible alignment is not good at all[7]. We used this gold standard for computing precision scores for the SRA Phase 1 methods: $phase1_{base}$, $phase1_{unary}$, and $phase1_{bin}$. As it is critical to get the exact output, we used strict evaluation with no partial credit. We performed the same procedure on $corpus_{def-use}$, and built a gold-standard for evaluating the precision of SRA Phase methods: $phase2_{base}$, $phase2_{EM}$, and $phase2^{+}_{EM+}$). Both $phase1_{base}$ and $phase2_{base}$ are baselines which deterministically align LHS roles with the same RHS roles[8]. The results of these experiments reporting precision and average confidence score is presented in Table 1. In this table, Sc_{err} is the average annotator confidence score on incorrect alignments and Sc_{corr} is the average annotator confidence score on correct alignments of the corresponding method.

[6] We presented each rule together with some example usages of the synset, to give the annotators the context (Szpektor et al., 2007).

[7] This mostly happens for vague definitions or essential parsing errors.

[8] For 78% of all verb synsets we could find an exact name-based role match going from LHS to RHS.

Table 1. Semantic role alignment evaluation results

Method	$phase1_{base}$	$phase1_{unary}$	$phase1_{bin}$	$phase2_{base}$	$phase2_{EM}$	$phase2_{EM+}$
Precision	51%	**90%**	**87%**	10%	72%	79%
Sc_{corr}	3.0	2.7	2.5	3.0	2.28	2.32
Sc_{err}	2.5	2.3	1.2	2.7	1.3	1.4

The results show that the alignment using EM performs very well, providing promising framework for the task of semantic role alignment. The points that the system has missed are mostly for 'Delete' actions (91% of the time) of the annotators. System prefers not to delete any piece of information from the RHS, as it might be necessary for next chaining levels. However, there are some roles on the RHS which are artifacts of bad parse or inconsistent definitions, which annotator can pinpoint but the system cannot. Parsing artifacts are quite easy to be corrected by human, so the average confidence score on those errors is high, which has resulted in pretty high Sc_{err}.

The results obtained for $phase1_{bin}$ show that the alignment on binary rules (which initially seemed more complex) performs as good as the alignment on unary rules. Our observations show that this is because of the fact that many of binary rules are composed of a core predicate such as 'cause', 'stop', or 'do' which all have a recurring usage pattern, making the unsupervised alignment more successful. The baseline $phase1_{base}$ performs mediocre as a simple alignment method for phase1. Phase 2 alignment is always more complicated than phase 1. The baseline $phase2_{base}$ performs very poorly because verbs are mostly used (in context) with different semframe as compared with the semframe they are defined with (out of the context). The method $phase2_{EM}$ performs good, but is not enough for handling complicated alignments in def-use cases. The low Sc_{err} for phase 2 methods indicate the complexity of alignment task at this phase. $Phase2_{EM+}$ outperforms $phase2_{EM}$, which is mainly because it better predicts the cases of occasional bad parsing.

Inference Rules Corpus: To our knowledge, none of the earlier works on acquiring inference rules (details in Section 6) have inference rules with complex and semantically rich semantic roles and sense disambiguation as we do. Hence, in order to compare our inference rules to earlier works we simplify our inference rules dataset, removing all the sense tags and semantic roles. Here we use the most recent manually created verb inference rules dataset (Weisman et al., 2012), hereafter, test-set. This test-set is created by randomly sampling 50 common verbs in the Reuters corpus, and is then randomly paired with 20 most similar verbs according to the Lin similarity measure (Lin, 1998). This dataset includes 812 verb pairs, which are manually annotated by the authors as representing a valid entailment rule or not. They have used rule-based approach for annotation of entailment, where a rule $v1 \rightarrow v2$ is annotated 'yes' if the annotator could think of plausible contexts under which the rule holds (Szpektor et al., 2004). In this dataset 225 verb pairs are labeled as entailing and 587 verb pairs were labeled as non-entailing. Although this dataset is not very rich, it is a good testbed for comparing our inference rules against the state-of-the-art work on verb inference rules. Table 2 shows the results of the following methods:

- $Semantic-Rules_{simplified}$ is our simplified approach: given our final Inference Rules corpus (containing rules up to three levels of chaining or until hitting a core concept), simplify the rules by removing all the semantic roles, all the sense tags, and introduce a new rule for each of the predicates of a multi-predicate RHS. Given a pair (v_1, v_2) from the test-set, if the entailment $v_1 \rightarrow v_2$ exists in the simplified corpus, classify the pair as 'yes'.
- $Supervised_{linguistically-motivated}$ is the work on supervised learning of verb inference rules from linguistically-motivated evidence (Weisman et al., 2012).
- $VerbOcean_{KB}$ is the method that classifies a given pair as 'yes' if the pair appears in the strength relation in the VerbOcean knowledge-base (Chklovski and Pantel, 2004).
- *Random* is the method that randomly classifies a pair as 'yes' with a probability 27.7%, proportional to the number of 'yes' instances in the test-set against the number of 'no' instances.

Table 2. Evaluation results on hand annotated verb entailment pairs test-set

Method	Precision	Recall	F1-Score
$Semantic-Rules_{simplified}$	**50.0%**	45.1%	0.47
$Supervised_{linguistically-motivated}$	40.2%	71.0%	0.51
$VerbOcean_{KB}$	33.1%	14.8%	0.2
Random	27.9%	28.8%	0.28

As the results show, our simplified method outperforms the best method by 10% in precision. This reveals that the accuracy of our inference rules is high and our approach is capable of acquiring more precise verb inferences than the other methods. As expected, our coverage is lower than the *Supervised* method, which is due to the fact that we acquire our rules by reading verb definition and not by mining significantly large web-scale corpora, resulting in a smaller-scale dataset. However, our recall outperforms the $VerbOcean$ method and has also a competing F-1 score compared with the *Supervised* method. Of course for a successful usage of a knowledge base in an application, accuracy is crucial and coverage can be mitigated by using various kinds of precise knowledge bases. A large but noisy and unreliable knowledge base will be of little use in reasoning.

Analyzing the pairs that we have miss-classified as 'yes', there are many pairs which do not seem to be correctly annotated as 'no' in the test-set, such as (reveal, disclose) and (require, demand), where we argue that according to rule-based approach one can indeed think of a reasonable context under which *reveal* $\rightarrow$ *disclose* and *require* $\rightarrow$ *demand* hold. Another example is the pair (stop, prevent), which we classify as 'yes' in the context of the sixth sense of the verb 'stop', but is classified as 'no' in the test-set as the verbs in the test-set are not sense-disambiguated and do not have any context. Overall, our simplified approach proves to be competent with other works and also outperforms the state-of-the-art in precision, which is very promising.

6 Related Work

Early research has shown that definitions in online resources (such as dictionaries and lexicons) contain the type of knowledge that systems can benefit from for conceptual understanding of the world (Ide and Vronis, 1994). More specifically, WordNet's glosses have substantial world knowledge that could leverage semantic interpretation of text (Clark et al., 2008). Some earlier works (Moldovan and Rus, 2001; Clark et al., 2008) have tackled the problem of encoding WordNet glosses as axioms in first-order logic. These works use syntactically processed glosses for extracting the logical information (e.g., they map the NP in a subject position to an *agent* role), and successfully incorporate these axioms in QA and RTE tasks (Moldovan and Rus, 2001; Clark et al., 2008). However, their syntactic representation limits the functionality of semantic representation. Semantically rich logical representations (as opposed to syntactic ones) are proven to perform better on textual similarity and understanding tasks (Blanco and Moldovan, 2013).

The recent work on Multilingual eXtended WordNet (Erekhinskaya et al., 2014) attempts to semantically parse the glosses which is promising. The work on deriving event ontologies (Allen et al., 2013) using WordNet glosses best addresses the shortcomings of semantic interpretation in previous works. It tries to build complex concepts compositionally using OWL-DL description logic and enables reasoning to derive the best classification of knowledge. However, their work mainly derives ontological information, whereas our work extracts full axioms in the form of inference rules. Also, as shown in Section 3, we use a more simple approach for expressing our inference rules (axioms), which enables semantic role alignment (a novel task introduced in this paper), resulting in a more precise, accurate, and easily usable inference rules for other NLP tasks. The earlier works on predicate-argument alignment have been mainly focused on finding lexical similarity and overlaps between pairs of sentences (Wolfe et al., 2013) which is different from aligning the semantic roles of not necessarily similar predicates as we do.

The main relevant work is on automatic acquisition of inference rules. Inference rules, e.g., 'someone$_x$ commutes $\rightarrow$ someone$_x$ changes positions', are very useful for tasks such as QA and RTE. The predominant approach, DIRT (Lin and Pantel, 2001), is based on distributional similarity, where two templates (such as 'X murder Y' and 'X kill Y') are deemed semantically similar if their argument vectors are similar. This similarity measure results in weak (and often incorrect) entailments (Melamud et al., 2013), but results in huge datasets. Among the 12 million DIRT inference rules only about about 50% seem correct and reasonable (Melamud et al., 2013). One instance of an incorrect rule is 'X entered Y $\rightarrow$ X left Y', which captures temporal relation between two predicates, and is incorrect as an entailment. Some later works have attempted to make the inference rules more precise by using lexical expansions for argument vectors (Melamud et al., 2013). However, their approaches still tend to produce many incorrect or too general entailments, such as 'Y is hijacked in X $\rightarrow$ Y crashes in X', which is the result of reporting bias which means there have been many reported hijacking events which have resulted in crashes, but hijacking

does not entail crash necessarily. All of the earlier inference rule acquisition approaches mostly use predicates with two arguments, which can result in limited and less-accurate application of rules for textual understanding tasks; however, our approach covers many complex predicate structures with various number of arguments and inter-connected predicates.

VerbOcean (Chklovski and Pantel, 2004) is another related work, which identifies verb entailment through instantiation of some manually constructed patterns. This idea led to more precise rules, but weak coverage since verbs do not co-occur often with patterns. In Section 5 we show that our approach outperforms VerbOcean by about 17% in precision and 30% in recall. Another recent work on acquiring inference rules is the work on learning verb inference rules from linguistically-motivated evidence (Weisman et al., 2012). This work argues that although most of the works on learning inference rules are using distributional similarity, they utilize information from various textual scopes ranging from verb co-occurrence within a sentence to a document, as well as corpus statistics, which results in richer set of linguistically motivated features in their supervised classification framework. Although they outperform some earlier methods, their method is still limited to verb to verb entailment without typed entities and semantic roles, which could make their rules less effective in actual language understanding tasks. In Section 5 we show that our approach results in a more accurate verb inference rules, outperforming this work by about 10%. More importantly, our approach attempts to produce semantically rich inference rules, i.e, sense-disambiguated predicates with all of the necessary semantic roles, which is far beyond the simple inference rules produced by this work.

Furthermore, paraphrases can be viewed as bidirectional inference rules. The works on automatic derivation of paraphrase databases (Dolan et al., 2004; Quirk et al., 2004) share some of the shortcomings of the works on acquiring inference rules. Mostly the paraphrase sets with the highest precision contain too general/trivial paraphrase rules (Ganitkevitch et al., 2013) such as 'higher than 90% $\leftrightarrow$ higher than 90 per cent' or 'and its relationship $\leftrightarrow$ and its link'. Inherently, definitions provide non-trivial pieces of information, so our set of high precision inference rules hardly contains such rules. Unlike these works, we rely on reading definitions instead of web-scale free texts which gives us higher precision of non-trivial inference rules, however, results in a smaller set of rules. By incorporating and linking various definitional resources, one can increase the size of the inference rules yielded by our approach.

7 Conclusion

We presented a novel attempt to automatically build a conceptual knowledge about events in the form of inference rules, which can serve as a semantically rich knowledge base useful for various language understanding tasks. We accomplish this by deep semantic parsing of glosses, inference rule learning enhanced by semantic role alignment, and chaining of the events. The evaluation results show that our semantic role alignment technique is very promising and our inference

rules are precise and informative pieces of knowledge. We have shown that learning inference rules by reading definitional resources can result in high accuracy and inherently non-trivial pieces of knowledge. In order to expand the coverage of our knowledge base, we are planning to apply our approach to other dictionaries. Moreover, we are looking into improving our semantic role alignment techniques for chaining of events, which can potentially result in more accurate inference rules. Our future goal is to experiment employing our definitional knowledge in QA and Reading Comprehension Tests.

Acknowledgments. We would like to thank William de Beaumont and anonymous reviewers for their invaluable comments. This work is funded by The Office of Naval Research under grant number N000141110417 and Nuance Foundation.

References

Allen, J., Beaumont, W.D., Blaylock, N., Ferguson, L.G.G., Orfan, J., Swift, M., Teng, C.M.: Acquiring commonsense knowledge for a cognitive agent. In: Proceedings of the AAAI Fall Symposium Series: Advances in Cognitive Systems (2011)

Allen, J., Beaumont, W.D., Galescu, L., Orfan, J., Swift, M., Teng, C.M.: Automatically deriving event ontologies for a commonsense knowledge base. In: IWCS (2013)

Allen, J., Ferguson, G., Swift, M., Stent, A.: Two diverse systems built using generic components for spoken dialogue (recent progress on trips). In: Proceedings of the ACL Demo, ACLdemo 2005, pp. 85–88. ACL (2005)

Blanco, E., Moldovan, D.: A semantically enhanced approach to determine textual similarity. In: Proceedings of EMNLP, pp. 1235–1245. ACL, Seattle (2013), http://www.aclweb.org/anthology/D13-1123

Bos, J.: Wide-coverage semantic analysis with boxer. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing, pp. 277–286. Research in Computational Semantics, College Publications (2008)

Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19, 263–311 (1993)

Chklovski, T., Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In: Proceedings of the EMNLP (2004), http://aclweb.org/anthology/W04-3205

Clark, P., Fellbaum, C., Hobbs, J.R., Harrison, P., Murray, W.R., Thompson, J.: Augmenting wordnet for deep understanding of text. In: Semantics in Text Processing (2008)

Dolan, B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: Proceedings of the 20th COLING, COLING 2004. ACL, Stroudsburg (2004), http://dx.doi.org/10.3115/1220355.1220406

Erekhinskaya, T.N., Satpute, M., Moldovan, D.I.: Multilingual extended wordnet knowledge base: Semantic parsing and translation of glosses. In: LREC, pp. 2990–2994 (2014)

Ganitkevitch, J., Durme, B.V., Callison-Burch, C.: PPDB: The paraphrase database. In: Proceedings of NAACL-HLT, pp. 758–764. ACL, Atlanta (2013), http://cs.jhu.edu/~ccb/publications/ppdb.pdf

Harris, Z.: Distributional structure. In: Katz, J.J. (ed.) The Philosophy of Linguistics. Oxford University Press, New York (1985)

Ide, N., Véronis, J.: Knowledge extraction from machine-readable dictionaries: An evaluation. In: Steffens, P. (ed.) EAMT-WS 1993. LNCS, vol. 898, pp. 17–34. Springer, Heidelberg (1995)

Levary, D., Eckmann, J.P., Moses, E., Tlusty, T.: Loops and self-reference in the construction of dictionaries. Phys. Rev. X (2012)

Lin, D., Pantel, P.: Dirt: Discovery of inference rules from text. In: Proceedings of the Seventh ACM SIGKDD, pp. 323–328. ACM, New York (2001), http://doi.acm.org/10.1145/502512.502559

Melamud, O., Dagan, I., Goldberger, J., Szpektor., I.: Using lexical expansion to learn inference rules from sparse data. In: Proceedings of ACL 2013 (2013)

Miller, G.: Wordnet: A lexical database for english. Communications of the ACM (1995)

Moldovan, D.I., Clark, C., Harabagiu, S.M., Hodges, D.: Cogex: A semantically and contextually enriched logic prover for question answering. J. Applied Logic 5(1), 49–69 (2007), http://dx.doi.org/10.1016/j.jal.2005.12.005

Moldovan, D.I., Rus, V.: Explaining answers with extended wordnet. In: ACL (2001)

Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. CL 29(1), 19–51 (2003), http://dx.doi.org/10.1162/089120103321337421

Picard, O., Blondin-Masse, A., Harnad, S., Marcotte, O., Chicoisne, G., Gargouri, Y.: Hierarchies in dictionary definition space. In: NIPS Workshop on Analyzing Networks and Learning With Graphs (2009)

Quirk, C., Brockett, C., Dolan, W.: Monolingual machine translation for paraphrase generation (2004)

Szpektor, I., Shnarch, E., Dagan, I.: Instance-based evaluation of entailment rule acquisition. In: Carroll, J.A., van den Bosch, A., Zaenen, A. (eds.) Proceeding of ACL Conference. ACL (2007)

Szpektor, I., Tanev, H., Dagan, I., Coppola, B.: Scaling web-based acquisition of entailment relations. In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004, pp. 41–48. ACL (July 2004)

Tarjan, R.: Depth first search and linear graph algorithms. SIAM Journal on Computing (1972)

Weisman, H., Berant, J., Szpektor, I., Dagan, I.: Learning verb inference rules from linguistically-motivated evidence. In: Proceedings of EMNLP-CoNLL, pp. 194–204. ACL, Jeju Island (2012), http://www.aclweb.org/anthology/D12-1018

Wolfe, T., Durme, B.V., Dredze, M., Andrews, N., Beller, C., Callison-Burch, C., DeYoung, J., Snyder, J., Weese, J., Xu, T., Yao, X.: Parma: A predicate argument aligner. In: Proceedings of ACL Short (2013), http://www.cs.jhu.edu/~vandurme/papers/PARMA:ACL:2013.pdf

Rehabilitation of Count-Based Models for Word Vector Representations

Rémi Lebret[1,2] and Ronan Collobert[1]

[1] Idiap Research Institute, Martigny, Switzerland
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
remi@lebret.ch, ronan@collobert.com

Abstract. Recent works on word representations mostly rely on predictive models. Distributed word representations (aka word embeddings) are trained to optimally predict the contexts in which the corresponding words tend to appear. Such models have succeeded in capturing word similarities as well as semantic and syntactic regularities. Instead, we aim at reviving interest in a model based on counts. We present a systematic study of the use of the Hellinger distance to extract semantic representations from the word co-occurrence statistics of large text corpora. We show that this distance gives good performance on word similarity and analogy tasks, with a proper type and size of context, and a dimensionality reduction based on a stochastic low-rank approximation. Besides being both simple and intuitive, this method also provides an encoding function which can be used to infer unseen words or phrases. This becomes a clear advantage compared to predictive models which must train these new words.

1 Introduction

Linguists assumed long ago that words occurring in similar contexts tend to have similar meanings [1,2]. Using the word co-occurrence statistics is thus a natural choice to embed similar words into a common vector space [3,4]. Common approaches calculate the frequencies, apply some transformations (tf-idf, PPMI), reduce the dimensionality and calculate the similarities [5]. Considering a fixed-sized word dictionary $\mathcal{D}$ and a set of words $\mathcal{W}$ to embed, the co-occurrence matrix C is of size $|\mathcal{W}| \times |\mathcal{D}|$. C is then dictionary size-dependent. One can apply a dimensionality reduction operation to C leading to $\bar{C} \in \mathbb{R}^{|\mathcal{W}| \times d}$, where $d \ll |\mathcal{D}|$. Dimensionality reduction techniques such as Singular Value Decomposition (SVD) are widely used (e.g. LSA [6], ICA [7]). In [8,9], the authors provide a full range of factors to use for properly extracting semantic representations from the word co-occurrence statistics of large text corpora. While word co-occurrence statistics are discrete distributions, an information theory measure such as the Hellinger distance seems to be more appropriate than the Euclidean distance over a discrete distribution space. In this respect, [10] propose to perform a principal component analysis (PCA) of the word co-occurrence

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 417–429, 2015.
DOI: 10.1007/978-3-319-18111-0_31

probability matrix to represent words in a lower dimensional space, while minimizing the reconstruction error according to the Hellinger distance. In practice, they just apply a square-root transformation to the co-occurrence probability matrix, and then perform the PCA of this new matrix. They compare the resulting word representations with some well-known representations on named entity recognition and movie review tasks and show that they can reach similar or even better performance.

This paper proposes an extension of the work of [10] by investigating the impact of different factors. [11] show that a subsampling approach to imbalance between the rare and frequent words improves the performance. Recent approaches for word representation have also shown that large windows of context are helpful to capture semantic information [11,4]. While, in [10], only the 10,000 most frequent words from the dictionary $\mathcal{W}$ are considered as context dictionary $\mathcal{D}$, we investigate various types of context dictionaries, with only frequent words or rare words, or a combination of both. In this previous work, the co-occurrence counts to build C are based on a single context word occurring just after the word of interest. In this paper, we analyse various sizes of context, with both symmetric and asymmetric windows. For deriving low-dimensional vector representations from the word co-occurrence matrix C, PCA can be done by eigenvalue decomposition of the covariance matrix C^TC or SVD of C. Covariance-based PCA of high-dimensional matrices can lead to round-off errors, and thus fails to properly approximate these high-dimensional matrices in low-rank matrices. And SVD will generally requires a large amount of memory to factorize such huge matrices. To overcome these barriers, we propose a dimensionality reduction based on stochastic low-rank approximation and show that it outperforms the covariance-based PCA.

Recently, distributed approaches based on neural network language models have revived the field of learning word embeddings [12,13,14,15,16]. Such approaches are trained to optimally predict the contexts in which words from $\mathcal{W}$ tend to appear. [17] present a systematic comparison of these predictive models with the models based on co-occurrence counts, which suggests that context-predicting models should be chosen over their count-based counterparts. In this paper, we aim at showing that count-based models should not be buried so hastily. A neural network architecture can be hard to train. Finding the right hyperparameters to tune the model is often a challenging task and the training phase is in general computationally expensive. Counting words over large text corpora is on the contrary simple and fast. With a proper dimensionality reduction technique, word vector representations in a low-dimensional space can be generated. Furthermore, it gives an encoding function represented by a matrix which can be used to encode new words or even phrases based on their counts. This is a major benefit compared to predictive models which will need to train vector representations for them. Thus, in addition to being simple and fast to compute, count-based models become a simple, fast and intuitive solution for inference.

2 Hellinger-Based Word Vector Representations

2.1 Word Co-occurrence Probabilities

"You shall know a word by the company it keeps" [2]. Keeping this famous quote in mind, word co-occurrence probabilities are computed by counting the number of times each context word $c \in \mathcal{D}$ (where $\mathcal{D} \subseteq \mathcal{W}$) occurs around a word $w \in \mathcal{W}$:

$$p(c|w) = \frac{p(c,w)}{p(w)} = \frac{n(c,w)}{\sum_c n(c,w)}\,, \tag{1}$$

where $n(c,w)$ is the number of times a context word c occurs in the surrounding of the word w. A multinomial distribution of $|\mathcal{D}|$ classes (words) is thus obtained for each word w:

$$P_w = \{p(c_1|w), \ldots, p(c_{|\mathcal{D}|}|w)\}\,. \tag{2}$$

By repeating this operation over all words from $\mathcal{W}$, the word co-occurrence matrix C is thus obtained:

$$C = \begin{pmatrix} p(c_1|w_1) & \cdots & p(c_{|\mathcal{D}|}|w_1) \\ p(c_1|w_2) & \cdots & p(c_{|\mathcal{D}|}|w_2) \\ \vdots & \ddots & \vdots \\ p(c_1|w_{|\mathcal{W}|}) & \cdots & p(c_{|\mathcal{D}|}|w_{|\mathcal{W}|}) \end{pmatrix} = \begin{pmatrix} P_{w_1} \\ P_{w_2} \\ \vdots \\ P_{w_{|\mathcal{W}|}} \end{pmatrix} \tag{3}$$

The number of context words to consider around each word is variable and can be either symmetric or asymmetric. The co-occurrence matrix becomes less sparse when this number is high. Because we are facing discrete probability distributions, the Hellinger distance seems appropriate to calculate similarities between these word representations. The square-root transformation is then applied to the probability distributions P_w, and the word co-occurrence matrix is now defined as:

$$\tilde{C} = \begin{pmatrix} \sqrt{P_{w_1}} \\ \sqrt{P_{w_2}} \\ \vdots \\ \sqrt{P_{w_{|\mathcal{W}|}}} \end{pmatrix} = \sqrt{C}\,. \tag{4}$$

2.2 Hellinger Distance

Similarities between words can be derived by computing a distance between their corresponding word distributions. Several distances (or metrics) over discrete distributions exist, such as the Bhattacharyya distance, the Hellinger distance or Kullback-Leibler divergence. We chose here the Hellinger distance for its simplicity and symmetry property (as it is a true distance). Considering two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, the Hellinger distance is formally defined as:

$$H(P,Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}\,, \tag{5}$$

which is directly related to the Euclidean norm of the difference of the square root vectors:

$$H(P,Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2 \,. \tag{6}$$

Note that it makes more sense to take the Hellinger distance rather than the Euclidean distance for comparing discrete distributions, as P and Q are unit vectors according to the Hellinger distance ($\sqrt{P}$ and $\sqrt{Q}$ are units vector according to the ℓ_2 norm).

2.3 Dimensionality Reduction

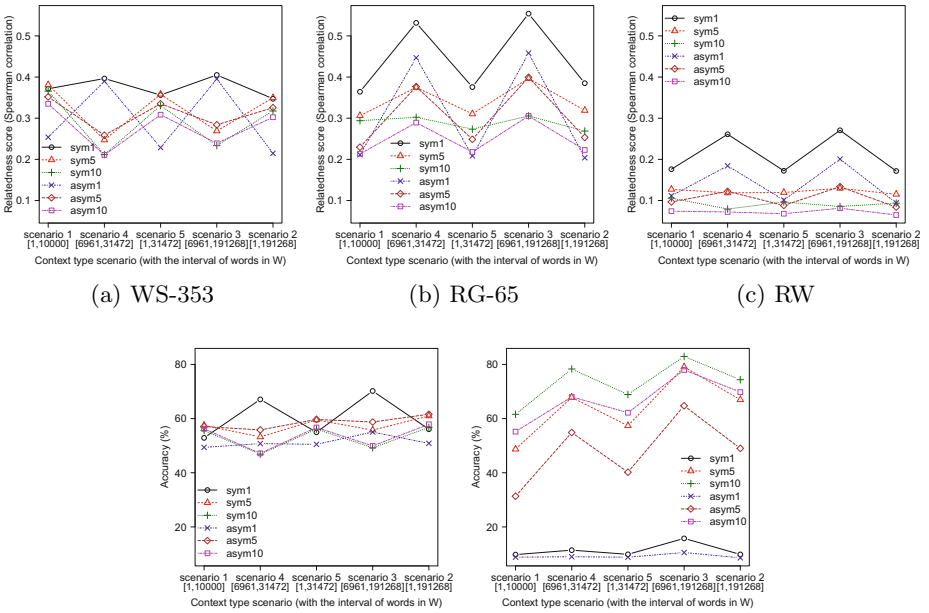
As discrete distributions are dictionary size-dependent, using directly the distribution as a word representation is, in general, not really tractable for large dictionary. This is even more true in the case of a large number of context words, distributions becoming less sparse. We investigate two approaches to embed these representations in a low-dimensional space: (1) a principal component analysis (PCA) of the word co-occurrence matrix $\tilde{C}$, (2) a stochastic low-rank approximation to encode distributions $\sqrt{P_w}$.

Principal Component Analysis (PCA). We perform a principal component analysis (PCA) of the square root of the word co-occurrence probability matrix to represent words in a lower dimensional space, while minimizing the reconstruction error according to the Hellinger distance. This PCA can be done by eigenvalue decomposition of the covariance matrix $\tilde{C}^T\tilde{C}$. With a limited size of context word dictionary $\mathcal{D}$ (tens of thousands of words), this operation is performed very quickly (See [10] paper for details). With a larger size for $\mathcal{D}$, a truncated singular value decomposition of $\tilde{C}$ might be an alternative, even if it is time-consuming and memory-hungry.

Stochastic Low-Rank Approximation (SLRA). When dealing with large dimensions, the computation of the covariance matrix might accumulate floating-point roundoff errors. To overcome this issue and to still fit in memory, we propose a stochastic low-rank approximation to represent words in a lower dimensional space. It takes a distribution $\sqrt{P_w}$ as input, encodes it in a more compact representation, and is trained to reconstruct its own input from that representation:

$$||VU^T\sqrt{P_w} - \sqrt{P_w}||^2 \,, \tag{7}$$

where U and $V \in \mathbb{R}^{|\mathcal{D}| \times d}$. U is a low-rank approximation of the co-occurrence matrix $\tilde{C}$ which maps distributions in a d-dimension (with $d \ll |\mathcal{D}|$), and V is the reconstruction matrix. $U^T\sqrt{P_w}$ is a distributed representation that captures the main factors of variation in the data as the Hellinger PCA does. U and V are trained by backpropagation using stochastic gradient descent.



(a) WS-353 (b) RG-65 (c) RW

(d) Mikolov's syntactic (e) Mikolov's semantic

Fig. 1. Performance on datasets with different types of context word dictionaries $\mathcal{D}$ (scenarios in the ascending order of their number of words), and different window sizes (in the legend, *sym1* is for symmetric window of 1 context word, *asym1* is for asymmetric window of 1 context word, etc.). Spearman rank correlation is reported on word similarity tasks. Accuracy is reported on word analogy tasks.

3 Experiments

3.1 Building Word Representation over Large Corpora

Our English corpus is composed of the entire English Wikipedia[1] (where all MediaWiki markups have been removed). We consider lower case words to limit the number of words in the dictionary. Additionally, all occurrences of sequences of numbers within a word are replaced with the string "NUMBER". The resulting text is tokenized using the Stanford tokenizer[2]. The data set contains about 1.6 billion words. As dictionary $\mathcal{W}$, we consider all the words within our corpus which appear at least one hundred times. This results in a 191,268 words dictionary. Five scenarios are considered to build the word co-occurrence probabilities with context words $\mathcal{D}$: (1) Only the 10,000 most frequent words within this dictionary. (2) All the dictionary. [11] have shown that better word representations can be obtained by subsampling of the frequent words. We thus define

[1] Available at `http://download.wikimedia.org`. We took the January 2014 version.
[2] Available at `http://nlp.stanford.edu/software/tokenizer.shtml`

the following scenarios: (3) Only words whose appearance frequency is less than 10^{-5}, which is the last 184,308 words in $\mathcal{W}$. (4) To limit the dictionary size, we consider words whose appearance frequency is less than 10^{-5} and greater than 10^{-6}. This results in 24,512 context words. (5) Finally, only words whose appearance frequency is greater than 10^{-6}, which gives 31,472 words.

3.2 Evaluating Word Representations

Word Analogies. The word analogy task consists of questions like, "a is to b as c is to ?". It was introduced in [16] and contains 19,544 such questions, divided into a semantic subset and a syntactic subset. The 8,869 semantic questions are analogies about places, like "*Bern* is to *Switzerland* as *Paris* is to ?", or family relationship, like "*uncle* is to *aunt* as *boy* is to ?". The 10,675 syntactic questions are grammatical analogies, involving plural and adjectives forms, superlatives, verb tenses, etc. To correctly answer the question, the model should uniquely identify the missing term, with only an exact correspondence counted as a correct match.

Word Similarities. We also evaluate our model on a variety of word similarity tasks. These include the WordSimilarity-353 Test Collection (WS-353) [18], the Rubenstein and Goodenough dataset (RG-65) [19], and the Stanford Rare Word (RW) [20]. They all contain sets of English word pairs along with human-assigned similarity judgements. WS-353 and RG-65 datasets contain 353 and 65 word pairs respectively. Those are relatively common word pairs, like *computer:internet* or *football:tennis*. The RW dataset differs from these two datasets, since it contains 2,034 pairs where one of the word is rare or morphologically complex, such as *brigadier:general* or *cognizance:knowing*.

3.3 Analysis of the Context

As regards the context, two main parameters are involved: (1) The context window size to consider, *i.e.* the number of context words c to count for a given word w. We can either count only context words that occurs after w (asymmetric context window), or we can count words surrounding w (symmetric context window). (2) The type of context to use, *i.e.* which words are to be chosen for defining the context dictionary $\mathcal{D}$. Do we need all the words, the most frequent ones or, on the contrary, the rare ones? Figure 1 presents the performance obtained on the benchmark datasets for all the five scenarios described in Section 3.1 with different sizes of context. No dimensionality reduction has been applied in this analysis. Similarities between words are calculated with the Hellinger distance between the word probability distributions. For the word analogy task, we used the objective function 3CosMul defined by [21], as we are dealing with explicit word representations in this case.

Table 1. Two rare words with their rank and their 5 nearest words with respect to the Hellinger distance, for a symmetric window of 1 and 10 context words

	WINDOW SIZE	
	1	10
BAIKAL	MÄLAREN	LAKE
(n°37415)	TITICACA	SIBERIA
	BALATON	AMUR
	LADOGA	BASIN
	ILMEN	VOLGA
SPECIAL-NEED	AT-RISK	PRESCHOOL
(n°165996)	SCHOOL-AGE	KINDERGARTEN
	LOW-INCOME	TEACHERS
	HEARING-IMPAIRED	SCHOOLS
	GRADE-SCHOOL	VOCATIONAL

Window Size. Except for semantic analogy questions, best performance are always obtained with symmetric context window of size 1. However, performance dramatically drop with this window size on the latter. It seems that a limited window size helps to find syntactic similarities, but a large window is needed to detect the semantic aspects. The best results are thus obtained with a symmetric window of 10 words on the semantic analogy questions task. This intuition is confirmed by looking at the nearest neighbors of certain rare words with different sizes of context. In Table 1, we can observe that a window of one context word brings together words that occur in a same syntactic structure, while a window of ten context words will go beyond that and add semantic information. With only one word of context, Lake *Baikal* is therefore neighbor to other lakes, and the word *special-needs* is close to other words composed of two words. With ten words of context, the nearest neighbors of *Baikal* are words in direct relation to this location, *i.e.* these words cannot match with other lakes, like Lake *Titicaca*. This also applies for the word *special-needs*, where we find words related to the educational meaning of this word. This could explain why the symmetric window of one context word gives the best results on the word similarity and syntactic tasks, but performs very poorly on the semantic task. Finally, the use of a symmetric window instead of an asymmetric one always improves the performance.

Type of Context. First, using all words as context does not imply to reach the best performance. With the 10,000 most frequent words, performance are fairly similar than with all words. An in-between situation with words whose appearance frequency is greater than 10^{-6} gives also quite similar performance. Secondly, discarding the most frequent words from the context distributions helps, in general, to increase performance. The best performance is indeed obtained

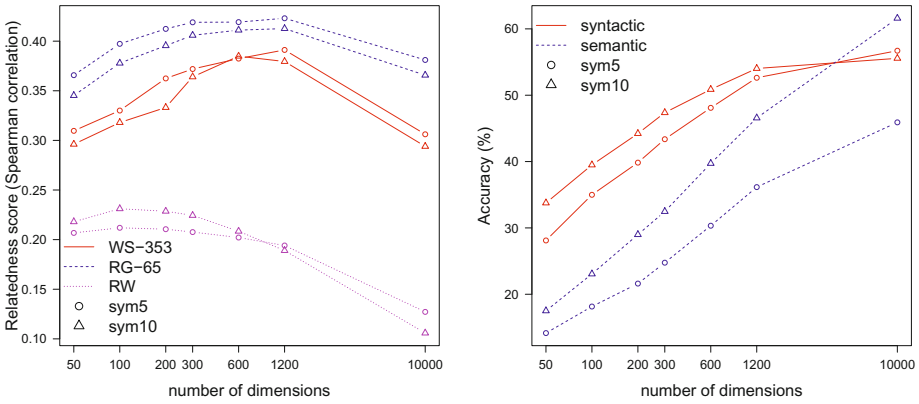
Table 2. The average number of context words in the co-occurrence matrix according to the type and the size of context

TYPE	DIM.	SIZE		
		1	5	10
Most frequent	10000	297	1158	1618
From 10^{-5} to 10^{-6}	24512	132	674	1028
Up to 10^{-6}	31472	396	1672	2408
From 10^{-5}	184308	249	1305	2050
All	191268	513	2304	3430

with scenarios (3) and (4). But all rare words are not necessarily essential to achieve good performance, since results with words whose appearance frequency is less than 10^{-5} and greater than 10^{-6} are not significantly lower. These two observations might be explained by the sparsity of the probability distributions. Counts in Table 2 show significant differences in terms of sparsity depending on the type of context. Similarities between words seem to be easier to find with sparse distributions. The average number of context words (*i.e.* features) whose appearance frequency is less than 10^{-5} and greater than 10^{-6} with a symmetric window of size 1 is extremely low (132). Performance with these parameters are still highly competitive on syntactic tasks. Within this framework, it then becomes a good option for representing words in a low and sparse dimension.

3.4 Dimensionality Reduction Models

The analysis of the context reveals that word similarities can even be found with extremely sparse word vector representations. But these representations lack semantic information since they perform poorly on the word analogy task involving semantic questions. A symmetric window of five or ten context words seems to be the best options to capture both syntactic and semantic information about words. The average number of context words is much larger within these parameters, which justifies the need of dimensionality reduction. Furthermore, this analysis show that a large number of context words is not necessary to achieve significant improvements. Good performance on syntactic and similarity tasks can be reached with the 10,000 most frequent words as context. Using instead a distribution of a limited number of rare words increases performance on the semantic task while reducing performance on syntactic and similarity tasks. We then focus on the two scenarios with the fewest number of context words: scenarios (1) and (4) with 10,000 and 24,512 words respectively. This reasonable number of context words allows for dimensionality reduction methods to be applied in an efficient manner.



(a) Word similarities datasets (b) Word analogies datasets

Fig. 2. Performance on datasets with different dimensions using scenario (1). Dimensionality reduction has been obtained with the Hellinger PCA. Spearman rank correlation is reported on word similarity tasks. Accuracy is reported on word analogy tasks.

Number of Dimensions. When a dimensionality reduction method is applied, a number of dimensions needs to be chosen. This number has to be large enough to retain the maximum variability. It also has to be small enough for the dimensionality reduction to be truly meaningful and effective. We thus analyse the impact of the number of dimensions using the Hellinger PCA of the co-occurrence matrix from scenario (1) with a symmetric context of five and ten words. Figure 2 reports performance on the benchmark datasets described in Section 3.2 for different numbers of dimensions. The ability of the PCA to summarize the information compactly leads to improved results on the word similarity tasks, where performance is better than with no dimensionality reduction. On the WS-353 and RG-65 datasets, we observe that the gain in performance tends to stabilize between 300 and 1,200 dimensions. The increase in dimension leads to a small drop after 100 dimensions on the RW dataset. However, adding more and more dimensions helps to increase performance on word analogy tasks, especially for the semantic one. We also observe that ten context words instead of five give better results for word analogy tasks, while the opposite is observed for word similarity tasks. This confirms the results observed in Section 3.3.

Stochastic Low-Rank Approximation vs Covariance-Based PCA. In this section, we compare performance on both word evaluation tasks using the two methods for dimensionality reduction described in Section 2.3. Experiments with symmetric window of five and ten context words are run to embed word representations in a d-dimensional vector, with $d = \{100, 200, 300\}$. All results are reported in Table 3. Except for some isolated results, performance is always much better with the stochastic low-rank approximation approach than

Table 3. Performance comparison between dimensionality reduction with stochastic low-rank approximation (SLRA) and Hellinger PCA (HPCA). A symmetric context of five or ten words with scenarios (1) and (4) have been used. The best three results for each dataset are in bold, and the best is underlined. Spearman rank correlation is reported on word similarity tasks. Accuracy is reported on word analogy tasks.

	SLRA						**HPCA**					
Dimension	100		200		300		100		200		300	
Window Size	5	10	5	10	5	10	5	10	5	10	5	10
Context dictionary = the $10,000$ *most frequent words*												
WS-353	0.48	0.54	0.54	**0.57**	**0.60**	**0.60**	0.40	0.38	0.41	0.39	0.42	0.41
RG-65	**0.55**	0.50	0.49	**0.56**	0.46	**0.52**	0.33	0.32	0.36	0.33	0.37	0.36
RW	0.27	0.25	**0.32**	**0.30**	**0.34**	**0.30**	0.21	0.23	0.21	0.23	0.21	0.22
Syn. Ana.	46.3	51.0	58.6	**61.0**	**61.7**	**59.2**	35.0	39.5	39.8	44.2	43.4	47.4
Sem. Ana.	20.4	35.9	29.1	47.0	34.0	**48.0**	18.1	23.1	21.6	29.0	24.8	32.5
Context dictionary = words whose frequency is between 10^{-5} *and* 10^{-6}												
WS-353	0.46	0.47	0.54	0.54	0.54	0.55	0.28	0.27	0.23	0.26	0.22	0.25
RG-65	0.46	0.40	0.41	0.42	0.49	0.45	0.29	0.31	0.26	0.30	0.23	0.29
RW	0.24	0.24	0.27	0.24	0.27	0.29	0.19	0.21	0.15	0.16	0.11	0.14
Syn. Ana.	39.0	45.1	52.9	53.7	56.4	58.8	45.2	47.4	46.1	48.7	47.3	49.2
Sem. Ana.	24.1	36.7	38.0	**54.3**	47.3	**62.5**	28.8	37.7	33.9	42.3	38.9	46.4

with a Hellinger PCA approach. Calculating the reconstruction error of both approaches confirms that the PCA fails somehow to properly reduce the dimensionality. For a reduction from 10,000 to 100 dimensions, the PCA reconstruction error is 532.2 compared with 440.3 for the stochastic low-rank approximation. This result is not really surprising, since it is well-known that standard PCA is exceptionally fragile, and the quality of its output can suffer dramatically in the face of only a few grossly corrupted points [22]. Covariance-based PCA as proposed in [10] is thus not an approach offering a complete guarantee of success. An approach to robustifying PCA must be considered. This is what we propose with the stochastic low-rank approximation which, moreover, ensures a low memory consumption. For a given dimension, a window of ten context words outperforms, in general, a window of five context words. This confirms once again the benefit of using a larger window of context. Performance are globally better with 300 dimensions, but performance with 200 dimensions is just slightly lower, or even better in certain cases. Finally, using a distribution of rare words instead of frequent words (*i.e.* scenario (4) instead of scenario (1) here) has only an impact on the semantic word analogy task.

3.5 Comparison with Other Models

We compare our word representations with other available models for computing vector representations of words: (1) the GloVe model which is also based on co-occurrence statistics of corpora [4][3], (2) the continuous bag-of-words (CBOW) and the skip-gram (SG) architectures which learn representations from prediction-based models [11][4]. The same corpus and dictionary $\mathcal{W}$ as the ones described in Section 3.1 are used to train 200-dimensional word vector representations. We use a symmetric context window of ten words, and the default values set by the authors for the other hyperparameters. We also compare these models directly with the raw distributions, computing similarities between them with the Hellinger distance. Results reported in Table 4 show that our approach is competitive with prediction-based models. Using the raw probability distributions yields good results on the semantic task, while a dimension reduction with a stochastic low-rank approximation gives a better solution to compete with others on similarity and syntactic tasks.

Table 4. Comparison with raw distributions and other models for 200-dimensional word vector representations. A symmetric context window of ten words is used. Spearman rank correlation is reported on word similarity tasks. Accuracy is reported on word analogy tasks.

	WS	**RG**	**RW**	**SYN.**	**SEM.**
Raw	0.37	0.31	0.10	56.8	83.0
SLRA	0.57	0.56	0.30	61.0	47.0
GloVe	0.57	0.57	0.38	82.2	84.1
CBOW	0.57	0.53	0.36	64.8	28.4
SG	0.66	0.53	0.42	72.7	66.9

3.6 Inference

Relying on word co-occurrence statistics to represent words in vector space provides a framework to easily generate representations for unseen words. This is a clear advantage compared to methods focused on learning word embeddings, where the whole system needs to be trained again to learn representations for these new words. To infer a representation for a new word w_{new}, one only needs to count its context words over a large corpus of text to build the distribution $\sqrt{P_{w_{\text{new}}}}$. This nice feature can be extrapolated to phrases. Table 5 presents some interesting examples of unseen phrases where the meaning clearly depends on the composition of their words. For instance, words from the entity *Chicago Bulls* differ in meaning when taken separately. *Chicago* will be close to other American cities, and *Bulls* will be close to other horned animals. However, it can be seen

[3] Code available at `http://www-nlp.stanford.edu/software/glove.tar.gz`
[4] Code available at `http://word2vec.googlecode.com/svn/trunk/`

Table 5. Examples of phrases and five of their nearest words. Phrases representations are inferred using the encoding matrix U with a symmetric window of ten context words and 300 dimensions.

NEW PHRASES	NEAREST WORDS
BRITISH AIRWAYS	AIRLINES, LUFTHANSA, QANTAS, KLM, FLIGHTS
CHICAGO BULLS	CELTICS, LAKERS, PACERS, KNICKS, BULLS
NEW YORK CITY	CHICAGO, BROOKLYN, NYC, MANHATTAN, PHILADELPHIA
PRESIDENT OF THE UNITED STATES	PRESIDENT, SENATOR, BUSH, NIXON, CLINTON

in Table 5 that our model infers a representation for this new phrase which is close to other NBA teams, like the *Lakers* or the *Celtics*. This also works with longer phrases, such as *New York City* or *President of the United States*.

4 Conclusion

We presented a systematic study of a method based on counts and the Hellinger distance for building word vector representations. The main findings are: (1) a large window of context words is crucial to capture both syntactic and semantic information; (2) a context dictionary of rare words helps for capturing semantic, but by using just a fraction of the most frequent words already ensures a high level of performance; (3) a dimensionality reduction with a stochastic low-rank approximation approach outperforms the PCA approach. The objective of the paper was to rehabilitate count-vector-based models, whereas nowadays all the attention is directed to context-predicting models. We show that such a simple model can give nice results on both similarity and analogy tasks. Better still, inference of unseen words or phrases is easily feasible when relying on counts.

Acknowledgements. This work was supported by the HASLER foundation through the grant "Information and Communication Technology for a Better World 2020" (SmartWorld).

References

1. Harris, Z.: Distributional structure 10 (1954)
2. Firth, J.R.: A Synopsis of Linguistic Theory 1930-55 (1957)
3. Turney, P., Pantel, P.: From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research (2010)
4. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: EMNLP (2014)
5. Lowe, W.: Towards a theory of semantic space. In: Conference of the Cognitive Science Society (2001)

6. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review (1997)
7. Väyrynen, J.J., Honkela, T.: Word Category Maps based on Emergent Features Created by ICA. In: STeP 2004 Cognition + Cybernetics Symposium (2004)
8. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior Research Methods (2007)
9. Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. Behav. Res. Methods (2012)
10. Lebret, R., Collobert, R.: Word Embeddings through Hellinger PCA. In: EACL (2014)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS (2013)
12. Collobert, R., Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: ICML (2008)
13. Huang, F., Yates, A.: Distributional Representations for Handling Sparsity in Supervised Sequence-Labeling. In: ACL (2009)
14. Turian, J., Ratinov, L., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning. In: ACL (2010)
15. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: NIPS (2013)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR Workshop (2013)
17. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: ACL (2014)
18. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing Search in Context: The Concept Revisited. ACM Transactions on Information Systems (2002)
19. Rubenstein, H., Goodenough, J.B.: Contextual Correlates of Synonymy. Communications of the ACM (1965)
20. Luong, M., Socher, R., Manning, C.D.: Better Word Representations with Recursive Neural Networks for Morphology. In: CoNLL (2013)
21. Levy, O., Goldberg, Y.: Linguistic Regularities in Sparse and Explicit Word Representations. In: CoNLL (2014)
22. Jolliffe, I.: Principal Component Analysis. Springer (1986)

Word Representations in Vector Space and their Applications for Arabic

Mohamed A. Zahran[1], Ahmed Magooda[1], Ashraf Y. Mahgoub[1], Hazem Raafat[2], Mohsen Rashwan[3], and Amir Atyia[1]

[1] Computer Engineering Department, Cairo University, Egypt
[2] Computer Science Department, Kuwait University, Kuwait
[3] Electronics and Communications Department, Cairo University, Egypt
{moh.a.zahran,ahmed.ezzat.gawad,ashraf.youssef.mahgoub}@gmail.com,
hazem@cs.ku.edu.kw, mrashwan@rdi-eg.com,
amir@alumni.caltech.edu

Abstract. A lot of work has been done to give the individual words of a certain language adequate representations in vector space so that these representations capture semantic and syntactic properties of the language. In this paper, we compare different techniques to build vectorized space representations for Arabic, and test these models via intrinsic and extrinsic evaluations. Intrinsic evaluation assesses the quality of models using benchmark semantic and syntactic dataset, while extrinsic evaluation assesses the quality of models by their impact on two Natural Language Processing applications: Information retrieval and Short Answer Grading. Finally, we map the Arabic vector space to the English counterpart using Cosine error regression neural network and show that it outperforms standard mean square error regression neural networks in this task.

Keywords: Word Representations, Word Vectors, Word Embeddings, Arabic Natural Language Processing, Arabic Information Retrieval, Arabic Short Answer Grading, Arabic-English vector space mapping, Cosine regression neural network.

1 Introduction

Researchers proposed various techniques to leverage large amount of unlabeled data. One particular method that is adopted by many researchers is representing individual words of a language as vectors in a multidimensional space that capture semantic and syntactic properties of the language. These representations can serve as a fundamental building unit to many Natural Language Processing (NLP) applications. Word representation is a mathematical model representing a word in space, mostly a vector. Each component (dimension) is a feature to this word that can have semantic or syntactic meaning.

Collobert and Weston [1] proposed a unified architecture for natural language processing. They used a deep neural network architecture that is jointly trained using

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 430–443, 2015.
DOI: 10.1007/978-3-319-18111-0_32

back propagation for many tasks: Part of Speech tagging, Chunking, Named Entity Recognition, and Semantic Role Labeling. Individual words are embedded into a d-dimensional vector where each dimension is regarded as a feature. Words representations (embeddings) are stored in a matrix $W \in \mathbb{R}^{d \times |D|}$, where D is a dictionary of all unique words. Look up tables are used for retrieving specific features for words. Sentences are represented using the embeddings of their forming words using a window around the word of interest, which solves the problem of variable length sentences. Mnih and Hinton [2] introduced Hierarchical log-bilinear model (HLBL) that is another form of word representations. For n-gram sentences, it concatenates the embedding of the first $n-1$ words and learns a neural linear model to predict the last word. It is a probabilistic model that uses softmax layer to transform the similarity between the predicted representations with the reference representations to a probability distribution.

Mikolov et al. [3] built a neural language model using a recurrent neural network (RNN) that encode the context word by word and predict the next word. He used the trained network weights as the words representation vectors. The network architecture has input layer with recurrent feed. It has one hidden layer and an output layer. The training procedure iterates over the sentences, each sentence is broken down to words, the input layer receives the current word encoded in a one-hot vector encoding (1-of-N coding, where N is the vocabulary size). The recurrent feed is the previously encoded context history. The length of the recurrent feed depends on the length of the hidden layer. The output layer is a softmax layer that generates a probability distribution over the vocabulary, which means that the length of the output layer equals the size of the vocabulary. The RNN is trained using back propagation to maximize the likelihood of the data using this model.

Turian et al. [4] presented a survey for various work that had been done for representing words in vector space, they also presented yet another neural language model resembles the work of Collobert and Weston to represent words in vector space.

Mikolov et al. [5, 6] proposed two new techniques for building word representation in vector space using log linear models; continuous bag of word (CBOW) and Skipgram (SKIP-G) models. These techniques are based on a neural network architecture with the hidden layer replaced with a simple projection layer to reduce the computational requirements. Although the hidden layer is the main reason that makes neural networks a tempting choice due to their ability to represent data accurately, however by using enough training data the new models can be accurately trained in a much faster setting.

CBOW predicts a pivot word using a window of contextual words around the pivot from the same sentence. The objective of this network architecture is to classify correctly the pivot word given its context by using log linear classifiers.

While most models uses certain context to predict the current word, Skip-gram models on the other hand, uses a pivot word to predict its context by trying to maximize the probability of the contextual word given that pivot word using log linear classifier. It uses the continuous vector representation of the pivot word as an input to the classifier to predict another word in the same context within a certain window. Increasing the context window increases the model accuracy reflected in the quality of the resulting word vectors, but it increases the computation complexity.

Pennington et al. [7] presented yet another technique to learn word representations called "GloVe" for Global Vectors. While CBOW and SKIP-G models can be classified as shallow window based approaches, because they represent a word in vector space as a function of its local context controlled by a window, GloVe on the other hand utilizes the global statistics of word-word co-occurrence in the corpus to be captured by the model. The co-occurrence matrix is used to calculate the probability of $word_i$ to appear in the context of another $word_j$ $P(i|j)$, this probability is postulated to capture the relatedness between these words. For example, the word "solid" is more related to "ice" than to "steam", this can be confirmed by the ratio between P("solid"|"ice") and P("solid"|"steam") to be high. GloVe uses this ratio to encode the relationship between words and tries to find vectorized representation for words that satisfy this ratio, thus the model is built with the objective of learning vector representation for words that captures linear linguistic relationship between them.

2 Building Word Representation for Arabic

Mikolov et al. [5] compared different techniques for building word representation in vector space: Mikolov's RNN embeddings, Collobert and Weston's embeddings, Turian's embeddings and Mnih's embeddings, and showed that the CBOW and SKIP-G models are significantly faster to train with better accuracy. Pennington et al. [7] showed that GloVe performed well compared to CBOW and SKIP-G models in the semantic and syntactic analogy test presented in [5]. Accordingly, we used CBOW, SKIP-G and GloVe models to build a word representation in vector space for Modern Standard Arabic (MSA). To train these models, we collected large amount of raw Arabic texts form these sources:

- Arabic Wikipedia.
- Arabic Gigaword Corpus.
- LDC Arabic newswire.
- Arabic Wiktionary.
- The open parallel corpus [8, 9].
- Combined glosses and definitions for Arabic words in Arabase [10].
- MultiUN; which is collection of translated documents from the United Nations [11].
- OpenSubtitles 2011, 2012, and 2013. They are a collection of movie subtitles [12].
- Raw Quran text [13].
- A corpus of KDE4 localization files [14].
- A collection of translated sentences from Tatoeba [9].
- Khaleej 2004 and Watan 2004 [15].
- BBC and CNN Arabic corpus [16].
- Meedan Arabic corpus [16].
- Ksucorpus; King Saud University Corpus [18].
- A text version of Zad-Almaad book.
- Microsoft crawled Arabic Corpus.

We compiled all these sources together and performed several cleaning and normalization steps to the combined corpus:

– Cleaning noisy characters, tags and removing diacritics.
– Arabic characters normalization: we normalized (آ ,إ,أ) to (ا) and (ة) to (ه).
– Normalizing all numerical digits to the token "NUM".

We formed short phrases from individual words by attaching n-gram tokens together to be treated as a single unit [6]. To form such phrases we choose a frequency threshold above which this n-gram will be treated as a short phrase. For words w_i and w_j and their bigram $w_i w_j$:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)} \quad (1)$$

Bigrams whose score above this threshold will be used as phrases. δ is a discounting factor to prevent the formation of phrases with infrequent words.

The vocabulary size of the compiled corpus is about 6.3 million entries (unigrams and bigrams), and the total number of words is about 5.8 billion. Training these models require choice of some hyper-parameters affecting the resulting vectors:

– **Word vector size**: The vector size is an input parameter. A Couple of hundreds is the recommended choice. This parameter affects the performance of the model, which means it is useful to tune this parameter if the resulting vectors will be used in a specific task.
– **Window**: For CBOW/SKIP-G it refers to the amount of context to consider around the pivot word in training the model, while for GloVe it refers to the maximum distance between two words to be considered in co-occurrence.
– **Sample**: For CBOW/SKIP-G it refers to a threshold for occurrence of words so that words appearing with frequency higher than this threshold will be randomly down-sampled.
– **Hierarchical Softmax (HS)**: For CBOW/SKIP-G, hierarchical Softmax is a computationally efficient approximation of the full softmax used to predict words during training.
– **Negative**: For CBOW/SKIP-G it refers to the number of negative examples in the training.
– **Frequency threshold**: Words appearing with frequency less than this threshold will be discarded.
– **Maximum number of iterations**: For GloVe, it is the number of iterations used to train the model.
– **X_Max**: For GloVe, this parameter is used in a weighting function whose job is to give rare and noisy co-occurrences low weights.

We built three models for Arabic (CBOW, SKIP-G and GloVe)[1]. Table 1 shows the training details for each model.

Table 1. Training configuration parameters used to build the Arabic models

	CBOW	SKIP-G	GloVe
Vector size	300	300	300
Window	5	10	10
Sample	1e-5	1e-5	N/A
HS	No	No	N/A
Negative	10	10	N/A
Freq. thresh.	10	10	10
Phrase thresh.	200	200	200
Max iterations	N/A	N/A	25
X_Max	N/A	N/A	100

3 Vector Quality Assessment

3.1 Intrinsic Evaluation

Having the individual words represented in vector space introduces new thinking strategies in using these representations in word-to-word relations. A relationship between two words can be measured by using a similarity function that maps a pair of word vectors to a real number: $F(v_1, v_2) \rightarrow \mathbb{R}$ This mapping function (similarity measure function) can be Cosine similarity, or Euclidean distance, or Manhattan distance, or any possible similarity measure techniques.

One particular interesting task, is to apply the relationship between a pair of words (e.g. singular/plural, feminization, tense change…) to a third word, this is called "analogy task". Let the first pair of words be w_1 and w_2 and the third word be w_3. Let the relationship between w_1 and w_2 be r_1 then $v(r_1) = v(w_2) - v(w_1)$ where the operator $v(\mathrm{x})$ returns a vector representation of x, and $v(r_1)$ represents a vector in space joining w_1 and w_2. We can apply r_1 to w_3 to give a fourth word w_4 such that $v(w_4) = v(w_3) + v(r_1)$ and so w_3, w_4 have the same relationship as w_1, w_2.

To test the quality of the vectors, Mikolov's analogy test for English vectors [5] is used by translating the test cases manually to Arabic. The test set contains five types of semantic questions, and nine types of syntactic questions. Examples are shown in Table 2. We compared our models to the English skip-gram model [19] and GloVe model [20] using the translated version of the test (Table 3).

The test focuses on calculating the relation between the first pair of words and apply it to a third word, then comparing the fourth word with the predicted word. The predicted word is the word with the highest Cosine similarity score to the predicted vector.

[1] Models are available at: `https://sites.google.com/site/mohazahran/data`

Table 2. A sample of Mikolov's semantic and syntactic analogy test for English and its translation to Arabic

Type of relation ship	Word pair 1				Word pair 2			
Common capital city	Athens	أثينا	Greece	اليونان	Oslo	اوسلو	Norway	النرويج
Man-Woman	brother	شقيق	sister	شقيقة	grand son	حفيد	grand daughter	حفيدة
Superlative	bad	سيء	worst	اسوأ	big	كبير	biggest	اكبر
Plural nouns	bird	طائر	birds	طيور	car	سيارة	cars	سيارات

Table 3. Total accuracy of English models and Arabic models on the test set and its Arabic translation. The first column per model shows the percentage of test cases covered by this model. The second column shows how many of the covered test cases are correct. All numbers in the table are percentages. (Cov. is short for coverage and Acc. is short for accuracy).

Model	**English SKIP-G300**		**English GloVe300**		**Arabic CBOW300**		**Arabic SKIP-G300**		**Arabic GloVe300**	
Training words	300B		840B		5.8B		5.8B		5.8B	
	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.
capital-common-countries	100	94.9	100	**100**	100	94.3	100	93.7	100	92.7
capital-world	100	93.1	100	**95.6**	100	74.7	100	77	100	80.4
currency	100	**37.8**	100	13.2	100	7.7	100	7.9	100	5.7
city-in-state	100	87.2	100	**87.4**	100	32.5	100	32.6	100	36.4
family	100	**95.3**	100	86.8	67.6	46.5	67.6	36.3	67.6	50.3
adjective-to-adverb	100	53.8	100	**65.6**	100	34.2	100	30.1	100	22
opposite	100	**57.6**	100	45.8	80	3.7	80	3.2	80	3.5
comparative	100	**97**	100	96.6	100	73.8	100	67	100	71.5
superlative	100	**95.4**	100	93.7	100	68.9	100	64.6	100	66.9
present-participle	100	96.7	100	**97.4**	93.9	46.1	93.9	42.1	93.9	30.5
nationality-adjective	100	**95.7**	100	89.2	100	49.9	100	55.5	100	44.2
past-tense	100	**93.7**	100	91.9	100	44.7	100	41.6	100	43.5
plural	100	95.8	100	**96.5**	100	56.1	100	56.9	100	57.7
plural-verbs	100	89.5	100	**90.3**	100	80.1	100	75.5	100	72.2
TOTAL	100	**87.4**	100	86.2	98	54.3	98	53.6	98	53.5

A test case is answered correctly only if one of the top five predicted words matches the fourth word, which means that synonyms and semantically close words are considered as mistakes.

By examining the results in Table 3, the Arabic models are trained on significantly smaller corpus compared to English. Having a large corpus is essential to enable the

models to represent the words more accurately, by large corpus we mean a corpus big enough such that each word in the vocabulary is repeated an adequate number of times for the models to represent accurately. However, our models show good performance on test cases whose Arabic translation is unambiguous as translating named entities (countries, capitals, and cities) and thus they are frequent in the corpus. On the other hand, they show low performance on other test cases as in the "opposite" test cases because most of the English words in this test do not have a clear Arabic translation. For example, the word "uncompetitive" has no direct (word-to-word) Arabic translation; the closest translation will be "غير_منافس". These unusual translations are either out of vocabulary or rarely found in the training corpus and thus receive a poor vectorized representation. Low frequent terms explain as well the low performance for the "currency" test cases. For example, the term "الزلوتي" (translated as zloty (Polish currency)) occurred only 63 time.

Another source of errors is the absence of diacritization, which is needed in Arabic to differentiate between words having the same form but with different meanings, however in practice the use of diacritics in Arabic is rare and the Arabic data collected is almost free of diacritics. These results suggests that there should be a tailored test set for Arabic rather than translating the English test cases in order to evaluate the Arabic vectors more accurately.

3.2 Extrinsic Evaluation

Information Retrieval

Many query expansion techniques have been proposed to enhance the performance of text retrieval task, which can be classified into semantic-based and statistical-based expansion techniques. Here, we propose using the Arabic vectors as a semantic expansion technique because the Arabic vectors capture the semantic properties of the language such that semantically close terms are clustered in close proximity in the vector space. Mahgoub et al. [23] proposed query semantic expansion techniques for Arabic information retrieval, the expansion techniques based on various language resources as Wikipedia, Google translate with WordNet, and other various Arabic linguistic resources. We compare our vector expansion technique with their techniques using TREC 2002 the cross-lingual (CLIR) track dataset [24], which contains 50 queries tested against 383,872 documents, we discarded any non-judged documents from our experiments before evaluation. The basic idea is to expand a query term such that these expansions are semantically related to the query, which means the query should act as a sense gauge to the expanded term. A term is expanded using its vector representation to retrieve all other terms in the vector space ordered descendingly by cosine similarity score, while a query is represented by adding up all the vectors of its terms together. The order of possible expansions for a term should be influenced by the query through re-ordering the terms in the expansion list using cosine similarity score with the query vector. In order to avoid bias in re-ordering the expansion list, the term being expanded is not included among the terms forming the query vector. For each query, we allow maximum 50 expansions for all of its terms, such that the number of expansions for each term is inversely

proportional to its frequency, thus allowing less frequent terms to have more expansions [23]. Figure 1 and 2 compare between the impact of using the Arabic vectors as an expansion scheme versus traditional resources as Wikipedia, WordNet translations and other resources using Indiri [25]. The following example in Table 4 shows how the query is used to disambiguate the expansion list of a term (underlined word). The query: "كيف يعامل طلاب الدين في النجف بعد اغتيال صادق الصدر؟" translated as (How are the religion students treated in Nagaf after the assassination of Sadeq Alsadr?). The expanded word (الصدر, Alsadr) has two senses, either "chest" or the name of a person "Alsadr" (which is the correct sense for this query).

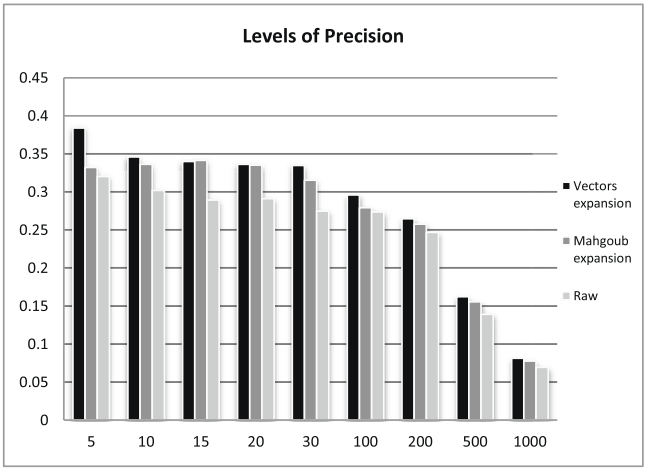


Fig. 1. Levels of precision for expansions using Arabic vectors, Mahgoub expansion using wikipedia, and other resources and raw text matching on TREC 2002

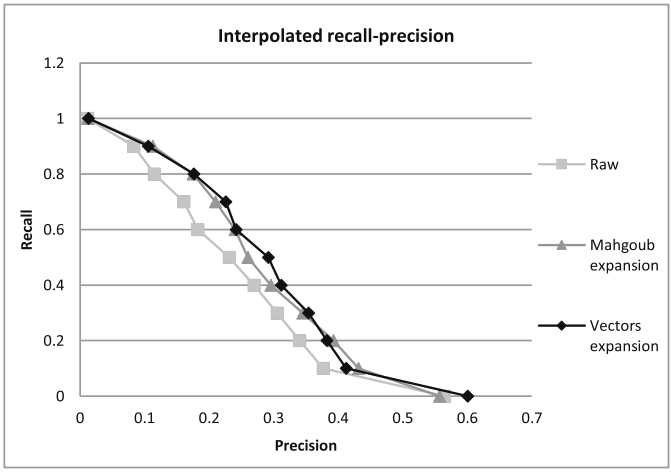


Fig. 2. Recall-precision curve for expansions using Arabic vectors, Mahgoub expansion using wikipedia, and other resources and raw text matching on TREC 2002

Table 4. A comparison between the expansion lists for the word "Alsadr" in a query before and after disambiguation using the query context vector

Before disambiguation		After disambiguation	
Arabic	**English translation**	**Arabic**	**English translation**
والصدر	And the chest	النجف	Alnagaf
للصدر	For the chest	انصار مقتدى	Supporters of Moqtada
بالصدر	By the chest	الصدر ببغداد	Alsadr in Baghdad
البطن	Stomach	مقتدي الصدر	Moqtada Alsadr

Short Answer Grading

Short answer grading is one interesting NLP application to assess how much the Arabic vectors capture semantic and syntactic properties of the language. In short answer grading, given a reference answer and a student answer; it is required to return a grade that represents the correctness of this answer. To employ the vectors in such problem it is essential to transform the grading problem into a sentence-to-sentence similarity measuring task. A sentence can be represented using a combination of the vectors of its words. For example, a simple addition of the word vectors can give a sufficient representation for a sentence in vector space especially for short sentences. Using combinations of different preprocessing steps (lemmatization, stemming …) with various vector based sentence representation schemes (CBOW, SKIP-G, GloVe) will result in a number of features relating a student answer with the reference answer via similarity measures as cosine similarity, these features are fed to an SVM regression module to scale similarity scores to a reasonable grade. Table 5 shows the impact of using the Arabic vectors on an Arabic dataset for short answer grading using root mean square error (RMSE) and Pearson's correlation against inter annotator agreement (IAA). We also report the results of Goma's system discussed in [22] on the equivalent humanly translated English data set.

Table 5. Arabic vectors results in short answer grading using RMSE (the lower the better) and correlation (the higher the better)

	RMSE	**Correlation**
IAA (Arabic and English Dataset)	0.69	0.86
Arabic vectors (Arabic Dataset)	0.95	0.82
Goma's system (Manual English translations data set)	0.75	0.83

4 Arabic-English Vector Space Mapping

Mapping the Arabic vector space to the English vector space is an attractive application especially for Arabic, because Arabic suffers from poor language resources support as compared to English. Mapping the two vector spaces will allow Arabic NLP applications to use the English language support in Arabic NLP domain.

Mikolov et al. [21] used a translation matrix to learn linear transformation between the two vector spaces by minimizing the mean square error between the reference and the predicted vectors, this translation matrix can be regarded as a simple neural network with no hidden layers. Alternatively, we propose training a neural network to learn vector mapping by minimizing the Cosine error instead of minimizing the mean square error (derivations in the appendix). The intuition behind this objective function is the use of Cosine similarity score in literature as the default measure to assess the similarity between two word vectors [5, 6, 7, 21], which means there is a mismatch between the objective function (mean square error) and the similarity metric (Cosine similarity score). We used an English-Arabic dictionary to train the neural network by retrieving vectors corresponding to the dictionary's entries to form parallel training data. Each entry consists of an Arabic word with a list of possible English translations totaling 8,444 entries divided into 5,872 entries for training, 1,254 for validation and 1,318 for testing. The training and validation entries are expanded such that an Arabic word form a different entry with each of its possible English translations, resulting in 27,089 entries for training and 5,718 entries for validation. We used vectors from the best scoring models in Table 3, CBOW300 for Arabic and SKIP-G300 for English.

Two simple neural networks are constructed with 300 neurons for both input and output layers with no hidden layers. Both networks have the same architecture, parameters and initial weights, the only difference is the objective functions, the first optimizes for Cosine error, the second optimizes for mean square error and both are trained using backpropagation. Some translation examples from the test set using the Cosine neural network are shown in Table 6.

Table 7 compares between the two networks using three measures, NDCG, recall and accuracy for the test set. For each instance in the test set, we use the predicted English vector to retrieve the top k English translations using Cosine similarity. The NDCG for test sample j using k predicted translations:

$$NDCG_{\mathrm{j}} = \frac{match(1) + \sum_{\mathrm{i}=2}^{\mathrm{k}} \frac{match(i)}{log_2 i}}{1 + \sum_{\mathrm{i}=2}^{\mathrm{k}} \frac{1}{log_2 i}} \tag{2}$$

Where the function $match$ takes a predicted translation and checks if it matches one of the possible reference English translations with no accounting for synonyms, semantically close terms and even morphological variations.

$$match(\mathrm{x}) = \begin{cases} 1, & \text{if } x \text{ is a match.} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The overall NDCG for M test sentences using k predicted translations is:

$$NDCG = \frac{\sum_{\mathrm{j}=1}^{\mathrm{M}} NDCG_{\mathrm{j}}}{M} \tag{4}$$

While the NDCG accounts for the rank k at which a match happens, the recall and accuracy on the other hand takes no regard to the rank. The accuracy calculates how many test samples were translated correctly by matching at least one of the k

predicted translations with one of the reference translations, while the recall calculates how many reference translations were covered by the predicted translations. The recall for a test sample j with l reference translations and k predicted translations:

$$Recall_{\mathrm{j}} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{k}} match(i)}{Min(k,l)} \tag{5}$$

The overall recall for M test sentences using k predicted translations is:

$$Recall = \frac{\sum_{\mathrm{j}=1}^{\mathrm{M}} Recall_{\mathrm{j}}}{M} \tag{6}$$

Table 6. Translation examples using Cosine neural network

Arabic Word	English translations using Cosine neural network
عصور	epochs,epoch,millenniums,millennia,eras,era
علم_الفلك	astronomy,cosmology,quantum_physics,astrology,science
الكونجرس	parliament,congress,legislature,senate,vote,Congress
يتسلل	infiltrate,sneak,penetrate,seep,slither,creep

Table 7. A comparison between optimizing for Cosine error versus mean square error using NDCG, Recall and Accuracy

k	Cos (NDCG)	MSE (NDCG)	Cos (Recall)	MSE (Recall)	Cos (Accuracy)	MSE (Accuracy)
1	27.1%	25.9%	27.1%	25.9%	27.1%	25.9%
2	19.3%	19.1%	22%	21.7%	34.7%	34.4%
3	16.8%	16.5%	20.8%	20.1%	38.8%	38.5%
4	15.2%	14.9%	20.1%	19.4%	41.4%	40.4%
5	14%	13.8%	20.2%	19.6%	43.6%	43.1%
6	13.1%	12.8%	20.6%	19.6%	45.8%	44.8%
7	12.4%	12.2%	21.2%	20.3%	47.8%	46.7%
8	11.8%	11.5%	21.8%	20.8%	49.3%	48.1%
9	11.2%	11%	22.3%	21.1%	50.2%	49%
10	10.7%	10.5%	22.6%	21.7%	50.8%	50.4%

Another way to assess the quality of the neural network is to compare the translated vectors with the native English vectors in a practical NLP task as in short answer grading. The idea is having an Arabic datasets for short answer grading and its human translation to English. Given an Arabic student answer a_{s} and its Arabic reference answer a_{r} and their corresponding student and reference English answers e_{s} and e_{r} respectively. First, we assign a grade to the English answers using the English vectors following the same ideas discussed in the previous section. Second, using the proposed neural network we translate the two Arabic answer vectors to English vectors (e'_{s} and e'_{r}) and assign a grade to them using the translated vectors. Finally, we compare between the Arabic vectors on the Arabic data, the translated vectors on the English data, and the English vectors on English data using the Pearson's

correlation between the predicted grades of each system with the reference grade as shown in Table 8.

Table 8. Correlation between predicted grades and reference grades for Native Arabic, Translated English and Native English for short answer grading using word vectors

Model	Correlation
Arabic vectors & Arabic data	0.79
Translated English Vectors & Human translated English data	0.75
English vectors & Human translated English data	0.76

These results show that the translated english vectors using the neural network performs remarkably closer to the native english vectors on the same data set (Human translated english data set) than to Arabic vectors on Arabic data.

5 Conclusion and Future Work

In this paper, we compared between different models for building continuous representation in vector space for Arabic and tested these vectors via intrinsic and extrinsic evaluations. In intrinsic evaluation, we used the analogy task to test the vectors' capability to capture semantic and syntactic properties for Arabic. While in extrinsic evaluations, we employed the vectors in two NLP applications: query expansion for information retrieval and short answer grading. For the query expansion, the Arabic vectors enhanced the retrieval process slightly better than other semantic expansion techniques, while for short answer grading, Arabic vectors made it possible to evaluate short answer grades for Arabic dataset without the need for Arabic-English translations.

We built a neural network to map Arabic vectors to English vectors and showed that minimizing for cosine error outperforms the standard mean square error minimization for word-to-word similarity using cosine score, which means that the objective function of the training procedure should match the similarity measure used. Using the proposed neural network, we succeed to achieve humanly translated-like results for short answer grading task. Many extensions can be related to the work presented here, starting with increasing the raw Arabic data used to train the vectorized representations. It will also be useful to have an analogy test built specifically for Arabic rather than the manual translation of the English test. We showed a simple technique for word sense disambiguation in the context of query expansion using the Arabic vectors, yet even more sophisticated techniques can be developed. Using deep neural networks, it can be possible to learn complex relations to map the Arabic and English vector spaces more accurately.

Acknowledgments. We would like to thank Microsoft Advanced Technology Lab in Cairo, Egypt for supporting this work with data and computing resources.

References

1. Collobert, R., Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 160–167 (2008)
2. Mnih, A., Hinton, G.: A Scalable Hierarchical Distributed Language Model. In: NIPS: Proceedings of Neural Information Processing Systems, Vancouver, B.C, Canada, pp. 1081–1088 (2009)
3. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: NAACL-HLT: Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
4. Turian, J., Ratinov, L., Bengio, Y.: Word representations: A simple and general method for semi-supervised learning. In: ACL: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
5. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA, pp. 1301–3781 (2013)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representation of Words and Phrases and their Compositionality. In: NIPS: Proceedings of Neural Information Processing Systems Nevada, United States, pp. 3111–3119 (2013)
7. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: EMNLP: Proceeding of the Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 1532–1543 (2014)
8. http://opus.lingfil.uu.se/ (accessed January 29, 2015)
9. Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: LREC: Proceedings of the 8th International Conference on Language Resources and Evaluation, Istanbul, Turkey, pp. 2214–2218 (2012)
10. Raafat, H., Zahran, M., Rashwan, M.: Arabase A Database Combining Different Arabic Resources with Lexical and Semantic Information. In: Proceeding of KDIR is part of IC3K, The International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Portugal, pp. 233–240 (2013)
11. Eisele, A., Chen, Y.: MultiUN: A Multilingual corpus from United Nation Documents. In: LREC: Proceeding of the International Conference on Language Resources and Evaluation, Valletta, Malta, pp. 17–23 (2010)
12. http://www.opensubtitles.org/ (accessed January 29, 2015)
13. http://tanzil.net/download/ (accessed January 29, 2015)
14. Tiedemann, J.: News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In (RANLP): Recent Advances in Natural Language Processing, pp. 237–248. John Benjamins, Amsterdam (2009)
15. https://sites.google.com/site/mouradabbas9/corpora (accessed January 29, 2015)
16. Saad, M.K., Ashour, W.: OSAC: Open Source Arabic Corpus. In: EEECS: the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, European University of Lefke, Cyprus, vol. 10 (2010)
17. https://github.com/anastaw/Meedan-Memory (accessed January 29, 2015)
18. http://ksucorpus.ksu.edu.sa/ar/ (accessed January 29, 2015)
19. https://code.google.com/p/word2vec/ (accessed January 29, 2015)
20. http://nlp.stanford.edu/projects/glove/ (accessed January 29, 2015)

21. Mikolov, T., Le, V.Q., Sutskever, I.: Exploiting Similarities among Languages for Machine Translation. In: arXiv, 1309-4168 (2013)
22. Gomaa, W.H., Fahmy, A.A.: Automatic scoring for answers to Arabic test questions. Computer Speech & Language, 833–857 (2014)
23. Mahgoub, Y.A., Rashwan, A.M., Raafat, H., Zahran, A.M., Fayek, B.M.: Semantic Query Expansion for Arabic Information Retrieval. In: EMNLP: The Arabic Natural Language Processing Workshop, Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 87–92 (2014)
24. Oard, D.W., Gey, F.C.: The TREC 2002 Arabic/English CLIR Track. In: TREC (2002)
25. `http://sourceforge.net/p/lemur/wiki/Indri/` (accessed January 31, 2015)

Appendix

The objective function is to maximize the cosine similarity between the predicted vector (y) and the reference vector (d). This is equivalent to:

$$Minimize\ E = 1 - \cos(y, d) = 1 - \frac{y.d}{|y||d|}$$

Let the activation function be $y_i^z = F(x_i^z)$. The superscript denotes the layer number, and the subscript denotes the input number. The notation $l(z)$ refers to the number of neurons in the layer z. The derivative of the error function E with respect to weights at layer z for the training sample m:

$$\frac{\partial E}{\partial w_{IJ}^z} = \frac{\partial E}{\partial y_I^{z+1}} \times \frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} \times \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} = \delta_I^{z+1} \times \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} \tag{7}$$

$$where,\ \delta_I^{z+1} = \frac{\partial E}{\partial y_I^{z+1}} \times \frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} \tag{8}$$

$$\frac{\partial E}{\partial y_I^{z+1}} = \frac{(y.d)y_I^{z+1} - d_I^{z+1}|y|^2}{|d||y|^3} \tag{9}$$

$$y_I^{z+1} = f(x_I^{z+1})$$

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right)$$

$$\frac{\partial y_I^{z+1}}{\partial x_I^{z+1}} = f'(x_I^{z+1}) = \left(1.7159 \times \frac{2}{3}\right)\left(1 - \left(\frac{y_I^{z+1}}{1.7159}\right)^2\right) \tag{10}$$

$$x_I^{z+1} = \sum_{j=1}^{l(z)} w_{Ij}^z\, y_j^z \quad \Rightarrow \quad \frac{\partial x_I^{z+1}}{\partial w_{IJ}^z} = y_J^z \tag{11}$$

Finally $\frac{\partial E}{\partial w_{IJ}^z}$ is calculated by substituting from (8), (9), (10) and (11) in (7).

Short Text Hashing Improved by Integrating Multi-granularity Topics and Tags

Jiaming Xu, Bo Xu, Guanhua Tian, Jun Zhao, Fangyuan Wang, and Hongwei Hao

Institute of Automation, Chinese Academy of Sciences. 100190, Beijing, P.R. China
{jiaming.xu,boxu,guanhua.tian,fangyuan.wang,hongwei.hao}@ia.ac.cn, jzhao@nlpr.ia.ac.cn

Abstract. Due to computational and storage efficiencies of compact binary codes, hashing has been widely used for large-scale similarity search. Unfortunately, many existing hashing methods based on observed keyword features are not effective for short texts due to the sparseness and shortness. Recently, some researchers try to utilize latent topics of certain granularity to preserve semantic similarity in hash codes beyond keyword matching. However, topics of certain granularity are not adequate to represent the intrinsic semantic information. In this paper, we present a novel unified approach for *short text Hashing using Multi-granularity Topics and Tags*, dubbed HMTT. In particular, we propose a selection method to choose the optimal multi-granularity topics depending on the type of dataset, and design two distinct hashing strategies to incorporate multi-granularity topics. We also propose a simple and effective method to exploit tags to enhance the similarity of related texts. We carry out extensive experiments on one short text dataset as well as on one normal text dataset. The results demonstrate that our approach is effective and significantly outperforms baselines on several evaluation metrics.

Keywords: Similarity Search, Hashing, Topic Features, Short Text.

1 Introduction

With the explosion of social media, numerous short texts become available in a variety of genres, e.g. tweets, instant messages, questions in Question and Answer (Q&A) websites and online advertisements [6]. In order to conduct fast similarity search in those massive datasets, hashing, which tries to learn similarity-preserving binary codes for document representation, has been widely used to accelerate similarity search. Unfortunately, many existing hashing methods based on keyword feature space usually fail to fully preserve the semantic similarity of short texts due to the sparseness of the original feature space. For example, there are three short texts as follows:

$d1$: "*Rafael Nadal missed the Australian Open*";
$d2$: "*Roger Federer won Grand Slam title*";
$d3$: "*Tiger Woods broke numerous golf records*".

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 444–455, 2015.
DOI: 10.1007/978-3-319-18111-0_33

Obviously, the hashing methods based on keyword space cannot see the similarity among $d1$, $d2$ and $d3$. In recent years, some researchers seek to address the challenge by latent semantic approach. For example, Wang et al. [12] preserve the semantic similarity of documents in hash codes by fitting the topic distributions, and Xu et al. [14] directly treat the latent topic features as tokens to represent one document for hashing learning. However, topics of certain granularity are not adequate to represent the intrinsic semantic information [4]. As we know, different topic models with pre-defined number of topics can extract different semantic level topics. For example, the topic model with a large number of topics can extract more fine grained topic features, such as "Tennis Open Progress" for $d1$ and $d2$, and "Golf Star News" for $d3$, but fail to construct the semantic relevance of $d3$ with the other texts, and the topic model with a few topics can extract more coarse grained semantic features, such as "Sport" and "Star" for $d1$, $d2$ and $d3$, but lack distinguishing information and cannot learn the hashing function effectively, As a reasonable assumption, multi-granularity topics are more suitable to preserve semantic similarity and learn hashing function for short text hashing.

On the other hand, tags are not fully utilized in many hashing methods. Actually, in various real-world applications, documents are often associated with multiple tags, which provide useful knowledge in learning effective hash codes [12]. For instance, in Q&A websites, each question has category labels or related tags assigned by its questioner. Another example is microblog, some tweets are labeled by their authors with hashtags in the form of "#keyword". Thus, we should fully exploit the information contained in tags to strengthen the semantic relationship of related texts for hashing learning.

Based on the above observations, this paper proposes a unified *short text Hashing using Multi-granularity Topics and Tags*, referred as HMTT for simplicity. In HMTT, two different ways are introduced to incorporate multi-granularity topics and tag information for improving short text hashing.

The main contributions of this paper are three-fold: Firstly, a novel unified short text hashing is proposed. To our best knowledge, this is the first time of incorporating multi-granularity topics and tags into a unified hashing approach, and experiments are conducted to verify our assumption that short text hashing can be improved by integrating multi-granularity topics and tags. Secondly, the optimal multi-granularity topics can be selected automatically, i.e., to extract effective latent topic features for hashing learning. The experimental results indicate the optimal multi-granularity topics can achieve better performances, compared with other multi-granularity topics. Finally, two strategies to incorporate multi-granularity topics for short text hashing are designed and compared through extensive experimental evaluations and analyses.

2 Related Work

Hash-based methods can be mainly divided into two categories. One category is data-oblivious hashing. As the most popular hashing technique, Locality-

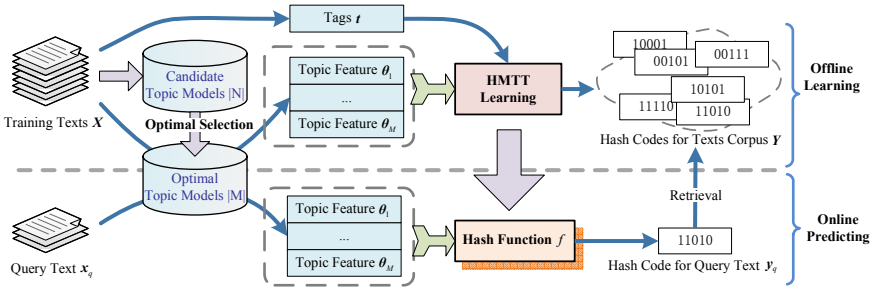


Fig. 1. The proposed approach HMTT for short text hashing

Sensitive Hashing (LSH) [1] based on random projection has been widely used for similarity search. However, since they are not aware of data distribution, those methods may lead to generate quite inefficient hash codes in practice [16]. Recently, more researchers focus attention on the other category, data-aware hashing, For example, the Spectral Hashing (SpH) [13] generates compact binary codes by forcing the balanced and uncorrelated constraints into the learned codes. Self-Taught Hashing (STH) [18] and Two Step Hashing (TSH) [9] decompose the learning procedure into two steps: generating binary code and learning hash function, and a supervised version of STH is proposed in [16] denoted as STHs. However, the previous hashing methods, directly working in keyword feature space, usually fail to fully preserve semantic similarity. More recently, Wang et al. [12] proposed a Semantic Hashing using Tags and Topic Modeling (SHTTM). However, the limitations of SHTTM are that: Although the topic distributions are used to preserve the content similarity to generate hash codes, they do not utilize the topics to improve hashing function learning; Even the number of topics must keep consistent with dimensions of hash code, that this assumption is too strict to capture the optimal semantic features for different types of datasets.

3 Algorithm Description

A unified short text hashing approach HMTT is depicted in Fig. 1. Given a dataset of n training texts denoted as: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{d \times n}$, where d is the dimensionality of the keyword feature. Denote their tags as: $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n\} \in \{0, 1\}^{q \times n}$, where q is the total number of possible tags associated with each text. A tag with label 1 means a text is associated with a certain tag/category, while a tag with label 0 means a missing tag or the text is not associated with that tag/category. The goal of HMTT is to obtain optimal binary codes $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}^T \in \{-1, 1\}^{n \times l}$, and a hashing function f: $\mathbb{R}^d \rightarrow \{-1, 1\}^l$, which embeds the query text $\mathbf{x}_q$ to its binary vector representation $\mathbf{y}_q$ with l bits. To achieve the similarity-preserving property, we require the similar texts to have similar binary codes in Hamming space. We first select the optimal topic models from the candidate topic models, and extract the multi-granularity topic features $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$. Then the

Algorithm 1. The Optimal Topics Selection

Input: n training texts $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ with tags $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n\}$, N candidate topic sets $\mathbf{T} = \{T_1, T_2, ..., T_N\}$ and a specified number M.
Output: The optimal topic sets $\mathbf{O}$, and the weight vector $\boldsymbol{\mu}$.

```
Sample a sub-set X̂ with tags t̂; Initialize μ ← 0, and O ← ∅;
for each text x̂ ∈ X̂ do
    Find nn⁺(x̂) and nn⁻(x̂);
    for i ← 1 to N do
        Update μ(T_i) by Eq. 1;
    end for
end for
while size(O) < M do
    T^(p) = arg max_{T_i ∈ T} μ(T_i); Update O = O ∪ {T^(p)}, T = T − {T^(p)};
end while
return O and μ;
```

binary codes and hash functions can be learned by integrating multi-granularity topic features and tags. In the second phase which is online, the query text is represented by binary code mapped from the derived hash function, and then the approximate nearest neighbor search is accomplished in Hamming space. All pairs of hash code found within a certain Hamming distance of each other are semantic similar texts.

The main challenges of the idea are that: (1). How to select the optimal topic models; (2). How to utilize the tag information efficiently; and (3). How to integrate the multi-granularity topics to preserve semantic similarity. The proposed approach HMTT will be described in detail in the following sections.

3.1 Estimate and Select the Optimal Topics

In this work, we straightforwardly obtain a set of candidate topics by pre-defining several different topic numbers of Latent Dirichlet Allocation (LDA) [3]. After training the topic models, we can draw multi-granularity topic features, corresponding as distributions over the topics, from the candidate topic models.

In order to select the optimal topic models, we should utilize the tag information to evaluate the quality of topics. Inspired by [4,7], the selection of optimal topic model sets depends on their capability in helping discriminate short texts without sharing any common tags. We denote N different sets of topics as $\mathbf{T} = \{T_1, T_2, ..., T_N\}$. For each entry T_i, the probability topics distributions over documents are denoted as $\boldsymbol{\theta} = p(\mathbf{z}|\mathbf{x})$. The weight vector is $\boldsymbol{\mu} = \{\mu(T_1), \mu(T_2), ..., \mu(T_N)\}$, where $\mu(T_i)$ is the weight indicating the importance of topic set. The purpose is to select the optimal topic sets $\mathbf{O} = \{T_1, T_2, ..., T_M\}$. In [4], Chen et al. evaluate the quality of topics based on two aspects: discrimination and complementarity of the multi-granularity topics. However, how to balance those two aspects is a tricky problem and the latter aspect, complementarity, is easy to introduce noises for preserving similarity. Thus, we propose a simple and effective method directly based on the key idea of Relief [7] as

follows: Firstly, a sub-set $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_m\}$ with tags $\hat{\mathbf{t}} = \{\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, ..., \hat{\mathbf{t}}_m\}$ is sampled from training dataset, and we find two groups of k nearest neighbors for each text $\hat{\mathbf{x}}_i$: one group is from the texts sharing any common tags (denoted as $nn^+(\hat{\mathbf{x}})$), and the other from the texts not sharing any common tags (denoted as $nn^-(\hat{\mathbf{x}})$). Then the weight is updated as follows:

$$\mu(T_i) = \mu(T_i) + \sum_{j=1}^{k} \frac{D_{KL}(T_i(\mathbf{x}), T_i(nn_j^-(\mathbf{x})))}{k} - \sum_{p=1}^{k} \frac{D_{KL}(T_i(\mathbf{x}), T_i(nn_p^+(\mathbf{x})))}{k} \quad (1)$$

where, D_{KL} is the symmetric Kullback-Leibler (KL) divergence:

$$\begin{aligned} D_{KL}(T_i(\mathbf{x}), T_i(nn_j^-(\mathbf{x}))) = \tfrac{1}{2} \sum_{z_k \in T_i} (p(z_k|\mathbf{x}) \cdot log(\tfrac{p(z_k|\mathbf{x})}{p(z_k|nn_j^-(\mathbf{x}))}) \\ + p(z_k|nn_j^-(\mathbf{x})) \cdot log(\tfrac{p(z_k|nn_j^-(\mathbf{x}))}{p(z_k|\mathbf{x})})), \end{aligned}$$

so is the value of $D_{KL}(T_i(\mathbf{x}), T_i(nn_p^+(\mathbf{x})))$. After updating the weight vector, we directly select the optimal topic sets $\mathbf{O}$ according to the top-M weight values. In summary, the optimal topics selection procedure is depicted in Algorithm 1.

3.2 Content Similarity and Tags Preservation

In hashing problem, one key component is how to define the affinity matrix $\mathbf{S}$. Diverse approaches can be applied to construct the similarity matrix. In this paper, we choose cosine function as an example and use the local similarity structure of all text pairs to reconstruct the similarity function as follows:

$$S_{ij} = \begin{cases} c_{ij} \cdot \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}, \; if \; \mathbf{x}_i \in \mathbf{NN}_k(\mathbf{x}_j) \; or \; vice \; versa \\ 0, \qquad\qquad otherwise \end{cases} \quad (2)$$

where $\mathbf{NN}_k(\mathbf{x})$ represents the set of k-nearest-neighbors of $\mathbf{x}$, and c_{ij} is an confidence coefficient. If two documents $\mathbf{x}_i$ and $\mathbf{x}_j$ share any common tag, we set c_{ij} a higher value a. In reverse, the c_{ij} is given a lower value b if two documents $\mathbf{x}_i$ and $\mathbf{x}_j$ are not related. The parameters a and b satisfy $1 \geq a \geq b > 0$. For a particular dataset, the more trustworthy the tags are, the greater difference between a and b we set. In our experiments, we set $a = 1$ and $b = 0.1$.

3.3 Learning to Hash with Multi-level Topics

Below, from different perspectives, we propose two strategies to integrate multi-granularity topics for improving short text hashing.

Feature-Level Fusion. In order to integrate multi-granularity topics, we here adopt a simple but powerful way to combine observed features and latent features for short text, similar as [10] and [4], and create a high dimensional vector $\mathbf{\Omega}$ as:

$$\mathbf{\Omega} = [\hat{\mu}_1 \boldsymbol{\theta}_1, \hat{\mu}_2 \boldsymbol{\theta}_2, ..., \hat{\mu}_M \boldsymbol{\theta}_M], \quad (3)$$

Algorithm 2. Feature-Level Fusion Procedure

Input: A set of n training texts $\mathbf{X}$ with tags $\mathbf{t}$, M optimal topic models $\mathbf{O}$ associated with their weight vector $\hat{\boldsymbol{\mu}}$.

Output: The optimal hash codes $\mathbf{Y}$ and the hash function: l linear SVM classifiers.

1: Extract M topic feature sets $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$ from the optimal topic models $\mathbf{O}$;
2: Produce the new feature $\boldsymbol{\Omega}$ by Eq. 3 and construct confidence matrix $\mathbf{S}$ by Eq. 2;
3: Obtain the l-dimensional vectors $\tilde{\mathbf{Y}}$ by optimizing Eq. 5;
4: Generate $\mathbf{Y}$ by thresholding $\tilde{\mathbf{Y}}$ to the median vector $\mathbf{m} = median(\tilde{\mathbf{Y}})$;
5: Train l linear SVM classifiers by the learned codes $\mathbf{Y}$;
6: **return** Hash codes $\mathbf{Y}$ and l linear SVM;

where, $\{\theta_1, \theta_2, ..., \theta_M\}$ are the optimal topic features, and

$$\hat{\mu}_i = \mu_i(T_i)/min_{T_k \in \boldsymbol{O}}(\mu_k(T_k)). \tag{4}$$

We can straightforwardly construct the similarity matrix $\mathbf{S}$ by Eq. 2 with the new features $\boldsymbol{\Omega}$ of training texts. Similar as Two-Step Hashing (TSH) [9], we see the binary code generation and hash function learning process as two separate steps. As a special example, Laplacian affinity loss and linear SVM are chosen to solve our problem. In first step, the training hash codes procedure can be formulated as following optimization:

$$\begin{aligned} &\min_{\mathbf{Y}} \sum_{i,j=1}^{n} S_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|_F^2 \\ &s.t. \mathbf{Y} \in \{-1, 1\}^{n \times l}, \mathbf{Y}^T \mathbf{1} = 0, \mathbf{Y}^T \mathbf{Y} = \mathbf{I} \end{aligned} \tag{5}$$

where S_{ij} is the pairwise similarity between documents $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{y}_i$ is the hash code for $\mathbf{x}_i$, and $\|\cdot\|_F$ is the Frobenius norm. To satisfy the similarity preservation, we seeks to minimize the quantity, because it incurs a heavy penalty if two similar documents are mapped far away. The problem is relaxed by discarding $\mathbf{Y} \in \{-1, 1\}^{n \times l}$, the optimal l-dimensional real-valued vector $\tilde{\mathbf{Y}}$ can be obtained by solving Laplacian Eigenmaps problem [2]. Then, $\tilde{\mathbf{Y}}$ can be converted into binary codes $\mathbf{Y}$ via the media vector $\mathbf{m} = median(\tilde{\mathbf{Y}})$. In hash function learning step, thinking of each bit $y_i^{(p)} \in \{+1, -1\}$ in the binary code as a binary class label for that text, we can train l linear SVM classifiers $f(\mathbf{x}) = sgn(\mathbf{W}^T \mathbf{x})$ to predict the l-bit binary code for any query document $\mathbf{x}_q$. Algorithm 2 shows the procedure of this strategy.

Decision-Level Fusion. From another perspective, we can treat the optimal multi-granularity topic feature sets $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$ extracted from short texts as multi-view features. In our situation, there are M-view features: $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$. We take a linear sum of those M-view similarities as follows:

$$\sum_{k=1}^{M} \sum_{i,j=1}^{n} S_{ij}^{(k)} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|_F^2 \tag{6}$$

where, $S_{ij}^{(k)}$ constructed as Eq. 2 is the affinity matrix defined on the k-th view features. By introducing a diagonal $n \times n$ matrix $\mathbf{D}^{(k)}$ whose entries are given

Algorithm 3. Decision-Level Fusion Procedure

Input: A set of n training texts $\mathbf{X}$ with tags $\mathbf{t}$, M optimal topic models $\mathbf{O}$ and trade-off parameters, C_1 and C_2.

Output: The optimal hash codes $\mathbf{Y}$ and a set of linear hash function matrices $\tilde{\mathbf{W}}$.

1: Extract M topic feature sets $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$ from the optimal topic models $\mathbf{O}$;
2: Construct a series of confidence matrices $\{\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, ..., \mathbf{S}^{(M)}\}$ by Eq. 2 for M feature sets: $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$;
3: Obtain the l-dimensional vectors $\tilde{\mathbf{Y}}$ and $\tilde{\mathbf{W}}$ by optimizing Eq. 7;
4: Generate $\mathbf{Y}$ by thresholding $\tilde{\mathbf{Y}}$ to the median vector $\mathbf{m} = median(\tilde{\mathbf{Y}})$;
5: **return** Hash codes $\mathbf{Y}$ and hash function matrix set $\tilde{\mathbf{W}}$;

by $D_{ii}^{(k)} = \sum_{j=1}^{n} S_{ij}^{(k)}$, Eq. 6 can be rewritten as $tr(\mathbf{Y}^T \sum_{k=1}^{M} (\mathbf{D}^{(k)} - \mathbf{S}^{(k)})\mathbf{Y}) = tr(\mathbf{Y}^T \sum_{k=1}^{M} \mathbf{L}^{(k)}\mathbf{Y})$, where $\mathbf{L}^{(k)}$ is the Laplacian matrix defined on the k-th view features. By introducing Composite Hashing with Multiple Information Sources (CHMIS) [15], as a representative of Multiple View Hashing (MVH), we can simultaneously learn the hash codes $\mathbf{Y}$ of the training texts $\mathbf{X}$ as well as a set of linear hash functions $\sum_{k=1}^{M} \alpha_k (\mathbf{W}^{(k)})^T \mathbf{X}^{(k)}$ to infer the hash code for query text $\mathbf{x}_q$. The overall objective function is given as follows:

$$\begin{aligned} &\min_{\mathbf{Y},\mathbf{W},\alpha} C_1 tr(\mathbf{Y}^T \sum_{k=1}^{M} \tilde{\mathbf{L}}^{(k)}\mathbf{Y}) + C_2 \left\| \mathbf{Y} - \sum_{k=1}^{M} \alpha_k (\mathbf{W}^{(k)})\mathbf{X}^{(k)} \right\|_F^2 + \sum_{k=1}^{M} \left\|\mathbf{W}^{(k)}\right\|_F^2 \\ &s.t. \mathbf{Y} \in \{-1, 1\}^{n \times k}, \mathbf{Y}^T\mathbf{1} = 0, \mathbf{Y}^T\mathbf{Y} = \mathbf{I}, \alpha^T\mathbf{1} = 1, \alpha \geq 0 \end{aligned} \tag{7}$$

where, C_1 and C_2 are trade-off parameters, $tr(\cdot)$ is the matrix trace function, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_M]$ is a combination coefficient vector to balance the outputs from each view features, and a series of linear hash function matrices: $\tilde{\mathbf{W}} = \{\alpha_1\mathbf{W}^{(1)}, \alpha_2\mathbf{W}^{(2)}, ..., \alpha_M\mathbf{W}^{(M)}\}$. In order to solve this hard optimization problem, we first relax the discrete constraints $\mathbf{Y} \in \{-1, 1\}^{n \times l}$, and iteratively optimize one variable with the other two fixed. More detailed optimization procedures of this method can be found in [15]. Different from the former strategy, we do not need to pre-allocate the weight value of each view features, because that the combination coefficient vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_M]$ learned iteratively in the process of optimization can balance the outputs of each view features, and the procedure of this strategy is shown in Algorithm 3.

3.4 Complexity Analysis

The training processes including binary code learning and hash function training are always conducted off-line. Thus, our focus of efficiency is on the prediction process. This process of generating hash code for a query text only involves some Gibbs sampling iterations to extract multi-granularity topics $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_M\}$ and dot products in hash function $\mathbf{y} = sgn(\mathbf{W}^T\mathbf{x})$, which can be done in $O(r\tilde{K}s + l\tilde{K})$. Here, r is the number of Gibbs sampling iterations for topic

inference, $\tilde{K}$ is the sum of multi-granularity topic numbers $\{K_1, K_2, ..., K_M\}$, l is the dimensionality of hash code and s denotes the sparsity of the observed keyword features. The values of the parameters above can be regarded as quite small constants. For example, $r = 20, \tilde{K} \approx 100, l \leq 64$ and the average number of sparsity per document s is no more than 100 in our experimental datasets. We can see the major time complexity is the Gibbs sampling for topic inference. In recent works, lots of studies focus to accelerate the topic inference. For example, in Biterm Topic Model (BTM), [5] gives a simplicity and efficient method without Gibbs sampling iterations and the time complexity for topic inference can be reduced to $O(Kb)$, where b is the number of biterms in a query text.

4 Experiment and Analysis

4.1 Dataset and Experimental Settings

We carried out extensive experiments on two publicly available real-world text datasets: one is typical short text dataset, *Search Snippets*[1], and another is normal text dataset, *20Newsgroups*[2].

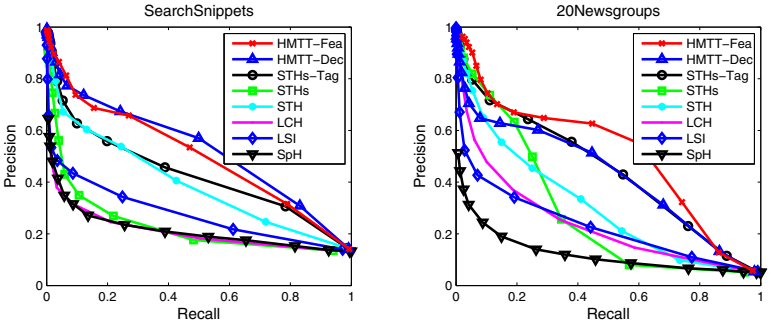
The **Search Snippets** dataset collected by Phan [10] was selected from the results of web search transaction using predefined phrases of 8 different domains. We further filter the stop words and stem the texts. 20139 distinct words, 10059 training texts and 2279 test texts are left, and the average text length is 17.1.

The **20Newsgroups** corpus was collected by Lang [8]. We use the popular 'bydate' version which contains 20 categories, 26214 distinct words, 11314 training texts and 7532 test texts, and the average text length is 136.7.

For these datasets, we denote the category labels as tags. For *Search Snippets*, we use a large-scale corpus [10] crawled from Wikipedia to estimate the topic models, and the original keyword features are directly used for learning the candidate topic models for *20Newsgroups* due to the sufficient keyword features. In order to evaluate our method's performance, we compute standard retrieval performance measures: recall and precision, by using each document in the test set as a query to retrieve documents in the training set within a specified Hamming distance. For the original keyword feature space cannot well reflect the semantic similarity of documents, even worse for short text, we simply test if the two documents share any common tag to decide whether a semantic similar text. This methodology is used in SH [11], STH [18], CHMIS [15] and SHTTM [12].

Five alternative hashing methods compared with our proposed approach are STHs [16], STH [18], LCH [17], LSI [11] and SpH [13]. The results of all baseline methods are obtained by the open-source implementation provided on their corresponding author's homepage. In order to distinguish the proposed two strategies in our approach, the feature level fusion method is denoted as HMTT-Fea, and the decision level fusion method is named as HMTT-Dec[3].

[1] `http://jwebpro.sourceforge.net/data-web-snippets.tar.gz`
[2] `http://people.csail.mit.edu/jrennie/20Newsgroups/`
[3] `https://github.com/jacoxu/short-text-hashing-HMTT`, `http://www.CICLing.org/2015/data/148`



(a) SearchSnippets (b) 20Newsgroups

Fig. 2. Precision-Recall curves of retrieved examples within Hamming radius 3 on two datasets with different hashing bits (4:4:64 bits)

In our experiments, the candidate topic sets $\mathbf{T} = \{T10, T30, T50, T70, T90, T120, T150\}$ and the number of the optimal topic sets is fixed to 3. The parameters C_1 and C_2 in Eq. 7 are tuned from $\{0.1, 1, 10, 100\}$. The number of nearest neighbors is fixed to 25 when constructing the graph Laplacians in our approach, as well as in the baseline methods, STHs and STH. We evaluate the performance of different methods by varying the number of hashing bits from 4 to 64. For LDA, we used the open-source implementation GibbsLDA[4], and the hyper-parameters are tuned as $\alpha = 0.5$, $\beta = 0.01$, 1000 iterations of Gibbs sampling for learning, and 20 iterations for topic inference. The results reported are the average over 5 runs.

4.2 Results and Analysis

We sample 100 texts for each category with tags information randomly from training dataset and set k in Eq. 1 to 10 to evaluate the quality of topic sets by Algorithm 1. As the number of optimal topic sets is fixed to 3, we get the optimal topic sets $\mathrm{O} = \{T10, T30, T50\}$ for both two datasets coincidentally, and the weight vectors $\hat{\mu} = \{3.44, 1.7, 1\}$ for *Search Snippets* and $\hat{\mu} = \{1.31, 1.22, 1\}$ for *20Newsgroups*. It is noteworthy that the weight values of the topic sets are affected by both the type of dataset and the settings of LDA. Below, a series of experiments are conducted to answer the questions: (1). How does the proposed approach HMTT compare with other baseline methods; (2). Whether the optimal multi-granularity topics can outperform single-granularity topics and other multi-granularity topics; (3). Which approach of the two strategies to integrate multi-granularity topics can achieve a better performance.

Compared with the Existing Hashing Methods: In this section, we design an improved version of STHs, denoted as STHs-Tag, by replacing the original

[4] `http://jgibblda.sourceforge.net/`

Table 1. Mean precision (mP) of the top 200 examples and the retrieved examples within Hamming radius 3 on *SearchSnippets* with 8 and 16 hashing bits. e.g. 10-30-50* means that the proposed methods incorporate the optimal multi-granularity topics, and 10-30-50W1 means that hashing method uses the multi-granularity topic sets {$T10$, $T30$, $T50$} while fixing the balance values to 1:1:1.

—	mP@Top 200				mP@Hamming Radius 3			
Methods	HMTT-Fea		HMTT-Dec		HMTT-Fea		HMTT-Dec	
Code Length	8 bits	16 bits	8 bits	16 bits	8 bits	16 bits	8 bits	16 bits
10-30-50*	**0.829**	0.799	**0.826**	**0.782**	**0.411**	**0.802**	**0.403**	**0.778**
10-70-90	0.819	**0.800**	0.797	0.762	0.375	0.789	0.328	0.754
30-90-150	0.802	0.787	0.801	0.755	0.393	0.777	0.382	0.757
10-30	0.810	0.789	0.776	0.757	0.382	0.776	0.374	0.744
10-50	0.813	0.788	0.772	0.752	0.383	0.790	0.334	0.740
30-50	0.806	0.796	0.805	0.777	0.393	0.779	0.369	0.764
10-30-50W1	0.811	0.780	0.822	0.778	0.368	0.761	0.398	0.774
10	0.627	0.624	0.639	0.602	0.316	0.610	0.296	0.576
30	0.792	0.764	0.728	0.708	0.377	0.757	0.335	0.692
50	0.782	0.758	0.731	0.723	0.360	0.730	0.320	0.707
70	0.771	0.755	0.728	0.720	0.365	0.747	0.318	0.704
90	0.757	0.733	0.735	0.708	0.363	0.736	0.332	0.692
120	0.730	0.705	0.707	0.700	0.366	0.714	0.309	0.683
150	0.740	0.727	0.675	0.674	0.370	0.729	0.304	0.660

construction of similarity matrix with the proposed method described in Section 3.2. We remove 60 percent tags randomly from the training dataset to verify the robustness for HMTT-Fea, HMTT-Dec, STHs and STHs-Tag. The precision-recall curves for retrieved examples are reported in Fig. 2. From these comparison results, we can see that HMTT-Fea and HMTT-Dec significantly outperform other baseline methods on *Search Snippets* as shown in Fig. 2 (a). For *20Newsgroups*, HMTT-Dec performs close results with STHs-Tag in Fig. 2 (b). The reasons to explain this problem are that: Firstly, *20Newsgroups* as a normal dataset has sufficient original features to learn hash codes so that STHs-Tag based on keyword features works well. Secondly, we directly learn the topic models of *20Newsgroups* from the training dataset that result in some restrictions. Furthermore, STHs get a worse performance than STHs-Tag on two datasets. Because STHs uses a complete supervised approach which only utilizes the pairwise similarity of the documents with common tags, that method cannot well deal with the situations that tags are missing or incomplete. In our approach, we extract the optimal multi-granularity topics depending on the type of dataset to learn hash codes and hashing function, and the tags are just utilized to adjust the similarity, which has stronger robustness. In the following experiment sets, we keep the all tags to improve the performance of hashing learning.

Compared with Single-Granularity and Other Multi-granularity Topic Sets: Here, the hashing performances of the optimal multi-granularity topics are compared with single-granularity and other multi-granularity topics. We further

evaluate the balance values of the multi-granularity topics by fixing them to 1. In particular, we keep the parameters $\hat{\mu}_i$ in Eq. 3 and α_i in Eq. 7 to 1 for HMTT-Fea and HMTT-Dec respectively. The quantitative results on *Search Snippets* are reported in Table 1. From the results, we can see that the performances of multi-granularity topics significantly outperform single-granularity topics and the optimal multi-granularity topics achieve a better performance in most situations. We also observe similar results on *20Newsgroups*. But due to the limit of space, we select to present the results on the typical short texts dataset *Search Snippets*.

Compared between the Proposed Two Strategies: Finally, we mainly discuss the performances between the proposed two strategies, HMTT-Fea and HMTT-Dec. In HMTT-Fea, we directly concatenate the multi-granularity topics to produce one feature vector and decompose the hashing learning problem into two separate stages. In HMTT-Dec, the multi-granularity topics extracted from the text content are treated as multi-view features, and we simultaneously learn the hash codes as well as hash function. From the results in Table 1, we can see that the performances of HMTT-Fea surpass HMTT-Dec on several evaluation metrics. Obviously, the former strategy is more simple and effective for short text hashing in our approach. In summary, no matter in HMTT-Fea or HMTT-Dec, the experimental results indicate that short text hashing can be improved by integrating multi-granularity topics.

5 Discussions and Conclusions

Short text hashing is a challenging problem due to the sparseness of text representation. In order to address this challenge, tags and latent topics should be fully and properly utilized to improve hashing learning. Furthermore, it is better to estimate the topic models from an external large-scale corpus and the optimal topics should be selected depending on the type of dataset. This paper uses a simple and effective selection methods based on symmetric KL-divergence of topic distributions, we think that there are many other selection methods worthy of being explored further. Another key issue worthy of research is how to integrate the multi-granularity topics effectively. In this paper, we propose a novel unified hashing approach for short text retrieval. In particular, the optimal multi-granularity topics are chosen depending on the type of dataset. We then use the optimal multi-granularity topics to learn hash codes and hashing function on two distinct ways, meanwhile, tags are utilized to enhance the semantic similarity of related texts. Extensive experiments demonstrate that the proposed method can perform better than the competitive methods on two public datasets.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No. 61203281 and No. 61303172.

References

1. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2006, pp. 459–468. IEEE (2006)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
4. Chen, M., Jin, X., Shen, D.: Short text classification improved by learning multi-granularity topics. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 1776–1781. AAAI Press (2011)
5. Cheng, X., Lan, Y., Guo, J., Yan, X.: Btm: Topic modeling over short texts. IEEE Transactions on Knowledge and Data Engineering, 1 (2014)
6. Jin, O., Liu, N.N., Zhao, K., Yu, Y., Yang, Q.: Transferring topical knowledge from auxiliary long texts for short text clustering. In: CIKM, pp. 775–784. ACM (2011)
7. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
8. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the Twelfth International Conference on Machine Learning, Citeseer (1995)
9. Lin, G., Shen, C., Suter, D., van den Hengel, A.: A general two-step approach to learning-based hashing. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2552–2559. IEEE (2013)
10. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, pp. 91–100. ACM (2008)
11. Salakhutdinov, R., Hinton, G.: Semantic hashing. International Journal of Approximate Reasoning 50(7), 969–978 (2009)
12. Wang, Q., Zhang, D., Si, L.: Semantic hashing using tags and topic modeling. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 213–222. ACM (2013)
13. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems, pp. 1753–1760 (2009)
14. Xu, J., Liu, P., Wu, G., Sun, Z., Xu, B., Hao, H.: A fast matching method based on semantic similarity for short texts. In: Zhou, G., Li, J., Zhao, D., Feng, Y. (eds.) NLPCC 2013. CCIS, vol. 400, pp. 299–309. Springer, Heidelberg (2013)
15. Zhang, D., Wang, F., Si, L.: Composite hashing with multiple information sources. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 225–234. ACM (2011)
16. Zhang, D., Wang, J., Cai, D., Lu, J.: Extensions to self-taught hashing: Kernelisation and supervision. Practice 29, 38 (2010)
17. Zhang, D., Wang, J., Cai, D., Lu, J.: Laplacian co-hashing of terms and documents. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 577–580. Springer, Heidelberg (2010)
18. Zhang, D., Wang, J., Cai, D., Lu, J.: Self-taught hashing for fast similarity search. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 18–25. ACM (2010)

A Computational Approach for Corpus Based Analysis of Reduplicated Words in Bengali

Apurbalal Senapati[1] and Utpal Garain[2]

[1] Central Institute of Technology, BTAD, Kokrajhar-783370, Assam, India
apurbalal.senapati@gmail.com
[2] Indian Statistical Institute, 203, B.T.Road, Kolkata – 700108, India
utpal.garain@gmail.com

Abstract. Reduplication is an important phenomenon in language studies especially in Indian languages. The definition of reduplication is the repetition of the smallest linguistic unit partially or completely i.e. repetition of phoneme, morpheme, word, phrase, clause or the utterance as a whole and it gives different meaning in syntax as well as semantic level. The reduplicated words has important role in many natural language processing (NLP) applications, namely in machine translation (MT), text summarization, identification of multiword expressions, etc. This article focuses on an algorithm for identifying the reduplicated words from a text corpus and computing statistics (descriptive statistics) of reduplicated words frequently used in Bengali.

Keywords: Reduplication, Bengali, Corpus, Descriptive statistics, Evaluation.

1 Introduction

Reduplication is one of the highly productive morphological processes in Bengali. It is frequently used in the language for various linguistic and pragmatic reasons and purposes. The use of reduplicated words in text or corpus is in different ways and manners to serve various means of information-sharing and communication. Although it is mostly used to express a sense of multiplicity of various countable items, it is also used as a process to refer to the act of continuation of an action or an event [1] or something else.

For example, S1: আপনি কোন গ্রামে যেতে চান ? / *aapni kon grame jete chan* ? (Which village do you want to visit?); S2: আপনি কোন কোন গ্রামে যেতে চান ? / *aapni kon kon grame jete chan* ? (Which are the villages you want to visit?). Clearly in sentence S2, the semantic changes to plural and it is due to the use of reduplication of word কোন /*kon* (which). Similarly for example S3: ঘরে কোন লোক নাই /*ghare kono lok nai* (There is no one in the house); S4: ঘরে ঘরে বেকার যুবক / *ghare ghare bekar jubak* (unemployment is in every house). The semantic meaning of reduplication of ঘরে/*ghare* (in house) in S3 and S4 are different. In S3, meaning of ঘরে /*ghare* is "in house" but in S4, the meaning of ঘরে ঘরে /*ghare ghare* is "in every house". Now it is clear that in many NLP applications especially in MT, the semantic of reduplication has to be considered carefully in order

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 456–466, 2015.
DOI: 10.1007/978-3-319-18111-0_34

to achieve high accuracy. For example, the machine translation (using Bengali to English Google translator, dated 28th January, 2015) of sentences S5: সে খেতে আসছে / *se khete aaschhe* and S6: সে খেতে খেতে আসছে / *se khete khete aaschhe* is "*He is coming to eat*" in both cases. But the actual translations are "*He is coming to eat*" and "*He is coming while eating*", respectively. It is obvious that the wrong translation producing in sentence S6 due to failure in capturing the semantic of reduplication. Similarly in many NLP applications the reduplication has to be tackled separately in order to reduce semantic analysis error.

2 Types of Reduplicated Words in Bengali

The process of reduplication is quite frequent in Bengali. A large number of words are capable of producing valid reduplication. But practically most of them are not used or used with very low frequencies. It is also observed that the reduplication can occur to all word categories including the pronouns and indeclinable.

From the structural point of view there are six types of reduplication in the Bengali texts [1], which are as follows:

i) The repetition of same word as a second member without the addition of any suffix of inflectional properties with any member i.e. the proper reduplication. Examples, হাসি হাসি / *hasi hasi* (smiling) [হাসি/ *hasi* (smile)]; বছর বছর / *bachhar bachhar* (every year) [বছর / *bachhar* (year)]; লাল লাল / *lal lal* (red in plural sense) [লাল / *lal* (red in singular sense)]; ভালো ভালো / *bhalo bhalo* (good in plural sense) [ভালো / *bhalo* (good in singular sense)]; দিন দিন / *din din* (day by day) [দিন / *din* (day)] etc. Note that in this category, each individual word has a valid mining which is different (same on some cases) their reduplicated meaning.

ii) The first word is repeated and while first word carries no inflection but the second word carries an inflection. Examples, ধব ধবে /*dhab dhabe* (pure white color) [ধব/*dhab* and ধবে /*dhaeb* are not valid words], টক টকে/*tak take* (deep red color) [টক/*tak* (sour) but টকে/*take* (not a valid word)], লক লকে/*lak lake* (flickering / attractive) [লক/*lak* and লকে/*lake* are not a valid words] etc. Note that in this category, each individual word is not a valid word whereas the reduplicated words are meaningful.

iii) The first word is inflected and then inflected word is repeated. Example, ঘরে ঘরে / *ghare ghare* (in every house) [ঘরে / *ghare* (in house)], কানে কানে / *kane kane* (secretly) [কানে / *kane* (in ear)], গাছে গাছে / *gachhe gachhe* (in every tree) [গাছে / *gachhe* (in tree)] etc. Note that, this category is also proper reduplication and in this case also the semantic behavior is same as of category i).

iv) A semantically or almost similar word is added with the first word to generate the reduplicated word. Example, চাল চুলো / *chal chulo* (economically poor) [চাল/chal (rice), চুলো / *chulo* (cooking burner)], চুরি চামারি / *churi chamari* (robbery) [চুরি / *churi* (theft), চামারি / *chamari* (illegal work)], অলি গলি / *ali gali* (narrow lane with complicated direction) [অলি / *ali* (narrow lane), গলি / *gali* (narrow lane)] etc. Note that, in this case the semantic meaning of each individual word is almost same and their re-duplicated meaning is also almost similar to the individual word.

v) An eco word is added as the second member with the first word to generate the reduplicated word. Example, জল টল/ *jal tal* (water, beverage etc.) [জল / *jal* (water) and টল/ *tal* (eco word)], খাবার দাবার/ *khabar dabar* (varieties food) [খাবার/ *khabar* (food) and দাবার/ *dabar* (eco word)], মাছ টাছ/ *mach tachh* (egg, fish, meat etc.) [মাছ/ *mach* (fish) and টাছ/*tachh* (eco word)] etc. Note that, in this case the first word has a specific meaning but after adding the eco word the meaning changes. Also note that the composite meaning is almost similar to the first word but in plural form but this property does not follow in all cases.

vi) Onomatopoeic words made with two words of identical structures. Examples, ছম ছম / *chham chham* (feeling of sound of silence), খিল খিল /*khil khil* (sound of laugh), ঝিন ঝিন /*jhin jhin* (jingling) etc. Note that, this category is also proper reduplication and in this case the semantic meaning is related to sound (real or virtual) of different events.

Whereas, the reduplicated words can be classified in other perspective like phonological perspective, morphological perspective, lexical perspective, constructional perspective, etc. In functional point of view, reduplicated can be classified based on the part of speech also. But our present work only concentrate on the computational aspect of identifying the reduplicated words from the corpus.

3 Existing Work on Reduplicated Word in Bengali

Most of the existing works on reduplication is contributed by the linguistic people and it has started long back in many Indian languages. Ananthanarayana [2] describes the reduplication in Sanskrit and Tamil, Abbi [3] focuses on the different aspect of reduplication on south Asian language, Murthy [4] worked on Kannada language, etc. The work on Manipuri reduplicated is found in identification of multiword expressions by in Nongmeikapam [5] work. In Bengali language, the linguistic study found from Chattopadhyay [6] , Chaudhuri [7], Thompson's [8] work. In computational point of view, Bandyopadhyay [9] has studied reduplicated words for semantic based analysis. Senapati [10] has studied the reduplicated pronoun in their anaphora resolution task in Bengali.

4 Our Contribution

From the literature survey it is clear that most of analysis is on the linguistic point of view and the works are common in nature i.e. analysis of reduplicated words and tried to capture their semantic meaning. Whereas the computational works are limited. But some basic issues like how many reduplicated words are there in Bengali or what are the frequencies in which reduplicated words appear in Bengali, etc. i.e. the corpus based statistics are still not studied. We have proposed an algorithm to identify the reduplicated words from a text corpus and also proposed a dictionary based tuning technique to enhance the accuracy of identifying such word in the corpus. Finally, the frequencies of reduplicated word have been calculated in word level as well as in sentence level.

5 Computational Approach to Identify Reduplication in Bengali

Our computational approach is based on the morphological similarities of the duplicated words. In our work, the morphologically similar reduplicated words implies that the similar or almost similar words in terms of their word length and use of characters or use of vowel modifiers in the words. In section 2, we have seen that in category (i), (iii) and (vi) the formation of reduplication by the repetition of same word i.e. of the form "*w w*" where "*w*" is word in the corpus. Also we observe that, in category (ii), (iv) and (v) the formation of reduplication by the repetition of almost similar word. And hence from the computational aspect we define the reduplicated words of two types. The proper reduplication i.e. when the repetition of same word; for example, খেতে খেতে / *khete khete* (continue eating), যেতে যেতে / *jete jete* (continue going), where above category (i), (iii) and (iv) come under this type. The other type is the partial reduplication i.e. first and second word is not exactly same but almost similar; for example, খাবার দাবার/ *khabar dabar*/ (food etc.), চাল চুলো / *chal chulo* (economically poor), where above category (ii), (iv) and (v) come under this type. There are some exceptional cases, e.g. মাথা মুন্ডু/*matha mundu* (meaning lass), লোটা কম্বল/*lota kambal* (belonging of poor man), etc. and we are now not considering these cases. The computational approaches for identifying two different types of reduplication are also different and handled by two different algorithms. Finally, to reduce the error we have used a dictionary and frequency based tuning technique. The details descriptions of the algorithms are given below.

Table 1. Algorithm to find the proper reduplication from the text corpus

	ALGORITHM	
s1:	$w_i \leftarrow$ word from corpus	
s2:	if w_i contains "-" then	
s3:	if w_i is of the form "*w*-*w*" then	// type 2
s4:	print "**re-duplication**";	
s5:	frequency= frequency+1;	
s6:	end if	
s7:	else if w_i is of the form "*ww*" then	// type 3
s8:	print "**re-duplication**";	
s9:	frequency= frequency+1;	
s10:	end if	
s11:	else	
s12:	$w_{i+1} \leftarrow$ next word from corpus	
s13:	if "w_i *is equal to* w_{i+1}" then	// type 1
s14:	print "**re-duplication**";	
s15:	frequency= frequency+1;	
s16:	end if	
s17:	end if	

For algorithmic approach, first we analyzed the proper reduplication in terms of morphological similarity. In lexical point of view the proper reduplication is of three

types. The first type is of the form "*w w*" i.e. repetition of same word with a single space; for example, খেতে খেতে / *khete khete* (continue eating). The second type is of the form "*w-w*" (or "*w - w*") i.e. repetition of same word with a "-" separation, for example, ধীরে - ধীরে/*dhire dhire* (slowly) and the third type is of the form "*ww*" i.e. repetition of same word without any space; for example, গজগজ/*gajgaj* (feeling of irritation). The formal algorithm of this category is given in Table 1. Also note that the algorithm also calculating the frequencies of reduplicated words separately.

To identify the partial reduplicated word is relatively complicated compared to proper reduplication and hence first we studied the features of partial reduplication to setup our algorithm. In earlier work some people have been used some heuristic rules. According to Bandyopadhyay [9] the partial reduplication are of three types, (i) change of the first vowel or the matra (vowel modifier) attached with first consonant, (ii) change of consonant itself in first position or (iii) change of both matra and consonant. They have also identified some exceptions e.g. আবল-তাবল/ *aabol-taabol* (irrelevant) etc. According to the linguistic study of Chattopadhyay [6], we found the rule formation of partial reduplication i.e. the consonants that can be produced after changing are ট, ফ, ম, স. Now from the above studied and from our observation on reduplicated words, the common features of partial reduplication are:

(i) Most of the cases the length of the individual words are same e.g. কখনো সখনো/*kakhano sakhano* (sometimes) e.g. *length(*কখনো*) = length(*সখনো*)* or length of reduplicated word is one more than the first word e.g. ধব ধবে */dhab dhabe* (pure white color) where *length(*ধব*)+1 = length(*ধ, ে*)*

Table 2. Algorithm to find partial reduplication from corpus

ALGORITHM	
s1:	w_i and w_{i+1} ← word from corpus
s2:	if (*length*(w_i) == *length*(w_i)) then
s3	count ← *charecterWiseDifferent*(w_i, w_{i+1});
s4	differentCharecterPair ← (c_1, c_2); //mismatch character pair in w_i & w_{i+1}
s5	if (*count ==1*&& (c_1 & c_2 *both vowel modifier or both alphabet*) then
s6:	print "**re-duplication**";
s7	frequency= frequency+1;
s8:	end if
s9	else if (*length*(w_i)+*1* == *length*(w_{i+1})) then
s10:	count ← *charecterWiseDifferent*(w_i, w_{i+1});
s11:	if (*count ==1*) then
s12:	misMatchChar ←(w_i , w_{i+1}); // mismatch character
s13:	if (*misMatchChar is vowel modifier*) then
s14:	print "**re-duplication**";
s15:	frequency= frequency+1;
s16:	end if
s17:	end if
s18:	end if

(ii) The difference between the reduplicated words in character wise is either a letter [e.g. কখনো সখনো/*kakhano sakhano* (sometimes) where difference character pair is (ক, স)] or a vowel modifier [e.g. খুচ খাচ/*khuch khach* (little bit) where difference character pair is (ু,া)]

(iii) Numbers of characters differs in one and

(iv) Most of the cases this letter is a consonant of specific types like ট, ফ, ম, স, etc.

Now based on these observations we have incorporated the features i.e. (i), (ii) and (iii) in our algorithm to identifying the partial reduplication and is given in Table 2. In this algorithm, the function *charecterWiseDifferent*(w_i, w_{i+1}) returns the number of mismatch between two words w_i and w_{i+1} character wise and also calculating the frequencies of each reduplicated word. Note that, this algorithm also considering cases like w_i-w_{i+1} (or w_i - w_{i+1}) but not shown in algorithm separately.

6 Corpus Based Study of Reduplicated Words in Bengali

For the corpus based study of reduplication in Bengali, the Technology Development for Indian Languages (TDIL) corpus [12] has been used. The TDIL corpus is developed by the Department of Electronics, Govt. of India for Bengali language (http://tdil.mit.gov.in/). This corpus contains texts from Literature (20%), Fine Arts (5%), Social Sciences (15%), Natural Sciences (15%), Commerce (10%), Mass media (30%), and Translation (05%). Where each category has some sub categories e.g. Literature includes novels, short stories, essays etc.; Fine Arts includes paintings, drawings, music, sculpture etc.; Social Science includes philosophy, history, education etc.; Natural Science includes physics, chemistry, mathematics, geography etc.; Mass Media includes newspapers, magazines, posters, notices, advertisements etc. Commerce includes accountancy, banking etc., and translation includes all the subjects translated into Bengali. The size and the number of reduplicated words found using above algorithms in the corpus are given in the following table (Table 3).

Table 3. Reduplicated words in TDIL corpus

Corpus	# Files	# Sentences	# Words	# Reduplicated words (unique)	# Frequency
TDIL	1362	334260	4429574	6196	61647

The Table 3 shows that the percentage of reduplicated words in the corpus is 1.4% and at a glance it looks like quite low but while it will be consider in sentence level then it shows that, 18.44% of the sentences contain reduplicated words. Since the semantic of sentence highly depends on the presence of reduplicated word and hence this percentage shows that it cannot just ignore in any NLP application.

7 Tuning Technique

Though our algorithm has potential to identifying the reduplicated words compared to other existing approaches but still in order to reduce the error we have used a dictionary and frequency based tuning technique. Table 3 shows that there are a large number of reduplicated words with high frequency in the corpus. But our observation is that many of them are erroneous or not reduplicated word at all. And some of them occur with very low frequency and can be ignored without loss of generality. For example, the algorithm produces output "*2424*" or "ऽऽऽऽ" or "((" as reduplicated words, since these are strings of the form "*ww*", but actually not the reduplicated words. Hence in order to improve the efficiency we have used the tuning technique. Also we have used a technique to identify the reduplicated with eco words. This identification is very helpful in many NLP applications especially in MT.

Frequency measure: The frequency measure is an important technique to validate the word or association of words in a corpus. The general phenomenon is that the high frequencies of two words occur together, then that is evidence that they have a special function that is not simply explained as the function that results from their combination. Based on this phenomenon have we fixed a threshold frequency (T_f) and hence if the frequency of reduplicated words exceeds the threshold frequency i.e. > T_f then we only consider them in our experiment. Whereas to fix the threshold value many factors has to be considered like, the size of the corpus, domain of the corpus etc. Note that in our experiment we have defined T_f = 5 by applying the random sample technique in the corpus. Using this technique many irrelevant entries has been eliminated. For example, পি. ভি./p.v. (abbreviation of a name), বৃদ্ধ বৌদ্ধ /*bridha boudha* (irrelevant word) etc. structurally look like reduplicated but actually not.

Online dictionary: In this case we have eliminated the incorrect words using an online dictionary in the following techniques. We validate "*ww*" in online dictionary [13] and if "*ww*" found a valid word in the dictionary then we reject "*ww*" i.e. do not consider it as a reduplicated word. For example, the algorithm will produce output "বাবা", "দাদা", "দিদি", "মামা" etc. as reduplication word, since these are strings of the form "*ww*". Now, once these words are checked in online dictionary and found as valid words, they are rejected as reduplicated words. Following this method, all (erroneous words like, বাবা/*baba* (father), দাদা/*dada* (elder brother), দিদি/*didi* (elder sister), মামা/*mama* (maternal uncle) etc. are eliminated.

Identification of eco words: In case of w_1-w_2 form, system splits it into w_1 and w_2 separately and validate in online dictionary separately. If it shows that the first one i.e. w_1 is a valid word but second one i.e. w_2 is not a valid word then identified is as eco words. For example, অঙ্ক-টঙ্ক/anka-tanka (maths etc.) [অঙ্ক/anka (maths), টঙ্ক/tanka (eco word)], আত্মীয়-টাত্মীয়/aantiya-taantiya (relatives) [আত্মীয়/aantiya (relative), টাত্মীয়/taantiya (eco word)], ব্যাপার-স্যাপার/bapar-sapar (matters) [ব্যাপার/bapar (matter), স্যাপার/sapar (eco word)], etc.

The interesting observation is that, after applying the tuning techniques the number of reduplicated words is reduced significantly and most of the erroneous entities are eliminated, and the revised result is shown in Table 4.

Table 4. Reduplicated words in TDIL corpus (after tuning)

Corpus	# Files	# Sentences	# Words	# Reduplicated words (unique)	# Frequency
TDIL	1362	334260	4429574	794	37919

After tuning, Table 4 shows that the percentage of reduplicated words in the corpus is 0.71% and in sentence level then it shows that 9.4% of the sentences contain reduplicated words. Clearly after tuning process system eliminates about 50% of reduplication produced by the above algorithm. Next section shows that improvement of accuracy after tuning technique.

8 Evaluation

The system has been evaluated by the stratified simple random technique on the TDIL corpus. The technique is due to Sharon [11]. The technique in brief is as follows. The corpus is partitioned into non-overlapping groups and then groups are selected in random. Now from a selected group the manual output and the system output have been considered for the final evaluation. The Precision, Recall and F-score have been used as evaluation metric and result shows in Table 5. Note that, though the system is identifying the eco words separately, we are not evaluating the performance of eco word identification separately.

Table 5. Result for identification Reduplicated words in TDIL corpus

	Corpus	Precision	Recall	F-score
Before Tuning	TDIL	0.63	0.85	0.72
After Tuning	TDIL	0.93	0.84	0.88

9 Error Analysis

In order to find the weakness of our algorithms the error analysis has been carried out. This analysis not only measures the error in terms of number of wrongly identified but also identified the major source of errors in different phases of the system. Broadly we have identified the source of errors in two phases; the error generated by the system output and the error generated in tuning phase.

The Table 6 and Table 7 are the confusion matrixes for identification of reduplicated words before and after applying the tuning technique respectively. Table 6 shows that there were 45685 reduplicated words in the corpus and the system capable to capture only 38719 instances correctly and identified 22928 instances wrongly. Note that, “Actual False (X)” shown in Table 6 in first row indicates that number of non reduplicated words present in the corpus. Since, this number is not relevant in our measure and hence it is omitted and similarly the value of “true negative (X)” is also not calculated.

Table 6. Confusion matrix before applying tuning technique used in TDIL corpus

	Actual True (45685)	Actual False (X)
System Identified true	true positive (38719)	true negative (X)
System Identified false	false negative (6966)	false positive (22928)

Table 7 shows the result after applying tuning process. Note that in this table the number of actual reduplicated words is 42230 i.e. it reduces 3455 (45685 - 42230) true instances for tuning technique. Based on our observation, major contribution of this elimination is due to the instances with low frequency i.e. below threshold level. Also note that, applying tuning technique system has eliminated 20273 (true negative) false instances. In this case, the major contribution of this elimination is due to the use of dictionary entries. Note that some very common word (false instances like, বাবা/*baba* (father) with frequency 1342, দাদা/*dada* (elder brother) with frequency 327, দিদি/*didi* (elder sister), with frequency 325 etc.) i.e. instances with very high frequencies are eliminated and results improve the system performance. The error analysis in algorithmic level is given below.

Table 7. Confusion matrix after applying tuning technique used in TDIL corpus

	Actual True (42230)	Actual False (22928)
System Identified true	true positive (35474)	true negative (20273)
System Identified false	false negative (6756)	false positive (2655)

The error produced by the algorithms can be categorized of two types. We consider the first type is the *false negative* i.e. the algorithm fails to identify the reduplicated words. Actually the algorithm is designed based on the analysis of lexical features (details is in section 5) of reduplicated words. But these features does not cover all types of reduplicated words, especially those reduplicated words, where first and second words having morphological variant. For example, the reduplicated words like মাথা মুন্ডু/*matha mundu* (meaning lass), লোটা কম্বল/*lota kambal* (belonging of poor man), etc. are not covered by the algorithms. And hence it affects the accuracy in terms of recall and it reflects the recall value shown in Table 5. The other type of error is the *false positive* i.e. the algorithm wrongly identify the reduplicated words. For examples, consider the system generated output with their frequencies, দমদম/*dumdum* (Dumdum, name of a place) [frequency 50], টাটা/*tata* (Tata, name of a place) [frequency 50], শ্রীশ্রী/srisri (Mr. like term come before a name of male person) [frequency 17] etc. Clearly, দমদম/*dumdum* (Dumdum) will not be eliminated in tuning mechanism, because দমদম/*dumdum* (Dumdum) is not a valid word in online dictionary and since its frequency greater than threshold frequency ($50 > T_f = 5$). Hence it contributes errors and it affects the accuracy in terms of precision and it reflects the precision value shown in Table 5.

The error produced by the tuning technique also can be categorized of two types. The first type is the *false negative* i.e. the tuning technique elements the true reduplicated words. This will happened because, if some reduplicated words present in the corpus with low frequency ($\leq T_f$). For example, consider the system generated output with their frequencies মড়মড়/*marmar* (sound of break) [frequency 4], ঝিরিঝিরি/*jhirjhir*

(sound of rain in slow motion) [frequency 4], গুটিগুটি/*gutiguti* (slowly) [frequency 4], etc. Though all these are valid reduplicated words but will be eliminated by tuning mechanism because of the low frequency ($\leq T_f$). Obviously it affects the accuracy in terms of recall and it reflects the recall value shown in Table 5. The other error type is the *false positive* i.e. the tuning technique fail to eliminate the false reduplicated words. The examples describes above, like দমদম/*dumdum* (Dumdum) [frequency 50], টাটা/*tata* (Tata) [frequency 50], etc. will not eliminated by the tuning technique.

10 Conclusion

This paper presents a pioneering attempt to develop a computational approach for corpus based study of reduplicated word in Bengali. The paper also shows the frequencies of reduplicated words and also shows that how frequently the reduplicated words are present in a corpus as well as at the sentence level. It also identified with examples that the affect of reduplication in MT system and has focused an untouched issue in Bengali-English MT. The algorithms used for identifying the reduplicated words are very simple. Though, the performances of the algorithms are not very high but after applying the tuning techniques the performance has improved to the satisfactory level. The error analysis part also identified the weaknesses of the system and hence there is future scope to improve the accuracy further.

Acknowledgement. The authors sincerely acknowledge Prof. B.B. Chaudhuri of Indian Statistical Institute, who kindly shared his expertise on Bengali reduplicated words with the authors.

References

1. Dash, N.: A Descriptive Study of Bengali Words, pp. 225–251. CUP (2015)
2. Ananthanarayana, H.S.: Reduplication in Sanketi Tamil OpiL, vol. 2, pp. 39–49 (1976)
3. Abbi, A.: Reduplicated Adverbs of Manner and Cause of Hindi. Indian Linguistics 38(2), 125–135 (1977)
4. Murthy, C.: Formation of Echo-Words in Kannada. In: All India Conference of Dravidian Linguistics(eds.) (1972)
5. Nongmeikapam, K.: Identification of Reduplication MWEs in Manipuri, a rule-based approach. In: 23rd International Conference on the Computer Processing of Oriental Languages, California, USA, pp. 49–54 (2010)
6. Chattopadhyay, S.K.: Bhasa-Prakash Bangala Vyakaran, 3rd edn. Pupa publication (1992)
7. Chaudhuri, B.B.: Bangla Dhwanipratik: Swarup o Abhidhan (Bangla Sound Symbolism: Properties and Dictionary). Paschimbanga Bangla Academy, Kolkata (2010)
8. Thompson, H.R.: Bengali: A Comprehensive Grammar, pp. 663–672. Routledge publication (2010)
9. Bandyopadhyay, S.: Identification of Reduplication in Bengali Corpus and their Semantic Analysis: A Rule-Based Approach. In: Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010), Beijing, pp. 72–75 (2010)

10. Senapati, A., Garain, U.: Anaphora Resolution in Bangla using global discourse knowledge. In: Int. Conf. of Asian Language Processing, Hanoi, Vietnam (2012)
11. Sharon, L.L.: Sampling: Design and Analysis, 2nd edn. Advanced Series, pp. 73–101 (2010)
12. TDIL Corpus: A nation-wide consortium for machine translation of Indic languages is being funded by the Ministry of Information Technology, Govt. of India (1995), http://www.tdil-dc.in
13. Digital Dictionaries of South Asia, http://dsal.uchicago.edu/dictionaries/biswas-bangala/

Automatic Dialogue Act Annotation within Arabic Debates

Samira Ben Dbabis[1], Hatem Ghorbel[2], Lamia Hadrich Belguith[1], and Mohamed Kallel[3]

[1] ANLP Research Group, MIRACL Laboratory, University of Sfax, Tunisia
{samira.benedbabis,l.belguith}@fsegs.rnu.tn
[2] University of Applied Science of West Switzerland HE-Arc Ingénierie, Switzerland
hatem.ghorbel@he-arc.ch
[3] Faculty of Letters and Human Sciences, University of Sfax, Tunisia
med.kalel@gmail.com

Abstract. Dialogue acts play an important role in the identification of argumentative discourse structure in human conversations. In this paper, we propose an automatic dialogue acts annotation method based on supervised learning techniques for Arabic debates programs. The choice of this kind of corpora is justified by its large content of argumentative information. To experiment annotation results, we used a specific annotation scheme relatively reliable for our task with a kappa agreement of 84%. The annotation process was yield using Weka platform algorithms experimenting Naive Bayes, SVM and Decision Trees classifiers. We obtained encouraging results with an average accuracy of 53%.

Keywords: Dialogue act annotation, argumentative scheme, Arabic debates, supervised learning classifiers.

1 Introduction

Dialogue Acts (DA) recognition is a hot topic of research in Discourse theory more precisely in conversational analysis. Influential works appeared in this context, a proliferation of annotation schemes has been developed through annotation projects like Maptask [1], Verbmobil [2, 3, 4], DAMSL [5] and DIT++ schema [6, 7], often started from the topology suggested by Searle [8]. The granularity of DA annotation labels varies considerably from domain-specific to open-domain annotation task.

Human annotation and possible labelling paradigms were then exploited in a variety of empirical methods to perform practical DA classification [9, 10].

The annotation task is fundamental to many studies in human discussions analysis as they reflect shallow discourse structures of language that can be investigated to build an argumentative structure of discussions.

The main purpose of automatic DA annotation in our work is to extract adjacency pairs (question/answer, opinion/reject, confirmation request/confirmation, etc.).

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 467–478, 2015.
DOI: 10.1007/978-3-319-18111-0_35

These pairs are then investigated to generate sequences of acts in order to build an argumentative discourse structure that can help user to answer complex queries. For example the argumentative chain "Thesis/Opinion_request/Opinion/Reject" is applied to answer to the complex question "who accepted the opinion of X related to the thesis Y?"

Tracking argumentative information is mainly based on exchanging opinions, raising issues, making suggestions, providing arguments, negotiating alternatives, and making decisions.

To facilitate extracting argumentative data, it is useful to automatically annotate participant interaction characteristics specifically by identifying agreement and disagreement in order to understand social dynamics. Annotating debate programs acts can be also a motivating task when a user needs information about a past discussion that he missed, or wants to recollect discussion dynamics (topic discussed, agreements, disagreements, arguments, etc).

In this perspective, we propose an automatic annotation of dialogue acts referring to the proposed labelling scheme specific for argumentative debates programs. We experiment supervised learning machine algorithms as SVM, Naïve Bayes and Decision Trees techniques from the Weka Toolkit.

This paper is structured in four sections. First, we present previous works in automatic annotation task. Then, we focus on the role of DA in building conversation structures. In the third section, we detail the proposed DA annotation scheme. Finally, we experiment learning algorithms using a set of manual annotated Arabic discussions collected from Aljazeera TV programs.

2 Related Work

Research has continued to experiment with machine learning techniques used to automatically label DAs. They are usually based on supervised learning modeling approaches including sequential approaches and vector-based models.

Sequential approaches typically formulate dialogue as a Markov chain in which an observation depends on a finite number of preceding observations. HMM-based approaches make use of the Markov assumption in a doubly stochastic framework that allows fitting optimal dialogue act sequences using the Viterbi algorithm [9], [11]. Research using sequential approaches usually involves combinations of Ngrams and Hidden Markov Models.

Vector-based approaches such as maximum entropy modeling [12, 13], also frequently take into account lexical, syntactic and structural features. Lexical and syntactic cues are extracted from local utterance context, while structural features involve longer dialogue act sequences in task-oriented domains.

More interestingly from a linguistic point of view, researchers were focused on features enhanced dialogue context [14, 15, 16]. [17] explore the predictive power of dialogue context on Dialogue Act classification. They extend Latent Semantic Analysis (LSA), which only uses words, with linguistic features as Feature LSA (FLSA). FLSA improves DA classification performance via the combination with k-Nearest Neighbor algorithm (k-NN).

Most DA annotation classifiers were experimented using several dialogue corpora (Switchboard [5]; Trains [18] and Spanish CallHome corpus [19, 20]) in different languages such as English, German and Spanish.

However, very few works were developed for Arabic language. To our knowledge, there is only one work achieved at Memphis University [21] that proposes speech acts classification model including the following set of predefined categories: *assertion, declaration, denial, expressive evaluation, greeting, indirect request, question, promise/denial, response to question, and short response.* This tagset includes general-purpose actions that can be applied to independent domain corpora. Nevertheless, these acts are incomplete to build discourse structure and are unable to describe argumentative structure. Thus, this annotation schema cannot annotate argumentative acts related to exchanging opinions, arguments, acceptations, rejects, etc.

3 Argumentative Discourse Structure

Dialogue acts play a fundamental role in the identification of discourse structure.

In this context, [22] claim about task structure influencing dialogue structure. It seems likely that there are structures higher than a single utterance, yet more fine grained than a complete dialogue. Several researchers identify structures within dialogue at levels higher than individual utterances or speaker turns, but below the level of complete discourse description. There has been some significant exploration of the use of sequences of Dialogue Acts, at a number of levels of granularity.

The simplest dialogue sequence model is the use of adjacency pairs [23] which are functional links between pairs of utterances such as question/answer, opinion request/opinion, etc.

Within the adjacency pairs model, the importance of tracking a deeper structured representation based on argumentation theory has been recognized in [24, 25]. These models help in constructing the argumentative information needed to express participants' intentions and to answer real user queries. A simple but expressive model of an argumentative structure is the "Issue Based Information Systems" (IBIS) model, proposed by [26] and adopted as a foundational theory in some computer-supported collaborative argumentation systems. Thus, this model captures and highlights the essential lines of a discussion in terms of what issues have been discussed and what alternatives have been proposed and accepted by the participants.

In our context, the argumentative structure of discussions can be helpful in browsing discussed topics, decisions, agreements and disagreements between participants.

4 DA Annotation Scheme

Our main goal is to track argumentative data from debate TV programs in order to build discourse structure based on dialogue acts set. To perform this task, we defined a suitable set of dialogue act tags relevant for argumentative conversations. Then we used a realistic corpus manually labeled using the dialogue act tag set, which is then used for training the statistical models for automatic dialogue act annotation.

In a first step, [27] proposed a DA taxonomy consisting of 40 acts divided into five main categories as: *Social Obligation Management, Dialogue Management, Argumentative, Request* and *other*. The given categories can be applied for other languages and can be common across other annotation schemes especially those tracking argumentative data. The proposed categories are detailed below:

- **Social Obligation Management:** includes conventional acts such as *opening, closing greetings* and *introducing*, in addition to the expressive acts following Searle's classification as *thanking, apology, regret* and *polite formula*.
- **Turn Management:** used to elicit and provide feedback in order to perform turn speaking management in the discussion like *acknowledgement*, *calm, clarification, feedback, out of topic* and *non-understanding signal* acts.
- **Request:** includes initiatives often called forward-looking acts. Request utterances can express several kinds of demands such as:

  Factual questions (q): non-opinion questions including yes/no questions, definition questions, questions requiring named entity as answer.
  Non-factual questions: opinion questions expressing confirmation, explanation, justification and Opinion requests.
  Other request acts: in Arabic language request construction " الإنشاء الطلبي" includes the following actions: *hope, wish, invoke, warn* and *order.*

- **Argumentative:** is mainly based on exchanging opinions, accepting or rejecting others ideas. It's the fact to convince others by giving arguments, explanations, examples, etc. The main acts including in this category are: *Opinion, Accept, Reject, Argument, Justification, Explanation, Confirmation, Conclusion, Hypothesis and Answer.*
- **Other:** includes non-interpretable and non-classifiable utterances.

The Kappa statistic was used to compare inter-annotator agreement [28]. Labeling of 800 utterances was carried out by two experts[1] with an agreement of 84% resulting from Kappa statistic, which is a satisfactory indication that our corpus can be labeled with high reliability using our annotation scheme.

A first experiment was yield to automatically annotate a set of 2364 utterances (8 discussions). We got low evaluation results with SVM (Precision=33.8%, Recall=31.7%, F-Measure=29.8%), Naive Bayes (Precision=35.8%, Recall=29.5%, F-Measure=25.2%) and Decision Trees classifiers (Precision=24.5%, Recall=25%, F-Measure=24.3%).

In order to improve the obtained results within the automatic annotation process, we extended the training corpora to 6050 utterances (22 discussions). We also reduced the initial tagset to 19 acts in a second step. We merge acts expressing social obligation management into a single dialogue act named *SOM*. We also combine acts expressing Turn Management in one act labeled *TM*. We eliminate acts having very few occurrence in the corpus like *statement*, *propose, hope, wish, invoke, warn* and *order.*

[1] Computational linguistics teachers.

We also eliminate the following tags expressing Appreciation (*app*), disapproval (*disap*), partial accept (*part_acc*) and partial reject (*part_rej*). In fact, we considered appreciation and partial accept acts as acceptation tags while disapproval and partial reject was considered as forms of reject. We add the tag *Thesis* in the argumentative category referring to a new topic or issue introduced by the presenter that can be retained or rejected by the audience. A complete list of the 19 dialogue acts we used in the second step is shown in Table 1 along with the occurrence number (*occ*) and the frequency (*freq*) of each dialogue act in our corpus.

Table 1. DA annotation scheme

Dialogue Act		Tag	Occ	Freq.%
Social Obligation Management		***SOM***	***540***	***8.93***
Turn Management		***TM***	***794***	***13.12***
Request			***789***	***13.04***
Question	استفهام	*Q*	*434*	*7.17*
Confirmation Request	طلب توكيد	*Conf_Req*	*104*	*1.72*
Explanation Request	طلب تفسير	*Expl_Req*	*63*	*1.04*
Justification Request	طلب تعليل	*Justif_Req*	*41*	*0.68*
Opinion Request	طلب إبداء الرّأي	*Op_Req*	*147*	*2.43*
Argumentative			***3780***	***62.48***
Thesis	أطروحة	*Thesis*	*229*	*3.78*
Opinion	إبداء الرّأي	*OP*	*1258*	*20.79*
Accept	موافقة	*Acc*	*158*	*2.61*
Reject	رفض	*Rej*	*316*	*5.22*
Argument	حجّة	*Arg*	*378*	*6.25*
Justification	تعليل	*Justif*	*299*	*4.49*
Explanation	تفسير	*Expl*	*417*	*6.89*
Confirmation	توكيد	*Conf*	*275*	*4.55*
Conclusion	استنتاج	*Conc*	*241*	*3.98*
Hypothesis	فرضية	*Hyp*	*96*	*1.59*
Answer	إجابة	*Ans*	*113*	*1.87*
Other	**أخرى**	***Oth***	***161***	***2.66***
Total			***6050***	

5 Automatic Annotation Task

5.1 Training Corpora

To automatically recognize Dialogue Acts, we experiment a set of human-human conversations collected from Aljazeera debate TV programs[2] discussing hot politic topics (Tunisian and Egyptian revolutions, Syrian war, Tunisian elections, etc). The choice of this corpus is argued by the important argumentation hold in its content mainly conveyed by exchanging opinions, agreements, disagreements, etc.

[2] www.Aljazeera.net

The training data was manually annotated by human experts using the ActAAr annotation tool (**Act A**nnotation in **Ar**abic) [27]. Basic information of the used learning corpora is detailed in the table below:

Table 2. Training corpora statistics

Total number of conversations	*22*
Total number of turns	*1805*
Total number of utterances	*6050*
Total number of words	*101169*
Average number of turns/conversation	*82*
Average number of utterances/conversation	*275*
Average number of words/conversation	*4599*
Average number of words/utterance	*16*
Average number of utterances/turn	*3*
Average number of participants/conversation	*6*

5.2 Learning Features

Textual features are the most widely used for DA classification [9], [16, 17], [29, 30, 31, 32, 33]. They include lexical, syntactic, and structural features.

In our context, we explored a set of lexical, morpho-syntactic, utterance and structural learning features we judged most effective to our task.

Lexical Features (LF)

Question words: indicate strongly if the speaker is asking a question or requiring a specific kind of request (explanation request, opinion request, etc). For example the word "how" "كيف" is generally used for an explanation request.

Cue phrases: the most common key words or expressions can serve as useful indicators of Dialogue Acts recognition. These cue phrases are words or groups of words that appear to be reliable indicators to predicate the convenient act for each utterance. Cue phrases occurred frequently as unigrams (*"ok" "طبعا", "yes" "نعم", "I think" "أعتقد", "I agree" "أوافق"*) or bigrams ("*of course*" "بطبيعة الحال", "*good evening*" "مساء الخير"). Negative verbs are also formed from a combination of two words in Arabic language (*"I don't think" "لا أعتقد", "I don't agree" "لا أوافق"*).

Morpho-syntactic Features (MF)

At this level, we used the MADA analyzer (Morphological Analyser and Disambiguisator of Arabic) [34]. This tool can be used for several NLP tasks as tokenization, lemmatization, morphological and morpho-syntactic analysis. In our work we applied MADA to extract word lemma, word category (POS) and verb tense (present or past). This tool shows high reliability in Arabic processing tasks as it performs data disambiguation. We encode:

Part-of-speech tags: we enhance the first word POS tagset (verb, noun, adjective, adverb, etc).

First Verb tense: generally verbs indicate the action done by the speaker. They are used to predict whether the speaker was presenting his opinion by using the present or just stating past events using the past tense.

Utterance Features (UF)
Previous research showed that utterance meta information can help classify DAs [30], [35]. Here we encode:

Utterance speaker: the actor of the current utterance.

Speaker role: whether the speaker is the animator of the discussion or just a participant. Mostly, the animator introduces and ends the discussion, manages the participants' turn taking, makes requests and asks questions.

Structural Features (SF)
Dialogue history features model what happened before the current utterance [17], [29]. We encode:

Previous act: the DA sequence in the conversation can help to anticipate the next DA label. For instance, a confirmation request is generally followed by a confirmation.

Previous utterance speaker: it is important to identify whether the previous utterance has the same actor as the current one.

5.3 Obtained Results

We run experiments classifying the DA tag for the current utterance. We use supervised learning approaches, namely Naive Bayes (NB), Decision Tree (J48) and SVM model (SMO). These algorithms are widely used for DA classification [29], [32, 33], [35]. We used the Weka package implementation [36] for the proposed models. All evaluation results shown below were carried out using 10 fold cross validation.

Results show that SMO classifier performs better for annotating dialogue acts with precision of 52.6%, recall score of 52.4% and f-score value of 51.2% when combining linguistic, discursive and structural characteristics. Classifiers performance decline when integrating N-Gram model. This is due to the fact that similar words or sequences of words are not so frequent in the corpus.

Given that annotation tags were clustered into five main classes, we started by evaluating the acts categories' classification task. Results detailed in table 6 show high reliability especially for argumentative class. This can be explained by the fact that argumentative tags generally starts with specific expressions like opinion words (*"أعتقد" "I think"*), negative words (*"لا" "No"*) and conjunctions (*"لأن" "because"*).

Table 3. SMO results

Features	*Precision*	*Recall*	*Fscore*
LF	*0.5*	*0.507*	*0.491*
LF+MF	*0.503*	*0.51*	*0.493*
LF+MF+UF	*0.511*	*0.519*	*0.503*
LF+MF+UF+SF	***0.526***	***0.524***	***0.512***
LF+MF+UF+SF+unigrams	*0.47*	*0.416*	*0.446*
LF+MF+UF+SF +bigrams	*0.4*	*0.341*	*0.311*
LF+MF+UF+SF +trigrams	*0.279*	*0.315*	*0.27*

Table 4. NB results

Features	*Precision*	*Recall*	*Fscore*
LF	*0.483*	*44.9*	*0.426*
LF+MF	*0.476*	*0.45*	*0.427*
LF+MF+UF	*0.474*	*0.47*	*0.442*
LF+MF+UF+SF	***0.48***	***0.487***	***0.457***
LF+MF+UF+SF+unigrams	*0.372*	*0.355*	*0.365*
LF+MF+UF+SF+bigrams	*0.367*	*0.36*	*0.319*
LF+MF+UF+SF+trigrams	*0.282*	*0.349*	*0.288*

Table 5. J48 results

Features	*Precision*	*Recall*	*Fscore*
LF	*0.401*	*0.423*	*0.372*
LF+MF	*0.425*	*0.441*	*0.388*
LF+MF+UF	*0.44*	*0.437*	*0.433*
LF+MF+UF+SF	***0.453***	***0.464***	***0.446***
LF+MF+UF+SF+unigrams	*0.367*	*0.40*	*0.319*
LF+MF+UF+SF+bigrams	*0.303*	*0.385*	*0.279*
LF+MF+UF+SF+trigrams	*0.279*	*0.315*	*0.27*

Table 6. SMO classification results

Precision	*Recall*	*Fscore*	*Act*
0.888	*0.779*	*0.83*	*Som*
0.617	*0.37*	*0.463*	*TM*
0.921	*0.802*	*0.858*	*Request*
0.811	***0.95***	***0.875***	***Argumentative***
0.2	*0.024*	*0.042*	*Other*
0.79	***0.813***	***0.791***	***Weighted Avg.***

Detailed results of the automatic annotation process of the entire tag set consisting of 19 dialogue acts are illustrated in table 7. Results show that Social Obligation Management acts are highly predicted (*Fscore=0.8*). This can be explained by the presence of cue words/phrases used for opening, closing, greeting and thanking utterances. Justification and conclusion dialogue acts are also relatively well classified due to the presence of relevant lexical markers (“لأن” “*because”* for justification and

"إذن" "*so*" for concluding). Moreover, the use of question words and markers help classifiers to predict question acts (*Fscore=0.679*). Analysis of the confusion matrix obtained from the 19 acts tagging within our corpus indicates that the most common misclassifications are confirmation requests as questions (54.3 %); explanation requests as questions (49.06%); justification requests as questions (47.37 %) and agreements as Turn Management (Acknowledgement);We also notice that 51.85% of arguments and 44.19 % of answers were wrongly classified as opinions. Agreements are also ambiguous and were confused with turn management tags (25.32%), in fact, the word "نعم" "yes" used frequently in agreements can express an acknowledgement act.

Table 7. Detailed annotation results per act using SMO

Precision	*Recall*	*Fscore*	*Act*
0.814	***0.787***	***0.8***	***Som***
0.421	*0.492*	*0.454*	*Tm*
0.667	***0.692***	***0.679***	***Q***
0.286	*0.076*	*0.12*	*conf_req*
0.6	*0.316*	*0.414*	*justif_req*
0.318	*0.178*	*0.228*	*op_req*
0.444	*0.226*	*0.3*	*expl_req*
0.453	*0.685*	*0.545*	*Op*
0.362	*0.367*	*0.364*	*Thesis*
0.375	*0.222*	*0.279*	*Arg*
0.398	*0.19*	*0.258*	*Conf*
0.675	*0.542*	*0.601*	*Hyp*
0.775	***0.556***	***0.647***	***Conc***
0.539	*0.514*	*0.526*	*Expl*
0.879	***0.624***	***0.73***	***justif***
0.548	*0.557*	*0.553*	*Rej*
0.507	*0.215*	*0.302*	*Acc*
0.196	*0.059*	*0.091*	*Oth*
0.328	*0.196*	*0.246*	*Ans*
0.526	***0.524***	***0.512***	***Weighted Avg.***

6 Discussion

The classification errors predominantly occur due to the ambiguity of distinguishing between different types of request and factual questions as they can have common question words and punctuation markers.

Difficulties in detecting argumentative acts such as explanations, justifications, arguments and opinions are generally related to the urgent need of pragmatic information involving the enunciation context. Thus, linguistic features are not sufficient to predict the right dialogue acts tags.

Disfluencies occurring in spoken conversations can lead to incorrect tags especially when using dialectical expressions or words from other languages ("ok", "merci").

Therefore, non voyelled words can express different meanings. For instance, the word "إذا" can take the following diacritized form "إذًا" "So" used for concluding. The same word can have another voyelled form "إذَا" "if" which expresses a condition.

The classification results are relatively reliable compared to the complexity of the annotation task especially of argumentative tags.

These results can be improved when including context-based features and even when enlarging the training corpora.

7 Conclusion and Future Work

In this paper, we proposed an automatic dialogue act annotation method for Arabic argumentative debates. We experimented supervised learning algorithms to perform the annotation process including lexical, morpho-syntactic, discursive and structural features. To run learning experiments, we used specific corpora involving mainly argumentative acts annotated manually by human expert using the ActAAr annotation tool. We obtained fairly satisfying results when using SVM classifier with an average accuracy of 53% (precision = 52.6%, recall =52.4% and f-score=51.2%).

As a future work, we intend to take into account the pragmatic information hold in argumentative conversations by including context-based characteristics in the training process. We also think that it is useful to integrate syntactic patterns presenting the words sequences order in each utterance instead of considering an utterance as a simple bag of words.

We intend extend the training corpora to improve dialogue act recognition results.

References

1. Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G.: The HCRC Map Task Corpus. Language and Speech 34, 351–366 (1991)
2. Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., Anderson, A.: The reliability of a dialogue structure coding scheme. Computational Linguistics 23, 13–31 (1997)
3. Walker, M.A., Moore, J.D.: Empirical Studies in Discourse. Computational Linguistics 20(2) (1997)
4. Allen, J., Core, M.: Draft of DAMSL: Dialog Act Markup in Several Layers. Technical report (January 1997)
5. Jurafsky, D., Bates, R., Coccaro, N., Martin, R., Meteer, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P., Van Ess-Dykema, C.: Automatic Detection of Discourse Structure for Speech Recognition and Understanding. In: Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding, Santa Barbara (1997)
6. Bunt, H.: Dimensions in dialogue annotation. In: Proceedings of LREC 2006 (2006)
7. Bunt, H.: The DIT++ taxonomy for functional dialogue markup. In: Heylen, D., Pelachaud, C., Catizone, R., Traum, D. (eds.) Proceedings of the AAMAS 2009 Workshop Towards a Standard Markup Language for Embodied Dialogue Acts (EDAML 2009), Budapest, May 12 (2009)
8. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)

9. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., Meteer, M.: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics 26(3), 339–373 (2000)
10. Singh, S., Kearns, M., Litman, D., Walker, M.: Reinforcement learning for spoken dialogue systems. Advances in Neural Information Processing Systems 12, 956–962 (1999)
11. Martınez-Hinarejos, C.D., Benedı, J.M., Granell, R.: Statistical framework for a Spanish spoken dialogue corpus. Speech Communication 50, 992–1008 (2008)
12. Bangalore, S., Di Fabbrizio, G., Stent, A.: Learning the structure of task-driven human–human dialogs. In: Proceedings of ACL, Sydney, Australia, pp. 201–208 (2006)
13. Rangarajan Sridhar, V.K., Bangalore, S., Narayanan, S.: Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework. In: Proceedings of NAACL-HLT (2007)
14. Webb, N., Hepple, M., Wilks, Y.: Dialogue act classification based on intra-utterance features. In: Proceedings of the AAAI Workshop on Spoken Language Understanding, Pittsburgh, PA (2005)
15. Hoque, M.E., Sorower, M.S., Yeasin, M., Louwerse, M.M.: What Speech Tells us about Discourse: The Role of Prosodic and Discourse Features in Dialogue Act Classification. In: IEEE International Joint Conference on Neural Networks (IJCNN), Orlando, FL (August 2007)
16. Bangalore, S., Di Fabbrizio, G., Stent, A.: Learning the structure of task-driven human–human dialogs. IEEE Transactions on Audio, Speech, and Language Processing 16(7), 1249–1259 (2008)
17. Di Eugenio, B., Xie, Z., Serafin, R.: Dialogue act classification, higher order dialogue structure, and instance-based learning. Dialogue & Discourse 1(2), 1–24 (2010a)
18. Core, M., Allen, J.: Coding Dialogs with the DAMSL annotation scheme. In: AAAI Fall Symposium on Communicative Action in Humans and Machines. MIT, Cambridge (1997)
19. Levin, L., Thymé-Gobbell, A., Lavie, A., Ries, K., Zechner, K.: A discourse coding scheme for conversational Spanish. In: Fifth International Conference on Spoken Language Processing (1998)
20. Ries, K.: HMM and neural network based speech act detection. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Phoenix, AZ, pp. 497–500 (1999)
21. Shala, L., Rus, V., Graesser, A.C.: Automated speech act classification in Arabic. Subjetividad y Procesos Cognitivos 14, 284–292 (2010)
22. Grosz, B., Sidner, C.: Attention, Intentions, and the Structure of Discourse. Computational Linguistics 19(3) (1986)
23. Schegloff, E.A., Sacks, H.: Opening Up Closings. Semiotica 7, 289–327 (1973)
24. Pallotta, V., Ghorbel, H., Ruch, P., Coray, G.: An argumentative annotation schema for meeting discussions. In: Proceedings of the 4th International Conference on Language Resources (LREC 2004), Lisbon, Portugal, May 26-28, pp. 1003–1006 (2004)
25. Galley, M., McKeown, K., Hirschberg, J., Shriberg, E.: Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In ACL 2004, Barcelona (2004)
26. Kunz, W., Rittel, H.W.J.: Issues as elements of information systems. Technical Report WP-131, Berkeley: University of California (1970)
27. Ben Dbabis, S., Mallek, F., Ghorbel, H., Belguith, L.: Dialogue Acts Annotation Scheme within Arabic discussions. In: Sixteenth Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2012), Paris, France, September 19-21 (2012)

28. Carletta, J.C.: Assessing agreement on classification tasks: the kappa statistic. Computational Linguistics 22(2), 249–254 (1996)
29. Sridar, V.K.R., Bangalore, S., Narayanan, S.S.: Combining lexical, syntactic and prosodic cues for improved online dialog act tagging. Computer Speech & Language 23(4), 407–422 (2009)
30. Kim, S.N., Cavedon, L., Baldwin2010, T.: Classifying dialogue acts in one-on-one live chats. In: Proceedings of EMNLP 2010, the Conference on Empirical Methods in Natural Language Processing, pp. 862–871. Association for Computational Linguistics (2010)
31. Boyer, K.E., Grafsgaard, J.F., Ha, E.Y., Phillips, R., Lester, J.C.: An affect-enriched dialogue act classification model for task-oriented dialogue. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-, vol. 1, pp. 1190–1199. Association for Computational Linguistics (2011)
32. Ha, E.Y., Grafsgaard, J.F., Mitchell, C., Boyer, K.E., Lester, J.C.: Combining verbal and nonverbal features to overcome the "information gap" in taskoriented dialogue. In: Proceedings of SIGdial 2012, the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Seoul, South Korea, pp. 247–256. Association for Computational Linguistics (July 2012)
33. Kim, S.N., Cavedon, L., Baldwin, T.: Classifying dialogue acts in multi-party live chats. In: Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation, Bali, Indonesia, pp. 463–472. Faculty of Computer Science, Universitas Indonesia (2012)
34. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt (2009)
35. Ivanovic, E.: Automatic instant messaging dialogue using statistical models and dialogue acts. Master's thesis, University of Melbourne (2008)
36. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)

E-Quotes: Enunciative Modalities Analysis Tool for Direct Reported Speech in Arabic

Motasem Alrahabi

Paris-Sorbonne University in Abou Dhabi,
Abou Dhabi - UAE
motasem.alrahabi@gmail.com

Abstract. With rapidly growing Arabic online sources aimed to encourage people's discussions concerning personal, public or social issues (*news, blogs, forums…*), there is a critical need in development of computational tools for the Enunciative Modalities analysis (*attitude, opinion, commitment…*). We present a new system that identifies and categorizes quotations in Arabic texts and proposes a strategy to determine whether a given speaker's quotation conveys some enunciative modalities and potentially its evaluation by the enunciator. Our system enables two query types search for keywords within the "categorized" quotations: searching for keywords in the part potentially containing the reported speech source (*the reporting clause*) or searching for keywords in the part concerning the topic (*the reported clause*). The annotation is performed with a rule-based system using the reporting markers' meaning. We applied our system to process a corpus of Arabic newspaper articles and we obtained promising results for the evaluation.

Keywords: Direct reported speech, Enunciative Modalities, Opinion Mining, Sentiment Analysis, categorization, Arabic language, rule-based system.

1 Introduction

The Reported Speech (RS) is an important linguistic phenomenon characterized by its syntactic structure: a matrix clause usually containing a reporting marker, and a subordinate clause embedding the conveyed information [1]. Among the various forms of RS (*direct speech, indirect paraphrases, direct speech introduced by "that"…*), we are particularly interested in the Direct RS (*quotations*). Many text mining applications use quotations to analyze, organize and summarize information because they are a major vehicle of communication in the news genre. We believe that a tool which identifies and semantically categorizes quotations would enable readers, journalists or researchers to place news in the context of all comments made on a given topic, and specifically to know how these comments were interpreted in the media.

Our work heads in this direction since we aim to automatically detect and categorize Arabic quotations according to the enunciative modalities. The automatic identification and interpretation of modalized statements in texts is a major concern in a large number of applications, especially with the recent attention to Opinion Mining,

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 479–490, 2015.
DOI: 10.1007/978-3-319-18111-0_36

Semantic Analysis and Appraisal Theory [2]. In our approach, enunciative modalities concern the manner in which the enunciator reports, interprets and evaluates the words of the speaker (*disengagement, commitment, attitude…*). It also concerns the manner by which the speaker expresses an attitude towards his interlocutor and towards the contents of his utterance (*commitment, control, opinion, judgment...*). For this purpose, we rely on the semantic *reporting markers* that introduce and modalize the reported words. Let us consider this sentence:

ولحسن الحظ اعترف أوباما "بعدم قدرة أميركا لوحدها على ضمان الأمن والسلام العالميين".
Fortunately, Obama acknowledged "the inability of the United States alone to ensure international peace and security."

The different elements are analyzed as follows:

Element		Label
أوباما	*Obama*	*Speaker (Source)*
اعترف	*acknowledged*	*Speaker commitment*
ولحسن الحظ	*Fortunately*	*Enunciator (author) positive attitude*

Our primary contributions making our current research significant are: i) developing linguistic resources (*markers and rules*) to identify and categorize quotations from texts in Arabic; ii) creating an operational application which allows users to directly query an annotated corpus by both classical and semantic criteria.

The remainder of this paper is organized as follows: first we show how difficult semantic analysis from quotations can be (§2). We describe our proposed method (§3) and give an overview of the system (§4). In (§5), we present the evaluation results and discuss them. We present in (§6) the related work and finally, in (§7) we draw our conclusions and future work.

2 Automatic Analysis Challenges from Direct Reported Speech

We situate our current work on Direct RS in the field of Opinion Mining and Sentiment Analysis which we consider as a part of modalities studies. In fact, the RS is a standardized way to relate opinion, sentiment or attitude expression of a certain Source regarding a certain Target. Inspired by Banfield, Uspensky and Quirk, the author of [3] considers that Subjectivity refers to aspects of language used to express opinions, feelings, evaluations, and speculations, including sentiments. Therefore, from a computational point of view, current research distinguishes between subjectivity and objectivity in opinions along with determining these elements: Opinion polarities which tell us whether the opinion's orientation or valence is positive, neutral, negative or, sometimes, mixed; the opinion strength (*attitude's degree, i.e., low, medium, high*); opinion holder (*the people or person expressing the opinion*), and opinion target (*the object of this opinion*).

Nevertheless, characterizing the opinions and sentiments analysis from quotations remains challenging for at least these three reasons: the target, the source and the expressed opinion:

- **Target:** In Opinion Mining (OM) over movie, product or book reviews, the target or topic is clearly identified. On the contrary, the target in news articles is not a concrete object [4] because when a text is argumentative and when it opposes different points of view, journalists may span larger subject domains, more complex event descriptions and a whole range of targets [5]. Thus, identifying a concrete target that can be resolved back to named entities recognition (NER) does not work for quotations [6], because quotes may not necessarily mention the debate topic (*implicit targets*), and there may be multiple relevant targets for a single topic (*mixed speeches, selective and partial targets*).
- **Source:** The opinion source (*holder*) identification aims to extract entities that express opinions in texts [7]. There are many challenges in the task of automatically attributing each quote to its correct speaker [8]. Sometimes, the source may not be located near the quotation, so syntactic parsing and NER may be necessary. The use of pronouns is also common, so that anaphora and co-reference resolution are needed to determine the name of the source. In the case of quotations, a source can be title or role (*Prime Minister*); proper name (*Vladimir Poutine*); pronominal reference (*she said…*); anonymous (*a passenger, a witness…*). In the following cases in Arabic, the source of quotation is not explicitly mentioned:

إليكم الخبر الآتي: ...	* *Take the following news: ...*
أتانا ما يلي: ...	* *We got the following: ...*
جاء في البيان أن...	* *It came in the report that…*
نص القرار على...	* *The decision consisted of …*

  For Arabic language, several challenges complicate the opinion source identification [9]: the lack of resources, the high inflectual nature of Arabic language, the variant sources of ambiguity, the rich metaphoric script usage and the absence of robust Arabic parser that understand the sentence structure [10].

- **Expressed Opinion:** Most work on OM has been carried out on subjective text types such as product reviews, blogs or even social media [11], where individuals express their opinions quite freely. On the contrary, the position of the journalist in relation to what s/he reports in newspaper articles is often more subtle [12], because the authors of newspaper articles try to make their articles look objective concerning the topics they are covering. In these cases, opinion or sentiment is not always expressed explicitly in the text. However, journalists try to remain flexible in exhibiting their attitude towards what they report : it goes for instance by highlighting some facts while omitting others, but especially by the choice of words to introduce the RS and to describe the position of the different actors of the original utterance situation.

For all these reasons, the OM and more generally the semantic analysis for quotations may not guarantee perfect results. Our current aim in this study is not trying to tackle all of these complex issues, but to focus our efforts on the last point (expressed opinion) for Arabic language, i.e. how a quotation is reported and interpreted by the enunciator.

3 Our Proposed Approach

Here, we will describe the different aspects of our approach for quotations identification and categorization.

3.1 Markers and Structures of Direct RS

We consider, on the formal level, that a Direct RS is any kind of speech delimited by meta-characters (*the typographical signs of quotations*) and introduced by, at least, one reporting marker referring to an act of communication, whether the speaker is explicitly defined or not. By convention, we consider the quotation span a verbatim transcription of the source utterance, despite the existence of rare cases where the quoted words are not certainly attributed to the speaker, as in the following case:

ولسان حال الشاب الذي يتقدم لزواجها يقول: "لا يصلح العطار ما أفسد الدهر".
It's as if the young man who is proposing to her said:
"the perfumer cannot fix what time has spoiled"

Here are some examples of the different possible constructions in Arabic with verbal, nominal or adverbial reporting markers:

يشير فلان الى أن " ...".	*X points that "..."*
يؤكد فلان : " ...".	*X affirms: "..."*
" ..."، أضاف فلان.	* "..." X added.*
يحتجّ فلان قائلاً : " ...".	*X objects saying: "..."*
في ما يلي تصريح فلان: " ...".	*Following is the declaration of X: "..."*
أرسل فلان التعليق الآتي: " ...".	*X sent the following comment: "..."*
بحسب فلان : " ...".	*According to X: "..."*
" ..." بحسب ما يقول فلان.	*"..." according to what X is saying.*
...	...

Sometimes, one or more intermediate entities can be part of the transmission chain between the utterer and the speaker. We call this entity the "transmitter". In the following sentence, (*Anadolu news agency* / وكالة أنباء الأناضول) is the speaker, while (*the local authorities* / السلطات المحلية) is the transmitter.

في غضون ذلك ذكرت وكالة أنباء الأناضول نقلاً عن السلطات المحلية "تدفق مئات اللاجئين السوريين
في الساعات الأخيرة إلى تركيا".
Meanwhile, Anadolu news agency mentionned, according to the local authorities :
"the flow of hundreds of Syrian refugees in the last hours to Turkey."

By ignoring the aspecto-temporal parameters in this analysis, the standard metalinguistic formula of a *reporting act* can be expressed by *operators* acting onto *operands*:

I-SAY (T_n-SAYS (X-SAYS (λ) to X') to T_n') to YOU

where the operator "**I-SAY**" represents the original utterer's act of speaking. "T_n" represent the transmitter(s) and "**X**" the speaker (*the source*). The symbols "**YOU**","T_n'" and "**X'**" represent respectively the interlocutors of the Enunciator (I), the Transmitter (T) and the Speaker (X). Finally, the "**λ**" represents the reported speech.

3.2 Semantic Categorization

We distinguish between the enunciator modalities and those attributed by the enunciator to the speaker (*in this current work we don't analyze the content of quotes*). In the enunciative approach [13], the enunciator (*utterer*) is the entity that reports the whole speech (*generally the author*), whereas the speaker (*or locutor*) is the last source or speech holder. For instance, when an enunciator uses the verb *to claim*, he attributes a modality to the speaker (*commitment*), in addition, he exhibits his own position towards the speaker's credibility.

Moreover, our added value is that the analysis covers a larger scale of semantic phenomena which are not easily classifiable when using only the categories of positive and negative opinions. Actually, the enunciator can use the mechanism of RS not only to reproduce the original utterance, but also to interpret it and to provide other information using several types of markers:

i) Reporting markers that introduce the quotes (*according to, to inform…*) ;

ii) Modality markers or modalizers [14] that indicate:

- the position of the enunciator towards the speech act (*unfortunately she admitted..*) or towards its characteristics (*by adding this short comment…*);
- the attitude of the speaker (*he said with skepticism that …*);
- circumstantial information which clarifies the speech act of the original utterance: spatio-temporal and audience settings [1], theme or topic (*concerning / about…*), communication medium (*He said in a letter…*).

In our current analysis, we only refer to the observable *reporting markers* which are lexically expressed in texts. An empirical examination of the corpora[1] allowed us to identify more than 150 reporting markers. The latter have been manually listed and organized with their derived forms (*gerunds or nouns*) into a semantic map (*linguistic ontology*) which includes, for instance, the following categories:

- Neutral: *say, observe, define…*
- Positive opinion: *encourage, praise…*
- Negative opinion: *criticize, denounce…*
- Commitment: *confirm, affirm, believe…*
- Disengagement: *deny, refute…*
- Control: *order, decide, refuse…*
- Speech organization: *add, ask, answer, conclude…*

Due to the polysemy phenomenon, a given marker can naturally belong to one or more categories. The general enunciative formula will become more complex to receive the operators **OP** which represent the different values of the semantic map:

$$\mathbf{I\text{-}SAY(OP_I(T_n\text{-}SAYS(OP_T(X\text{-}SAYS(OP_X(\lambda))to\ X')to\ T_n')))to\ YOU}$$

The enunciative dimension of our analysis permits us to associate to each set of markers whether the speaker is "sending" information (*X informs X'…*), "receiving" information (*X reads in the journal that…*), "transmitting" information (*X reports that…*) or even he is totally absent as a source (*I heard that…*).

[1] Our corpus is a collection from internet-based Arabic media (Al-Jazeera, BBC Arabic, CNN Arabic, Al-Nahar, Le Monde Diplomatique in Arabic…).

4 System Overview

In this section, we describe the system pipeline configuration and how it is deployed in practice. For processing resources, we use EXCOM-2 [15], a rule-based system that performs annotations by using surface markers and heuristic rules. Annotated texts are indexed for rapid retrieval at query time with Solr search engine platform.

4.1 Corpus Preparation

The starting point of the system is the corpus preparation. Technically, to annotate a corpus, EXCOM-2 needs a pre-treatment phase of segmentation (*splitting*). It helps in determining the search fields for linguistic markers and the textual snippets which are to be annotated. This consists in defining by heuristic rules the boundaries of sections, titles, paragraphs and sentences. For this, all corpus documents have to be normalized and converted to raw texts files in UTF-8 encoding.

4.2 Annotation: Quotations Recognition and Categorization

The core of the system is the semantic annotation task. In our perspective, we consider two types of surface markers: "indicators" and "clues". The presence of a potential indicator in the search space triggers the associated CE rules, and then, additional clues are searched in a specified context. If all the rule conditions are satisfied, the segment specified by the rule will be annotated and assigned one or more semantic values. For our processing, we consider quotations marks as indicators and reporting markers as clues.

Different types of rules can be implemented in EXCOM-2, depending on the research space or the type of markers (*linguistic units, regular expressions, text structure tags*…). The tool enables to use the already annotated segments, to use the text structure (*titles, paragraphs*…), to sort the rules according to their importance and to use negative clues that can inhibit certain rules. For our current task, the annotation of each semantic category requires the creation of three rules on average.

EXCOM-2 is based on the Contextual Exploration (CE) method [16]. The tool does not deal with any preliminary morpho-syntactic analysis or NER. We believe that this system can be an addition to these approaches, not a substitution (*for example, identifying the quotations sources by NER*). A basic version of the tool is available online at this address: http://www.excom.fr/.

4.3 Indexing and Web Interface

The output of the annotation processing pipeline is indexed using the Apache Solr framework, which is based on the Lucene engine. We store all annotated XML documents in an inverted index that enables flexible search for keywords in all annotated quotations.

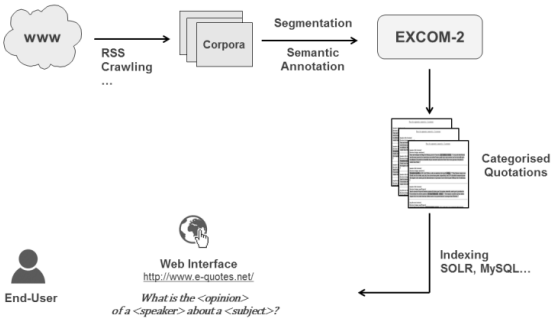


Fig. 1. General architecture of E-Quotes

E-Quotes end-user web interface supports search in three different ways:

- Search for keywords in all identified quotations;
- Filter the quotations according to a specific category and then search for keywords in one of its sub-categories. For example, the user can search for a word in all the quotations that are annotated as "Negative Opinion" and more especially in those that hold the value "Accusation" ;
- Filter the quotations according to a specific category or sub-category and then search for keywords in one of the two options:
  - the space containing the quotation's topic (*the reported clause*) ;
  - the space containing potentially the quotation's source (*the reporting clause*).

Since we do not proceed to the recognition of speakers or targets, the last feature allows user to find answers to such a question:

`What is the <attitude> of a <speaker> towards a <subject>?`

where the parameters <speaker> and <subject> are specified by the user as combination of keywords or entities. The list of <attitude> (*or enunciative modalities*) categories is automatically extracted from the semantic map and proposed to the user.

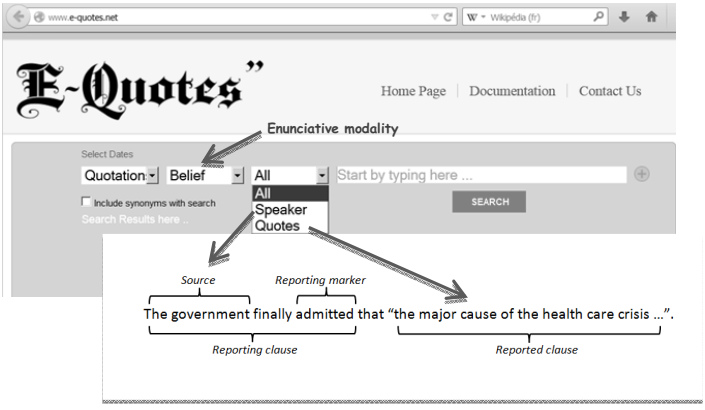


Fig. 2. E-Quotes homepage

The system supports boolean combinations of multiple fields, i.e. AND, OR, NOT.

5 Evaluation

We conducted two evaluations by computing traditional measures in order to test the capacity of our system to identify and to categorize quotations. We also performed a detailed analysis of error cases introduced by our system and their root causes.

5.1 Quotations Recognition Evaluation

We randomly selected 21 new documents from online newspaper articles which included 1049 sentences and over 25000 words. Topics covered in these articles are mostly political, economic, social news and events. We then annotated these texts with EXCOM-2 that identified exactly 269 quotations. In parallel, we asked three Arabic native speakers with language-related academic background to read the selected articles and to highlight manually only the snippets they judge as quotations. After the comparison, we obtained the following results: 89 for recall and 97 for precision.

We conducted a manual inspection over all evaluation documents to identify detection errors. Here are the recall result analysis:

i - The value of silence is due to the fact that some markers are not yet covered by our resource collection, such as nominal markers and gerunds derived from reporting verbs: *declaration, by adding...*

ii - Some quotations are introduced by markers indicating the speaker's attitude but are not considered as reporting markers:

أما المعنيون فيهزون أكتافهم غير مبالين: "أنظروا ليس هنا سوى أشخاص مسالمين ينشطون من أجل مصلحة المجتمع".

The people concerned shake their shoulders, indifferently: "Look, there is no one but peaceful persons working for the sake of their society".

Concerning the precision rate, we can mention the cases of misguiding quotation marks and the polysomic reporting markers.

i - A large number of noise is usually caused by the presence of misguiding quotation marks (*quotation marks that do not surround quotes*). Example:

لكن المنظمة الاسرائيلية لحقوق الانسان "بيتسالم" نشرت تقريراً يؤكّد رواية شهود العيان الفلسطينيين.

But the Israeli human rights organization "B'Tselem" published a report confirming the Palestinian eyewitnesses' version.

ii - In Arabic, the surface forms are generally polysemics [17], especially the forms that have a three-letter root (شرح, علق, بين, عبر...). This difficulty is due to the morphological ambiguity in Arabic, caused, above all, by the absence of vocalization, the agglutination and the relatively free word order in a sentence. Here is an example of wrongly assigned quotation, caused by the presence of a polysemic reporting marker (نقلت) in the context of misguiding quotation marks:

ونقلت سيارات الاغاثة التابعة للمنظمة "مواد إغاثية وطبية" إلى داخل المدينة.

The Relief Organization cars have transported « relief and medical materials » to the center of the city.

We also mention a difficulty caused by the nested quotation marks, where a quotation contains another one. This case can produce errors in the annotation.

5.2 Quotations Categorization Evaluation

To obtain a preliminary assessment of the categorization task, we carried out a limited evaluation, mostly to guide our future efforts. Thus, we only tested the following categories of the semantic map: *positive opinion; negative opinion; commitment* and *disengagement*. For this evaluation, we used the same corpus of the first evaluation, and we selected only the well annotated quotations that belong to the aforementioned four categories. We then obtained 57 quotations.

The same three evaluators were asked to tag each quotation and decide whether the text snippet (*the reporting clause*) is being talked about one or more of our categories. Thus, we decided to hide the contents of the quotes (*the reported clause*) to ensure that the annotators will judge based only on the words of the enunciator, without mixing them with the words of the speaker. The conflicts of tagging were resolved using majority voting principle and the average final agreement was 84% between annotators. The system achieved a recall value of 87 and a precision value of 94.

Taking into account the complexity of the analysis, we consider the overall results to be rather good. The major difficulty encountered in the categorization task is the mixed and nested opinions. In fact, different cases can be found:

- One source (*speaker*), several opinion markers. In the following example, the quotation should be annotated as a Definition and a Denunciation:

وقد استنكر المسؤولون الفلسطينيون إطلاق النار هذا وسمّوه "رصاص صوب الشمس."
The Palestinian officials have condemned this shooting and called it "bullets targeting the sun".

- Different sources, different opinion markers. Here's an example:

وصف قائد المجلس العسكري الأعلى تصريح "أوباما" بإرسال الأسلحة للمعارضة بأنه "تصريح شجاع جدا".
The commander of the Supreme Military Council described the declaration of "Obama" to send weapons to the opposition as "a very brave declaration".

This issue makes it often hard to automatically decide to which speaker the system attributes this or that opinion. At that point, we need to refine the linguistic analysis in order to improve the attribution rules.

6 Related Works

Quotation extraction has been previously approached using different techniques and for several languages. But, to our knowledge, there are only few operational systems that detect quotations from Arabic texts, and even less for the OM or enunciative modalities task from quotations [18].

NewsExplorer [19] is developed in the European Commission's JRC[2]. This tool detects quotations from multilingual live news feeds, including Arabic. The system is able to extract quotes, the name of the entity making the quote and also entities mentioned in the quote. According to the authors, the system recognizes quotations only if it successfully detects three parts: the speaker name, the reporting verb and the

[2] http://press.jrc.it/NewsExplorer

quotation. For the English language evaluation, the system aimed for high precision (87.5%) at the expense of low recall, as their data contained many redundant quotes.

[20] propose a quotation extraction and attribution tool from English newspapers. The system is implemented in GATE and combines a lexicon of 53 common reporting verbs and a hand-built grammar to detect constructions that match 6 general lexical patterns. The authors evaluate their work on 7 newspaper articles, which contain 133 quotations. For the detection of reporting verb and source, the system achieved a recall value of 0.79 and a precision value of 1.00, thus an F-measure of 0.88. For quotation span detection the results are: 99% of precision and 74% of recall.

Google's InQuotes application[3] allows users to search for quotes made - in English - by a small selected set of politicians. The web-based interface is structured in topics and displays side-by-side quotes from two actors. Users can search for any keywords in the search area and quotes containing the keywords would be returned. They do not enable search on the speaker itself other than from the selected set, and no implementation details are published about this system.

[21] describe SAPIENS, a system that relies on a deep linguistic processing chain (*NE extraction, anaphora resolution, deep parsing*…) in order to extract quotations from French news with their author and context. The evaluation was carried out both for the span of the quotation and for the correctness of the author. The evaluation found that 19 out of 40 quotes had a correct span and author, while a further 19 had an incorrect author, and 4 had an incorrect span.

We can observe that most of the prior approaches deliberately choose to focus on the more frequent syntactic structures and on limited lists of reporting markers. On the other hand, all these works carry out a pre-recognition of quotations' sources (*holders*) and retain only the quotations where the speaker is identified unambiguously. Finally, none of these systems applies enunciative and semantic analysis like we do in order to automatically categorize quotations according to the reporting markers.

7 Conclusion and Future Works

Our system makes the semantic information explicit and accessible for end-users. We demonstrated it by adapting the standard IR technologies (*i.e. keywords queries matched against bag-of-words document representation*) to semantically tagged natural texts. By indexing semantic annotations using such keyword search engine, we provide a highly scalable and fast semantic search capability by enabling users to search for quotes made by a particular person or about an entity. Each quotation is categorized according to the opinion of the source (*speaker*) and potentially to that of the enunciator. The used method is simple and does not require morpho-syntactic preprocessing or NER. For the categorization task we achieve a recall rate of 87% and a precision of 94%. As future work, we envision to do the following:

- Extend the lexical resources with new markers such as adjectives (*doubtful, boring*…); adverbs (*finally, unfortunately*…) and gerunds (*laughing, shouting*…).

[3] `http://labs.google.com/inquotes/` (*deprecated*).

This allows to have more fine-grained categorization and to analyze the intensity of opinions (*strong, medium or weak…*).

- Evaluate the impact of using markers that modify the polarity of an expressed opinion such as valence shifters (*negations, intensifiers…* [22]), connectives or even modals.
- Classify and analyze the content of quotes (*the reported clause*). This feature will give us a complete vision of the polarity of each quotation. Using a classifier could also help us to assign topic tags to each quote.
- Extend the analysis of RS in Arabic and cover indirect, mixed and unmarked RS forms [23].
- Last, integrate the annotation module as a webservice that can be automatically queried by the user interface in order to directly process new submitted documents in different formats.

The application is publicly available at the address: http://e-quotes.net.

References

1. Bergler, S.: Semantic Dimensions in the Field of Reporting Verbs. In: The 9th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research, Oxford (1993)
2. Martin, J.R., White, P.R.R.: The Language of Evaluation: Appraisal in English. Palgrave Macmillan, London & New York (2005)
3. Wiebe, J.: Tracking point of view in narrative. Computational Linguistics 20 (1994)
4. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
5. Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Van Der Goot, E., Halkia, M., Pouliquen, B., Belyaeva, J.: Sentiment analysis in the news. In: 7th International Conference on Language Resources and Evaluation, pp. 2216–2220 (2010)
6. O'Keefe, T., Curran, J.R., Ashwell, P., Koprinska, I.: An Annotated Corpus of Quoted Opinions in News Articles. In: 51st Annual Meeting of the Association for Computational Linguistics (2013)
7. Lun-Wei, K., Chia-Ying, L., Hsin-Hsi, C.: Identification of Opinion Holders. International Journal of Computational Linguistics and Chinese Language Processing 14(4), 383–402 (2009)
8. Pareti, S.: The independent encoding of Attribution Relations. In: Eighth Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-8), Pisa (2012)
9. Elarnaoty, M., AbdelRahman, S., Fahmy, A.: A Machine Learning Approach For Opinion Holder Extraction Arabic Language. In: CoRR, abs/1206.1011 (2012)
10. Habash, N.: Introduction to Arabic Natural Language Processing. In: Hirst, G. (ed.) Synthesis Lectures on Human Language Technologies, 187p. Morgan Kaufmann (2010)
11. Abdul-Mageed, M., Kuebler, S., Diab, M.: SAMAR: A system for subjectivity and sentiment analysis of social media Arabic. In: 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), ICC Jeju, Republic of Korea (2012)
12. Balahur, A., Steinberger, R., Van der Goot, E., Pouliquen, B., Kabadjov, M.: Opinion mining on newspaper quotations. In: 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 523–526. IEEE (2009)

13. Desclés, J.-P., Guentchéva, Z.: Enonciateur, locuteur, médiateur. In: Monod Becquelin, A., Erikson, P. (eds.) Les Rituels du dialogue: Promenades ethnolinguistiques en terres amérindiennes, pp. 79–112. Société d'ethnologie (Recherches thématiques 6), Nanterre (2000)
14. Alrahabi M., Desclés, J.-P., Suh J.-Y.: Direct Reported Speech in multilingual texts: Automatic annotation and semantic categorization. In: FLAIRS 2010, Florida (2010)
15. Alrahabi, M., Desclés, J.-P.: EXCOM: Plate-forme d'annotation sémantique de textes multilingues. In: TALN 2009, Senlis (2009)
16. Desclés, J.-P.: Contextual Exploration Processing for Discourse Automatic Annotations of Texts. In: FLAIRS 2006, Florida (2006)
17. Dichy, J.: Morphosyntactic Specifiers to be associated to arabic lexical entries-Methodological and theorical aspects. In: ACIDCA 2000, Volume Corpora and Natural Language Processing, Monastir (2000)
18. Korayem, M., Crandall, D.J., Abdul-Mageed, M.: Subjectivity and Sentiment Analysis of Arabic: A Survey. In: Advanced Machine Learning Technologies and Applications, AMLTA (2012)
19. Pouliquen, B., Steinberger, R., Best, C.: Automatic Detection of Quotations in Multilingual News. In: The 6th International Conference on Natural Language Processing, GoTAL 2008, Gothenburg (2008)
20. Krestel, R., Bergler, S., Witte, R.: Minding the Source: Automatic Tagging of Reported Speech in Newspaper Articles. In: 6th International Language Resources and Evaluation (LREC 2008), Marrakech (2008)
21. De la Clergerie, E., Sagot, B., Stern, R., Denis, P., Recource, G., Mignot, V.: Extracting and visualizing quotations from news wires. In: L&TC 2009, Poznan (2008)
22. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: AAAI Spring Symposium on Exploring Attitude and Affect inText: Theories and Applications (2004)
23. Pareti, S., O'Keefe, T., Konstas, I., Curran, J.R., Koprinska, I.: Automatically Detecting and Attributing Indirect Quotations. In: Empirical Methods in Natural Language Processing (EMNLP), Seattle (2013)

Textual Entailment Using Different Similarity Metrics

Tanik Saikh[1], Sudip Kumar Naskar[2], Chandan Giri[1],
and Sivaji Bandyopadhyay[2]

[1] Indian Institute of Engineering Science and Technology, Shibpur, India
{tanik4u,chandan.giri}@gmail.com
[2] Jadavpur University, Kolkata, India
{sudip.naskar,sbandyopadhyay}@cse.jdvu.ac.in

Abstract. Textual entailment (TE) relation determines whether a text can be inferred from another. Given two texts, one is called the "Text" denoted as T and the other one is called "Hypothesis" denoted as H, the process of textual entailment is to decide whether or not the meaning of H can be logically inferred from the meaning of T. Different semantic, lexical and vector based similarity metrics are used as features for different machine learning classifiers to take the entailment decision in this study. We also considered two machine translation evaluation metrics, namely BLEU and METEOR, as similarity metrics for this task. We carried out the experiments on the datasets released in the shared tasks on textual entailment organized in RTE-1, RTE-2, and RTE-3. We experimented with different feature combinations. Best accuracies were obtained on different feature combinations by different classifiers. The best classification accuracies obtained by our system on the RTE-1, RTE-2 and RTE-3 dataset are 55.91%, 58.88% and 63.38% respectively. MT evaluation metrics based feature alone produced the best classification accuracies of 53.9%, 59.3%, and 62.8% on the RTE-1, RTE-2, and RTE-3 datasets respectively.

Keywords: Textual Entailment, Similarity metrics, Machine Translation Evaluation Metrics, Machine Learning.

1 Introduction

In Natural Languages Processing, Textual Entailment is a directional relation between two text fragments. The relation holds whenever the truth of one text fragment follows from another text. In the textual entailment framework, the entailing and entailed text is called Text (T) and Hypothesis (H) respectively. Textual entailment is not the same as pure logical entailment; it can be defined in a relaxed way "T entails H" (T => H) if, typically, a human reading T would infer that H is most likely true. The relation is directional because even if "T entails H", the reverse "H entails T" is much less certain.

In the present work, we make use of supervised learning techniques to take binary entailment decision. Weka[1] machine learning tool has been employed for this purpose. We make use of different semantic, lexical and vector based similarity metrics

[1] http://www.cs.waikato.ac.nz/ml/weka/

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 491–501, 2015.
DOI: 10.1007/978-3-319-18111-0_37

like cosine similarity, unigram match, jaccard similarity, dice coefficient, text overlap, harmonic mean as the machine learning features for different machine learning classifiers for this task. Besides using these similarity metrics which are typically used as features in textual entailment, we used two machine translation (MT) evaluation metrics - BLEU and METEOR, as similarity metrics. BLEU and METOR are two popular MT evaluation metrics which are used to evaluate the quality of machine-translated text. The MT evaluation metrics are applied on the machine-translated text and human generated reference translation(s) to find out how close the machine-translated text is to human translation(s). As far our knowledge goes, use of MT evaluation metrics as features for taking entailment decision by machine learning approach is new and first of its kind. We carried out experiments on the two MT evaluation metrics alone as features, as well as by combining them with other features. The experiment results reveal that MT evaluation metrics are effective features in taking entailment decision between a text and hypothesis pair.

2 Related Work

Survey shows that in RTE-1 the best result was obtained by Pérez and Alfonseca [9] using word the overlap method by BLEU algorithm. The output of BLEU was taken as confidence score and it was used to give TRUE or FALSE value to each entailment pair. They performed an optimization procedure for the development set that chose the best threshold according to the percentage of success of correctly recognized entailments and got a particular value, if the BLEU's output is higher than that threshold value then the entailment relation is TRUE for that T-H pair, otherwise the entailment relation is deemed FALSE. They obtained an accuracy of 70% on the RTE-1 dataset. In RTE-2, the best result was obtained by Hickl et al. [12], using lexical relation and syntactic matching, and the accuracy was 75%. The best result obtained on the RTE-3 dataset so far is 80% by Hickl and Bensley, [13] using discourse commitments, lexical alignment and knowledge extraction methods. Miguel et al. [11] used cosine similarity along with causal non-symmetric measure and obtained 63.5% accuracy in naïve Bayes on the RTE-3 dataset. Li et al. [5] used seven features namely lexical semantic similarity, named entities, dependent content word pairs, average distance, negation, task, and length and produced an accuracy of 62.7% on the RTE-3 dataset. Ferrés and Rodriguez [7] compute the similarity between two sentences in terms of the degree of overlapping between the semantic contents of the two sentences and obtained 61.5% accuracy on the RTE-3 dataset. Malakasiotis and Androutsopoulos [6] use support vector machine (SVM) technique to take entailment decision between each T–H pair and they achieved 61.75% accuracy on the RTE-3 dataset. In all the above cases conventional lexical and semantic features have been applied.

On the other hand, the objective of machine translation (MT) evaluation metrics is to measure how close the translation hypothesis is to a reference translation, the closer they are the better the translation hypothesis and hence the MT system. There exist several MT evaluation metrics like word error rate (WER) [14], position-independent WER (PER) [15], BLEU [1], NIST [16], Meteor [4], Translation error/edit rate (TER)

[17], General Text Matcher (GTM) [18], etc. Among these MT evaluation metrics BLEU is perhaps the most widely used MT evaluation metric among the MT researchers. Pérez and Alfonseca [9] demonstrate a comparative evaluation between this BLEU based algorithm and a Latent Semantic Analysis (LSA) based system for recognizing textual entailments. Volokh and Neumann [10] apply METEOR to T– H pairs assuming that they are two different translations of the same source sentence. However, no such notable works could be found in the study that make use of MT evaluation metric like BLEU, METEOR, etc. in machine learning platform to take entailment decision and to add these features with general conventional feature like cosine similarity, dice coefficient, etc. Our experiments use MT metrics as features along with conventional lexical and semantic features. Additionally, we carried out experiments by considering only the MT the MT evaluation metrics as features to compare the efficacy of the MT evaluation metrics as features for the textual entailment recognition task. The experiments reveal a new direction of research that MT metrics can take part to take entailment decision and it opens many new avenues in the research field.

3 Feature Analysis

Feature selection in machine learning approach is the most important part. Combining different sets of features which produce better result was our main motive in this experiment. The features which have been used in the experiment are described below.

3.1 Cosine Similarity

Cosine similarity is a vector based similarity measure. It is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The lower the angle between the two vectors the more similar the two vectors are.

3.2 Unigram Match with Respect to Text

Here we calculate the number of unigram match between T and H and subsequently normalize it by the number of unigrams in T.

3.3 Unigram Match with Respect to Hypothesis

Like the previous one here we calculate the number of unigram match between T and H, however, in this case we normalize it by the number of unigrams in H.

3.4 Jaccard Similarity

Jaccard similarity is defined as JS (A, B) = $|A \cap B|/|A \cup B|$, where A and B are two sets of elements. Jaccard similarity essentially tells how much the two sets have in common.

3.5 Dice Similarity

Dice similarity (or Dice coefficient) is a vector based similarity metric. It is defined as twice the number of terms common between the two entities divided by the total number of terms in the two entities. The value of 1 reveals that the vectors are identical whereas a 0 value signifies orthogonal vectors. Mathematically it can be defined as:

$$\text{Dice}(A, B) = 2(A \cap B)/ (|A|+|B|)$$

3.6 Overlap

This text similarity function is based on set. In this function a text is represented as a set where the set elements are the words. The similarity score varies between 0 to 1. It can be defined as:

$$\text{Overlap}(A, B) = |A \bigcap B|/\min (|A|, |B|).$$

3.7 Harmonic Mean

This is a text similarity function. Here also a text is represented as a set, where set elements are words. It can be defined as:

$$\text{Harmonic}(A, B) = (|A \bigcap B|. (|A|+|B|)) / (2.|A|.|B|)$$

3.8 Machine Translation Evaluation Metric

Machine translation (MT) evaluation metrics typically measures how close the translation output is to a human translation (or reference translation). The closer the translation hypothesis is to the reference translation, the better the translation system is. Over the years, MT researchers have proposed several MT evaluation metrics like word error rate (WER), position-independent word error rate (PER), BLEU, NIST, Meteor, Translation error/edit rate (TER), GTM, etc. Among these MT evaluation metrics BLEU is perhaps the most widely used among the MT researchers. In the present work we have considered two MT evaluation metrics, BLEU (Bilingual Evaluation Understudy) [1] and METEOR (Metric for Evaluation of Translation with Explicit Ordering) [2] as similarity measures to take the entailment decision by a classifier.

BLEU: BLEU [1] is an IBM-developed automatic metric and probably the best known and most adopted MT evaluation metric. The algorithm is based on n-gram precision. It compares n-grams of the candidate and n-grams of the reference translation and counts the number of n-gram matches. It has been found in the study that BLEU correlates very well with human judgements.

METEOR: It is an automatic metric for machine translation evaluation which is based on the notion of unigram matching between the candidate translation and human translation. Given a pair of sentences to be compared, Meteor computes the word

alignment between the two sentences, such that every word in each sentence maps to at most one word in the other sentence. This alignment is incrementally produced by a sequence of word-mapping modules. Additionally it does exact word match, stem matching and synonymy matching.

4 Experimental Setup and Results

The system of consists of several modules shown in figure 1.

4.1 Preprocessing Module

The system extracts a pair T and H from the development set. The datasets contains T-H pairs as given below.

```
<pair id="12" value="FALSE" task="IR">
        <t>Oracle had fought to keep the forms from being released. </t>
        <h>Oracle released a confidential document</h>
</pair>
```

From this XML data, we extract T and its corresponding H part and remove the stop words from both text and hypothesis.

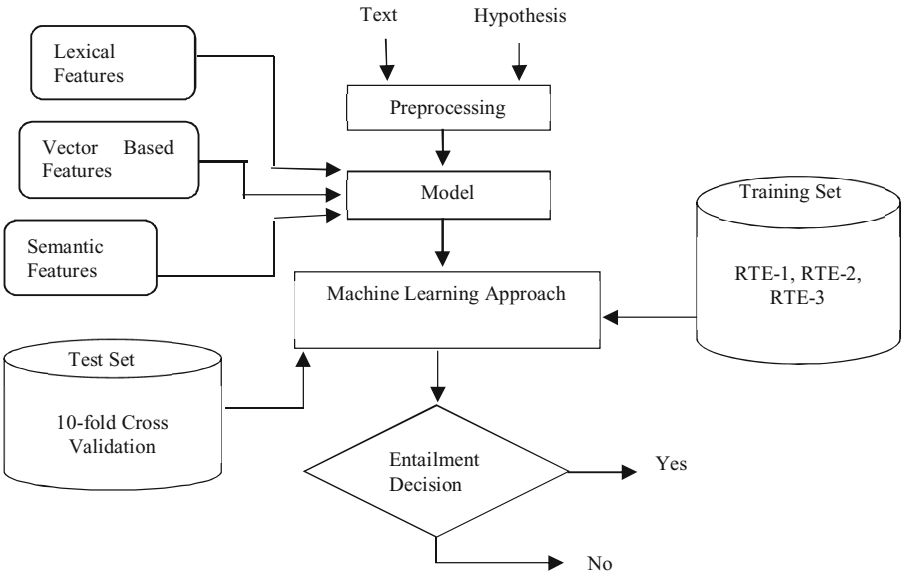


Fig. 1. System Architecture

4.2 Dataset

We carried out our experiments on the datasets released in the shared tasks on textual entailment organized in RTE-1, RTE-2 and RTE-3. Table 1 shows the statistics of the three datasets. Table1 provides the number of text-hypothesis pairs (THP), average text length (ATL) and average hypothesis length (AHL) for the development set and the testset belonging to each dataset. ATL and AHL provide average sentence length in words. However, we only made use of the development set which serves as both the training set as well as the testset in a 10-fold cross validation framework. Our model predicts textual entailment relation between a pair of text and we want to estimate how accurately our predictive model performs in practice. In prediction problem, a model is generally given a dataset of known data on which training is performed, and a dataset of unknown data against which a model is tested. The aim of cross validation is to define a dataset to test the model in the training phase, in order to limit problem like overfitting, give a demonstration on how the model will generalize to an independent dataset, etc. This was the reason we used the development set in a 10-fold cross validation framework instead of using a separate test set.

Table 1. The Statistics of the datasets

Dataset	Development Set			Test Set		
	THP	ATL	AHL	THP	ATL	AHL
RTE-1	567	23	9	800	25	10
RTE-2	800	26	9	800	27	8
RTE-3	800	34	8	800	29	7

4.3 System Description

We calculate several similarity scores for each T-H pair contained in the three datasets. Development set from each dataset has been used for this purpose. These scores are used as feature values to build a model. We combine the different features to build different models. The three models used in the experiments and the corresponding features are presented in table 1. The models are trained on different machine learning algorithms using weka machine learning tool. Based on 10-fold cross validation different classification accuracies have been achieved. The LibSVM, SMO, Naïve Bayes, AdaboostM1 and J48 machine learning algorithms in the Weka tool are employed for the experiments in our work.

Table 2. Different sets of features and the corresponding models

Feature Set	Model
CosineSimilarity, Unigram_Sim_Text, Unigram_Sim_Hypo, Jaccard_Similarity, Dice, Overlap, Harmonic	Model1
CosineSimilarity, Unigram_Sim_Text, Unigram_Sim_Hypo, Jaccard_Similarity, Dice, Overlap, Harmonic, Meteor	Model2
CosineSimilarity, Unigram_Sim_Text, Unigram_Sim_Hypo, Jaccard_Similarity, Dice, Overlap, Harmonic, Meteor, Bleu	Model3

4.4 Results and Discussions

We plotted the results of three different models on three different datasets using different machine learning approaches. In RTE-1 model1 gives the best result in AdaBoostM1, i.e., 58.2% which has been depicted in figure 2.

Results obtained on RTE-2 are depicted in figure3. Here model1 gives the highest score of 58.8%, in SMO.

Results obtained on RTE-3 datasets are presented in figure 4. Here we have achieved the highest score of 63.3% in Model2 and Model3 by SMO.

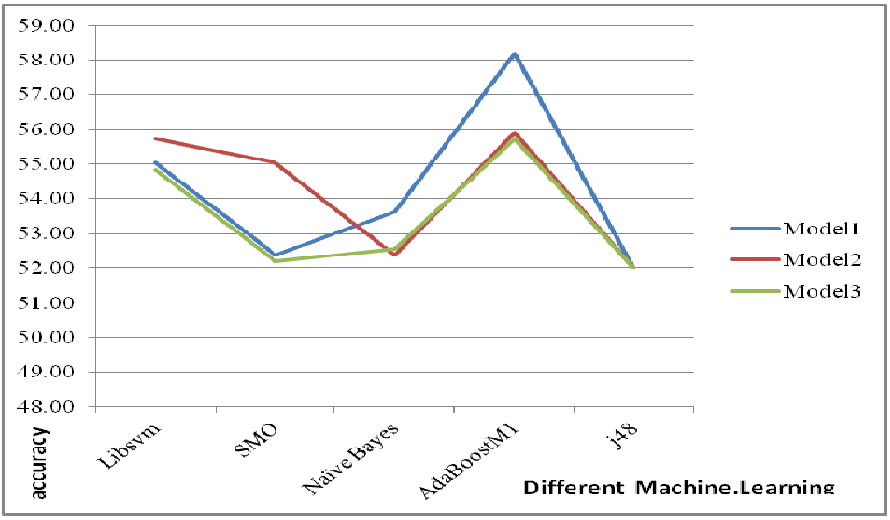


Fig. 2. Results on the RTE-1 dataset on different models using different machine learning approaches

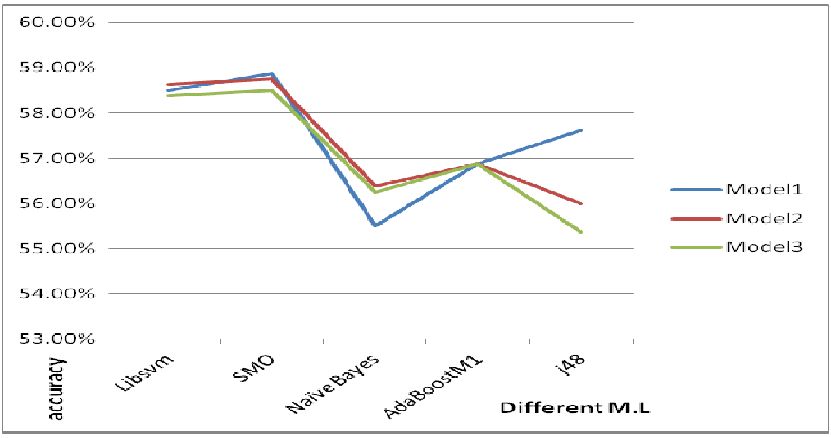


Fig. 3. Results on the RTE-2 dataset on different models using different machine learning approaches

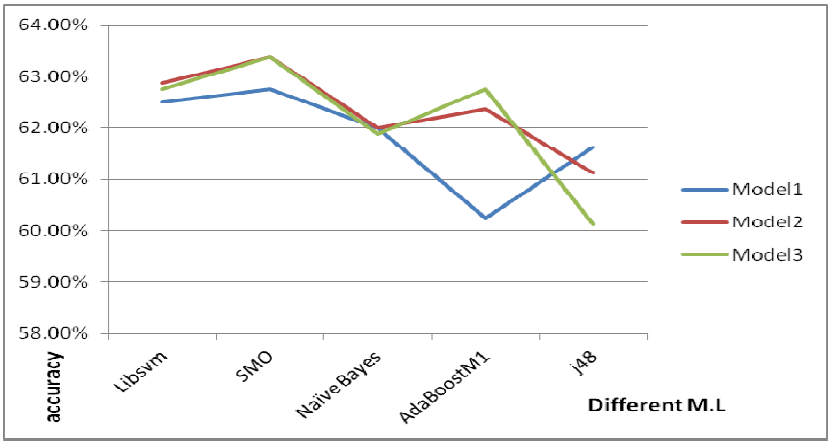


Fig. 4. Results on the RTE-3 dataset on different models using different machine learning approaches

4.5 Results Using MT Metrics

We carried out another set of experiments taking only BLEU, METEOR as features and using the same experimental setup mentioned above. The results have been shown in figure 5. We observed that theses MT metrics are useful features in taking entailment decision in machine learning platform. The combined results on the three datasets on the same features are shown in figure 5.

In RTE-1 LibSVM gives the best classification accuracy of 53.9%, in RTE-2 RBFNetwork gives the highest score of 59.3% where as in RTE-3 RBFNetwork also gives the best result of 62.8%.

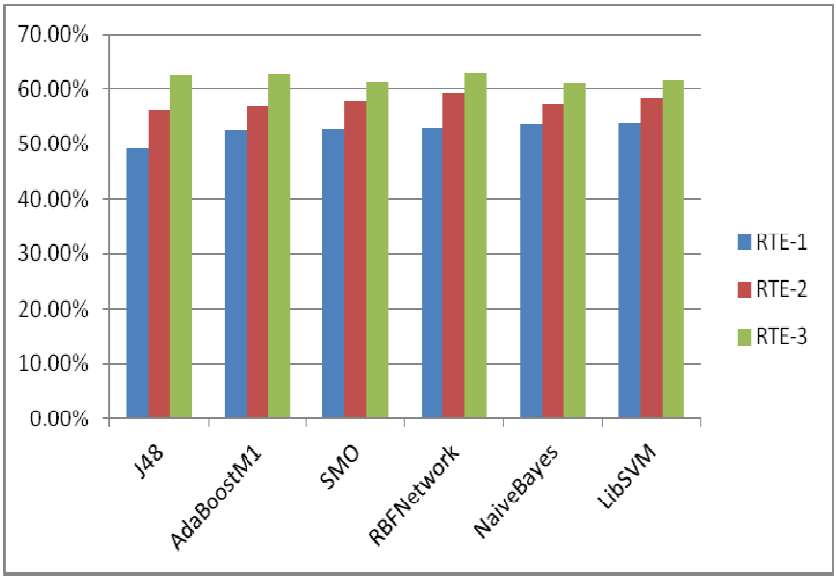


Fig. 5. Results on three different datasets on models built on MT evaluation metrics using different machine learning approaches

By comparing the results presented in figures 1–4, it can be noticed that the models built using only the MT evaluation metrics perform almost at par with the models built on other traditionally used features. In fact, the MT evaluation metrics based TE model provides the best result for the RTE-2 dataset.

5 Future Work and Conclusion

The above sets of experiments conclude that MT evaluation metric as features in machine learning platform can predict the entailment relation between a pair of T-H. Conventional similarity metrics like cosine similarity, Jaccard similarity, Disc coefficient, etc. are also very good predictor of entailment relation. Since MT metrics can contribute in taking entailment decision, this opens up new avenues for studying textual entailment.

In future we plan to experiment more with other similarity metrics with the same dataset and also datasets. We have a plan to use other MT evaluation metrics such as GTM and CDER as features for our experiment.

Acknowledgements. The research leading to these results has received funding from the project "Development of English to Indian Languages Machine Translation Systems (E-ILMT) - Phase II" funded by The Department of Electronics and Information Technology (DeitY), Ministry of Communications and Information Technology, Government of India.

References

1. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, pp. 311–318 (2002)
2. Banerjee, S., Lavie, A.: 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, Ann Arbor, Michigan, pp. 65–72 (2005)
3. Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. In: Monz, C., de Rijke, M. (eds.) PASCAL Workshop on Text Understanding and Mining 2001. Light-Weight Entailment Checking for Computational Semantic. Proceedings ICoS-3 (2004)
4. Lavie, A., Agarwal, A.: METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In: Proceedings of the Second ACL Workshop on Statistical Machine Translation, Prague, Czech Republic, pp. 228–231 (2007)
5. Li, B., Irwin, J., Garcia, E.V., Ram, A.: Machine Learning Based Semantic Inference: Experiments and Observations at RTE-3. In: Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic, pp. 159–164 (2007)
6. Malakasiotis, P., Androutsopoulos, I.: Learning Textual Entailment using SVMs and String Similarity Measures. In: Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic, pp. 42–47 (2007)
7. Ferrés, D., Rodríguez, H.: Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In: Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic, pp. 60–65 (2007)
8. Pakray, P., Bandyopadhyay, S., Gelbukh, A.: Binary-class and Multiclass based Textual Entailment System. In: Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, Japan, National Institute of Informatics (2013)
9. Perez, D., Alfonseca, E.: Application of the Bleu algorithm for recognising textual entailments. In: Proceedings of the First Challenge Workshop Recognising Textual Entailment, pp. 9–12 (2005)
10. Volokh, A., Neumann, G.: Using MT-based metrics for RTE. In: The Fourth Text Analysis Conference. NIST (2011)
11. Ríos Gaona, M.A., Gelbukh, A., Bandyopadhyay, S.: Recognizing textual entailment using a machine learning approach. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part II. LNCS, vol. 6438, pp. 177–185. Springer, Heidelberg (2010)
12. Hickl, et al.: Recognizing textual entailment with LCC's GROUNDHOG system. In: Proceedings of the Second PASCAL Challenges Workshop (2006)
13. Hickl, A., Bensley, J.: A discourse commitment-based framework for recognizing textual entailment. In: Association for Computational Linguistics, pp. 171–176 (2007)
14. Vidal, E.: Finite-State Speech-to-Speech Translation. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 111–114 (1997)
15. Tillmann, C., Vogel, S., Ney, H., Sawaf, H., Zubiaga, A.: Accelerated DP based Search for Statistical Translation. In: Proceedings of the 5th European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 2667–2670 (1997)
16. Doddington, G.: Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)

17. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp. 223–231 (2006)
18. Turian, J.P., Shen, L., Dan Melamed, I.: Evaluation of Machine Translation and Its Evaluation. In: Proceedings of MT Summit 2003, New Orleans, Luisiana, pp. 386–393 (2003)

Machine Translation and Multilingualism

Translation Induction on Indian Language Corpora Using Translingual Themes from Other Languages

Goutham Tholpadi, Chiranjib Bhattacharyya, and Shirish Shevade

Computer Science and Automation
Indian Institute of Science
Bangalore 560012 India
{gtholpadi,chiru,shirish}@csa.iisc.ernet.in

Abstract. Identifying translations from comparable corpora is a well-known problem with several applications, e.g. dictionary creation in resource-scarce languages. Scarcity of high quality corpora, especially in Indian languages, makes this problem hard, e.g. state-of-the-art techniques achieve a mean reciprocal rank (MRR) of 0.66 for English-Italian, and a mere 0.187 for Telugu-Kannada. There exist comparable corpora in many Indian languages with other "auxiliary" languages. We observe that translations have many topically related words in common in the auxiliary language. To model this, we define the notion of a *translingual theme*, a set of topically related words from auxiliary language corpora, and present a probabilistic framework for translation induction. Extensive experiments on 35 comparable corpora using English and French as auxiliary languages show that this approach can yield dramatic improvements in performance (e.g. MRR improves by 124% to 0.419 for Telugu-Kannada). A user study on *WikiTSu*, a system for cross-lingual Wikipedia title suggestion that uses our approach, shows a 20% improvement in the quality of titles suggested.

1 Introduction

The task of identifying translations for terms is usually posed as one of generating translation correspondences. A translation correspondence for a source word assigns a score to every target word proportional to its topical similarity to the source word, so that the translation is assigned the highest score. Translation correspondences are key inputs for building human readable dictionaries, as well as for many language processing systems, including machine translation and cross language information retrieval [1].

Comparable corpora-based[1] translation correspondence induction (CC-TCI) is a popular approach for obtaining translation correspondences. Most methods using this approach require dictionaries and parsers, or make assumptions about

[1] "Comparable corpora" are document-aligned multilingual corpora, where the aligned documents are in different languages and "talk about the same thing" [2].

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 505–519, 2015.
DOI: 10.1007/978-3-319-18111-0_38

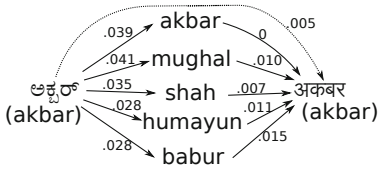


Fig. 1. A subset of the *translingual theme* in English (words in center) for a Kannada (left)–Marathi (right) translation pair. The arrow from w_1 to w_2 is labeled with the probability $P_{\mathrm{CC}}(w_2|w_1)$ (see Section 3.2).

properties of the languages involved (see Section 2). However, for many language pairs such as in Indian languages, the CC-TCI problem poses several challenges:

- Resources such as seed bilingual lexicons and linguistic tools (POS taggers, morpho-syntactic analyzers, etc.) required by some methods (e.g. [3], [4]) are not be available.
- Language properties such as presence of cognates, and orthographic similarity, cannot be assumed in general, ruling out some methods (e.g. [5], [6]).
- The only available cross-language resource is a comparable corpus. However, even this is relatively small for most language pairs, so that "CC-only" methods (e.g. [7], [8]) do not perform well.

We observe that source and target translations have many topically related words in common in other "auxiliary" language corpora[2], which can be a useful cue for identifying translations. To model this, we define the notion of a *translingual theme* (for a source–target word pair) as a set of words derived from auxiliary language comparable corpora that statistically co-occur with the source and target words. For example, Figure 1 shows the source–target pair ಅಕ್ಬರ್ /akbar/ and अकबर /akbar/ (both referring to the proper noun "Akbar"[3]) from a Kannada–Marathi corpus, and a subset {'mughal', 'shah', 'humayun', 'babur'}[4] of its translingual theme derived from Kannada–English and Marathi–English auxiliary corpora.

In this work, we investigate the utility of *auxiliary* language corpora for boosting CC-TCI performance. For this purpose, we leverage Wikipedia, a large web-based multilingual encyclopedia with more than 26 million articles in 285 languages. In Wikipedia, articles in different languages on the same topic are linked (by "`langlink`"s), which enables us to quickly construct corpora for a large number of language pairs.

Cross-lingual Wikipedia Title Suggestion. The proportion of content in Wikipedia in different languages varies widely [9], and the topics covered also

[2] Comparable corpora where one language is from the pair under consideration, and the other can be any other (auxiliary) language.

[3] Akbar was a king from the Mughal dynasty who ruled parts of North India in the 16^{th} century A.D.

[4] Shah is a royal title; Humayun and Babur were both Mughal kings.

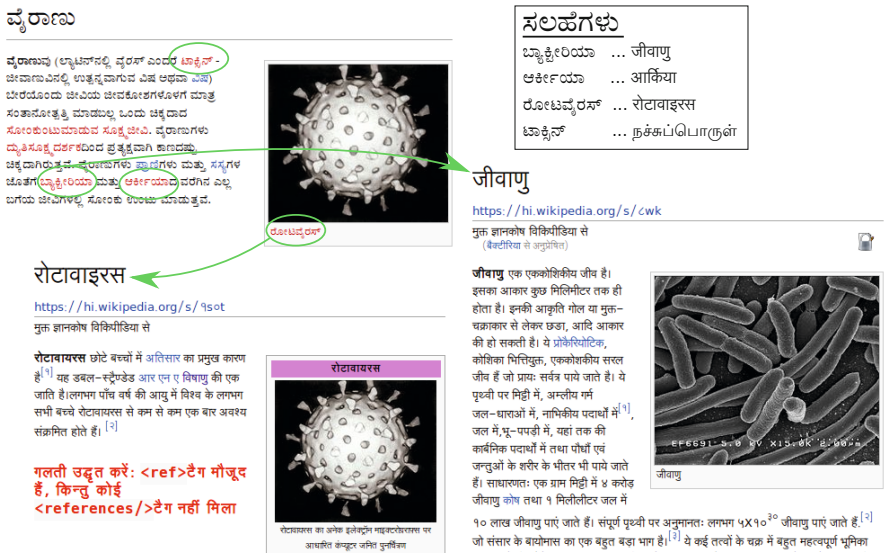


Fig. 2. A multilingual user reading a Kannada article on ವೈರಾಣು (“virus”) (top-left) finds the words ಟಾಕ್ಸಿನ್ (“toxin”), ಬ್ಯಾಕ್ಟೀರಿಯಾ (“bacteria”), ಆರ್ಕೀಯಾ (“Archaea”) and ರೋಟವೈರಸ್ (“rotavirus”) interesting, but there are no Kannada articles for these concepts. In response, the system gives Wikipedia title suggestions (box at top-right) from Hindi and Tamil (जीवाणु (“bacteria”), and so on).

vary with language. If a Wikipedia concept has no article in one language, articles in other languages might be suggested to a multilingual user. For example (see Figure 1), an Indian user browsing the Kannada article ವೈರಾಣು /ʋaiɾaːɳu/ (‘virus’) might want to know about ಬ್ಯಾಕ್ಟೀರಿಯಾ /bjaːkʈiːrijaː/ (‘bacteria’), and ರೋಟವೈರಸ್ /roːʈaʋaiɾas/ (‘rotavirus’). There are no articles for these concepts in Kannada, but there *are* articles in Hindi, viz. जीवाणु /ʤiːʋaːɳu (‘bacteria’) and रोटावाइरस /roːʈaːʋaːiras (‘rotavirus’). These titles can be suggested to the user (the box at top-right in the figure) for further reading. Recently, [10] attempted a similar task using `langlinks`, where the setting was restricted to source words that are Wikipedia titles. The task of suggesting target-language Wikipedia titles for *source words that are not Wikipedia articles* has not been attempted before. In the absence of `langlinks`, this task is difficult to solve, especially for under-resourced languages without machine translation (MT), dictionaries, parsers, and parallel corpora. In this resource-scarce setting, we attempted the title suggestion task using a CC-TCI approach, leveraging auxiliary language corpora from Wikipedia. The resulting system *WikiTSu* can work for any Wikipedia language pair, and uses a Wikipedia corpus as the *only* resource.

Contributions. Our main contributions are:
• We define a new probabilistic notion of cross-language similarity in the context of comparable corpora. We show how this notion naturally admits auxiliary language corpora under certain assumptions. We also show how to combine similarities from multiple auxiliary languages using a simple mixture model, and use the combined score for translation correspondence induction. (Section 3.1)
• We perform extensive experiments on 35 comparable corpora in 9 languages from 4 language families (Indo-Aryan, Dravidian, Germanic, Romance) extracted from Wikipedia, and show significant boosts (upto 124%) in performance for a state-of-the-art CC-TCI method. (Section 4.2)
• To address the cross-lingual Wikipedia title suggestion task for the difficult resource-scarce setting, we built a system *WikiTSu* that works for *any* language pair in Wikipedia, using *no other resources*. We show via a user study that *WikiTSu* does significantly better than a state-of-the-art baseline. (Section 4.4)
• We are releasing translation correspondences for 42 language pairs (nearly 5000 words per language, 10 candidates per word) for public use as probabilistic dictionaries, or as inputs to annotator tools for dictionary building. As of today, there exist *no* dictionaries for most of these language pairs.
• We are making publicly available[5] a large curated collection of comparable corpora and gold standard translation pair sets in 7 under-resourced languages. We are also releasing the code for *WiCCX*, an in-house tool for generating preprocessed and algorithm-ready comparable corpora from Wikipedia dumps.

2 Related Work

Translation Correspondence Induction Using Comparable Corpora. The problem of inducing translation correspondences from bilingual comparable corpora was introduced by [11]. There have been several approaches to this task, differentiated by the resource assumptions made.

Knowledge-based Approaches. Many approaches to translation correspondence induction use seed lexicons [2][4][12,13,14,15], syntactic/morphological analyzers [16,17,18,19], parallel corpora, translation/transliteration models [20], and other resources [3,21]. Other approaches make assumptions about the languages or corpora, such as syntactic structure, orthographic similarities, presence of cognates, monogenetic relationships, domain-specific content [5,6][22,23,24,25,26]. [27] and [28] use existing dictionaries to induce translation correspondences. There is also work on comparable corpora-based named entity mining [29,30,31] which has a similar setting, but addresses a different problem. [9] use canonical correlation analysis for Wikipedia name search, and [32] use Wikipedia link structure for translation correspondence induction. These are complementary to our statistical approach, and they can be combined to improve performance.

[5] `http://www.cicling.org/2015/data/31`

Comparable Corpora-only Approaches. [7] and [33] proposed methods that use only comparable corpora and were applied to relatively high quality corpora. The most recent work using only comparable corpora is by [8] and [34] who use latent space models, and demonstrate good performance on Wikipedia data.

Improving CC-TCI. There have been efforts to improve the results from existing methods by pre- or post-processing. [35] and [36] attempt to improve corpus quality before doing translation correspondence induction. [37] take a noisy translation correspondence obtained from any method and incorporates knowledge from *monolingual corpora in the languages of the pair* to improve accuracy. Our method, on the other hand, takes a noisy translation correspondence and incorporates knowledge from *comparable corpora in auxiliary languages* to improve accuracy. These approaches are complementary to our approach, and they can be combined to improve accuracy further.

Combination Approaches. [16] represent different kinds of relationships between words on a graph and use SimRank [38] to compute a combined score. [21] combine information with a mixture model similar to ours, while [25] use a voting scheme instead.

Using Auxiliary Languages. [39] attempted to use auxiliary languages for translation correspondence induction, but using parallel corpora. [40], [1], [27], and [41] use existing dictionaries or monogenetic relationships, while we work in the comparable corpora-only setting and make no assumptions about the language family. Auxiliary language approaches have also been used for other problems, e.g. *triangulation* for machine translation [42,43,44], word alignment [45], transliteration [46], and paraphrase extraction [47].

3 Problem Formulation and Approach

3.1 Problem Definition

Let L_S and L_T denote the source and target languages, with vocabularies V_S and V_T respectively. The translation correspondence for $s \in V_S$ is the set $TC(s) = \{(t, r_{st})\}_{t \in V_T}$ where $r_{st} \in [0, \infty)$ is the topical similarity of t to s. A translation correspondence can be viewed as being generated from a scoring function $S_{raw}()$ such that $S_{raw}(t|s) = r_{st}$. Given a comparable corpus, any method in Section 2 can be used to learn the scoring function $S_{raw}(t|s)$.[6] This function induces a ranking over the words in V_T for each word s in V_S. We assume that there exists an auxiliary language L_A which has comparable corpora with L_S and L_T, so that we can learn scoring functions $S_{raw}(a|s)$, $S_{raw}(s|a)$, $S_{raw}(t|a)$ and $S_{raw}(a|t)$, analogous to $S_{raw}(t|s)$.

The objective is to compute a scoring function $S_A(t|s)$ that uses the S_{raw} scoring functions and gives a better ranking over V_T for each s.

[6] We use the method by [8] to obtain S_{raw}, and also as the baseline (Section 4.1).

3.2 Incorporating Information from an Auxiliary Language

Cross-language Similarity in Terms of a Comparable Corpus. A document-aligned multilingual comparable corpus in l languages can be viewed as a set of tuples (each tuple contains l documents, one per language). Consider a random experiment where we sample a word from one of the documents of such a tuple. Define the random variables: $S \triangleq$ the word sampled from the L_S–document in the tuple; $T \triangleq$ the word sampled from the L_T–document in the tuple. Let $P_{\mathrm{CC}}(T = t|S = s)$ be the probability that the sampled L_T–word is t given that a sampled L_S–word is s. This probability will be high for values of t (i.e. L_T–words) that are topically related to s. For example, given that we sampled ಬ್ಯಾಕ್ಟೀರಿಯಾ /bjaːkʈiːrijaː ('bacteria') from the L_S–document, we are very likely to sample words like जीवाणु /dʒiːʋaːɳu ('bacteria') or रोग /roːg ('disease') from the L_T–document.[7] This is similar in spirit to the idea of *lexical triggers* [48]. We can use a baseline scoring function S_{raw} (as defined in Section 3.1) and define the *trigger probability* $P_{\mathrm{CC}}(t|s) \triangleq \frac{S_{raw}(t,s)}{\sum_{t'} S_{raw}(t',s)}$.[8] This models topical relatedness in the context of comparable corpora in a probabilistic setting.[9] Since this model is asymmetric, i.e. in general $P_{\mathrm{CC}}(t|s) \neq P_{\mathrm{CC}}(s|t)$, we can expect that the translation induction performance depends on the choice of the source language, and this is confirmed by our experiments (Section 4.2).

Translingual Themes. Define the random variable $A \triangleq$ the word sampled from the L_A document in the tuple. Similar to $P_{\mathrm{CC}}(t|s)$, we get $P_{\mathrm{CC}}(t|a)$ and $P_{\mathrm{CC}}(a|s), \forall a \in V_A$. We define the *source theme* for s as the set $ST_A(s) \subset V_A$ that satisfies $\forall a \in ST_A(s), a' \in V_A \setminus ST_A(s), P_{\mathrm{CC}}(a|s) \geq P_{\mathrm{CC}}(a'|s)$, and $\sum_{a \in ST_A(s)} P_{\mathrm{CC}}(a|s) < \tau$, where $\tau < 1$ is threshold determined empirically. The source theme is a set of L_A words that have the highest trigger probability given the source word s. We define the *target theme* for t as the set $TT_A(t) = \{a|t \in ST_T(a)\}$, i.e. the target theme is the set of L_A words for which the target word t has a high trigger probability. Finally, we define the *translingual theme* for the ordered pair (t, s) as $TLT_A(t, s) = ST_A(s) \cap TT_A(t)$.

Using Translingual Themes to Compute Word Similarity. Our probabilistic definition allows us to write $P_{\mathrm{CC}}(t|s) = \sum_{a \in V_A} P_{\mathrm{CC}}(t|a, s) P_{\mathrm{CC}}(a|s)$. Using the entire vocabulary V_A introduces a lot of noise [7]. Instead, we use the *translingual theme*, which is a more focused and reliable indicator of topical relatedness. In addition, if we assume that T is independent of S given A, we get $P_A(t|s) \triangleq \sum_{a \in TLT_A(t,s)} P_{\mathrm{CC}}(t|a) P_{\mathrm{CC}}(a|s)$. [10] The independence assumption means that we are no longer constrained to use a multilingual corpus, but

[7] Here, L_S=Kannada and L_T=Hindi.

[8] We abbreviate $P_{\mathrm{CC}}(T = t|S = s)$ to $P_{\mathrm{CC}}(t|s)$.

[9] This is different from $P_{\mathrm{MT}}(t|s)$, the probability that a translator would consider that t is a translation of s, which is usually used in machine translation literature [49].

[10] While this equation looks identical to the triangulation equation [43], the underlying probabilistic model there is $P_{\mathrm{MT}}()$ (see Footnote 9), while in our case it is $P_{\mathrm{CC}}()$.

can use several bilingual corpora—one for each language pair. This is critical, since multilingual corpora are far more difficult to obtain than bilingual corpora. Also, if the word a is not present in the L_A–L_T corpus, we need to use a non-informative uniform back-off distribution for $P(t|a)$ (as suggested by [43] for dissimilar corpora).

We use $P_A(t|s)$ as a measure of the topical similarity between t and s. In the example in Figure 1, using $P_{\{\text{en}\}}(t|s)$ along with $P_{\text{CC}}(t|s)$ results in a high value for $S_{\{\text{en}\}}$ (अकबर|ಅಕ್ಬರ್), and thus improves the ranking of the translation अकबर from 6 (using S_{raw}) to 3 (using $S_{\{\text{en}\}}$).

3.3 Model for Combining Languages

Since both $P_{\text{CC}}(t|s)$ and $P_A(t|s)$ are imperfect indicators of translation correspondence, we would like to combine both scores, but weight the contribution of each distribution according to its performance on a small training set. Consequently, we chose a simple mixture model for combining information. The generative story for the model is as follows:

1. Sample a source word s uniformly from the source vocabulary V_S.
2. For each s:
   (a) Sample $j \sim Discrete(\lambda)$. (j is one of the mixture components.)
   (b) Sample $t \sim Discrete(\beta_{js})$. (A mixture component is a discrete distribution over the target vocabulary.)

Suppose we have learned, using a set of comparable corpora, the distributions $P_0(t|s) \triangleq P_{\text{CC}}(t|s)$ and $P_j(t|s) \triangleq P_{A_j}(t|s),\ j = 1 \ldots J$, for the auxiliary language set $A = \{A_j\}_{j=1}^{J}$.[11] Define

$$p(t|s,\lambda) \triangleq \sum_{j=0}^{J} \lambda_j \beta_{jst}$$

where $\beta_{jst} = P_j(t|s)$, $\lambda_j \geq 0\ \forall j$ and $\sum_j \lambda_j = 1$. Given a small training set of source-target translation pairs $\{(s_n, t_n)\}_{n=1}^{N}$[12], we can learn λ by grid search, or by maximizing the log-likelihood $\sum_n \log \sum_j \lambda_j \beta_{js_nt_n}$ w.r.t. λ.[13] For the maximum likelihood approach, we used the EM algorithm. We initialize λ randomly, and then use the following updates till convergence:

$$b_{nj} = \frac{\beta_{js_nt_n}\lambda_j}{\sum_{j'} \beta_{j's_nt_n}\lambda_{j'}}\ ,\ \lambda_j = \frac{\sum_n b_{nj}}{\sum_{j'}\sum_n b_{nj'}}\ .$$

[11] In our experiments, we have tried $J = 1, 2$ and 3.

[12] Note that this training set of a few (< 100) translation pairs is different from the seed lexicons mentioned in Section 1, which are bilingual lexicons of a few thousand translation pairs that are used by some methods (e.g. [4]) to bootstrap cross-language comparisons. We do not use such seed lexicons.

[13] We report results using the grid search in the paper, and the results using EM in the supplementary material.

We do multiple random initializations, and keep the λ with the best likelihood. Having learnt λ, we can compute $p(t|s,\lambda)$ for any word pair (s,t). The **new scoring function** $S_A()$ is defined as $S_A(t|s) \triangleq p(t|s,\lambda) = \sum_{j=1}^{J} \beta_{jst}\lambda_j$. The translation candidate t^* for s is defined as $t^* = \arg\max_t S_A(t|s)$.

Through β, other cues can also be introduced, e.g., other scoring functions on the same corpus, limited-coverage dictionaries, and multilingual WordNets.

4 Experiments and Results

We evaluated our method on 21 language pairs derived from 7 Indian languages from 2 language families—Indo-Aryan: Bengali (*bn*), Hindi (*hi*), and Marathi (*mr*), and Dravidian: Kannada (*kn*), Malayalam (*ml*), Tamil (*ta*), and Telugu (*te*). We used two auxiliary languages from different language families—Germanic: English (*en*), and Romance: French (*fr*). We extracted 35 comparable corpora (624,856 documents in total) from Wikipedia, which were the largest possible corpora possible (using all available `langlinks`). We used a state-of-the-art method for CC-TCI to measure the impact of using auxiliary languages. We also performed a user study on *WikiTSu* for the language pair Kannada-Hindi. In the remainder of this section, we refer to our method as AUX-COMB.

4.1 Experimental Setup

Corpora and Gold Standard Sets. We downloaded the Wikipedia XML dumps[14] for the 9 languages and processed them using *WiCCX*, a tool that extracts comparable corpora, cleans the documents, and restricts them to a "useful" subset of the vocabulary. The *WiCCX* tool also extracts translation pairs using `langlinks` between article titles—an approach discussed in earlier work [32]. We also create reduced gold sets for each auxiliary language set by removing words that are not present in the auxiliary corpora. Thus we obtained several gold sets $G(A)$ depending on the choice of the auxiliary language set A. The details of the corpora and gold sets are given in the supplementary material.

Evaluation Procedure. We used Monte Carlo cross-validation, which has been shown to be asymptotically consistent [50] resulting in more pessimistic predictions of performance on test data compared to normal cross-validation. The gold-standard translation pair set was divided into training and test sets in k different ways by random sampling[15]. The size of the training set (for learning λ) was fixed at d[16] for all language pairs, and the remaining translation pairs were used for testing.

Given a test set in languages L_1 and L_2, for each word in L_1 in the test set, each method was used to generate a ranked list of candidate words in language

[14] `http://dumps.wikimedia.org/`

[15] We fixed $k = 10$ in our experiments.

[16] We set d=40 for A={en}, {fr} and {hi}, and d=35 for A={en, fr} (proportional to the size of the gold standard set $G(A)$ available).

L_2. Similarly, L_1 candidates were generated for L_2 words. Each ranked list was evaluated in terms of mean reciprocal rank (MRR) [51].[17] Let $tr(w)$ be the translation of w in the gold set. Given a ranked list generated for w, $RR(w) = \frac{1}{\text{Rank of } tr(w) \text{ in the list}}$. The reciprocal ranks were averaged over all words in the test set, and again averaged over all k folds in the Monte Carlo cross-validation to get the final score. Since the gold sets differed between experiments, the scores are not directly comparable. Instead, we report performance improvement over the baseline score (computed on the same gold set).[18]

Scoring Function and Baseline. Given the noisy nature of the Wikipedia corpus, we chose the **TI+Cue** method as our baseline. The TI+Cue method is a state-of-the-art method for CC-TCI, proposed in [8]. It is based on topic models [52], which work at the coarser level of topics (rather than words, or documents), and hence can be expected to smooth out noise better.[19] This method also yielded the scoring function S_{raw} (see Section 3.1) used by AUX-COMB.

For bilingual topic modeling, we used the Mallet toolbox [53] with the following configuration: regex for importing data = "`[\p{L}\p{M}]+`" (to read Unicode text with tokenization on whitespace and punctuation), Number of topics $K = \lceil \frac{\#\text{doc pairs}}{10} \rceil$, $\alpha = \frac{50}{K}$, $\beta = 0.01$ (to favor peaked distributions for topics and words [54]), Number of iterations = 1000 for estimation and 100 for inference, and Burn-in period = 100 iterations (the default settings in the toolbox).

4.2 Discussion of Results

The performance of the baseline method for $G(\{\text{en}\})$ is shown in Table 1 (left). The number in row L_S and column L_T is the performance measured when identifying translations for L_S words in language L_T. It can be seen that MRR is in the range [0.2,0.3] for most language pairs, and even lower for *bn-kn*, *kn-ml*, *kn-mr* and *ml-mr*, which have small corpora sizes (<1000). We believe that using auxiliary language corpora will be especially useful for such language pairs.

Auxiliary Languages Boost Performance. Table 1 (right) shows the improvement in MRR for AUX-COMB with English as the auxiliary language[20]. We see reasonable improvement in MRR in general, with large improvements (upto **91%**) for some language pairs. We see similar behavior with French and Hindi as the auxiliary language (Table 2). To show the contribution of the auxiliary language model, we shade each cell in Table 1 (right) proportional to $\lambda_{\{\mathbf{en}\}}$, the

[17] We also measured "Presence-at-k" (Pres@k) for k = 1 and 5. These measures showed the same trends as MRR. The details are given in the supplementary material.

[18] We report the absolute scores for the baseline on $G(\{\text{en}\})$ in Table 1 (left) to give the reader an idea of the absolute MRR scores. The absolute scores for all cases are reported in the supplementary notes.

[19] The baseline method is described in detail in the supplementary notes.

[20] We report the mean MRR across samples, and omit variances due to lack of space (e.g. the average variance was .04 for $S_{\{\text{en}\}}()$).

Table 1. *Left*: Absolute performance (in terms of MRR) of the baseline method (TI+Cue) on the English gold set $G(\{\text{en}\})$. (Poorly performing language pairs are in bold). *Right*: Percentage improvement (over baseline MRR) of AUX-COMB using $S_{\{\text{en}\}}()$. (The shading darkness of a cell is proportional to $\lambda_{\{\text{en}\}}$.).

MRR	bn	hi	kn	ml	mr	ta	te
bn	–	.3174	**.1842**	.2422	.2439	.2923	.2271
hi	.284	–	.2837	.2408	.3145	.283	.2942
kn	.2113	.2966	–	**.1273**	**.165**	.2342	.2313
ml	.2500	.3228	**.1522**	–	.2226	.2416	.2381
mr	.2230	.349	**.1403**	**.1876**	–	.2832	.2488
ta	.2731	.3232	.241	.2472	.2511	–	.2483
te	.2506	.2943	**.1748**	.3543	.2318	.2571	–

%Imp	bn	hi	kn	ml	mr	ta	te
bn	–	24.95	**90.34**	20.81	10.46	28.16	38.00
hi	7.89	–	5.71	24.09	25.02	25.97	26.14
kn	55.04	26.50	–	**91.83**	**58.55**	50.21	65.93
ml	12.32	19.08	**37.45**	–	17.74	3.93	36.67
mr	21.17	29.46	**65.93**	**39.71**	–	14.05	23.59
ta	8.46	9.41	9.67	4.81	7.81	–	21.35
te	29.49	36.94	**81.69**	19.53	33.91	42.98	–

Table 2. Percentage improvement (over baseline MRR) of AUX-COMB using $S_{\{\text{fr}\}}()$ on $G(\{\text{fr}\})$ (left), and $S_{\{\text{hi}\}}()$ on $G(\{\text{hi}\})$(right)

%Imp	bn	hi	kn	ml	mr	ta	te
bn	–	32.50	**60.04**	34.47	23.59	24.13	27.52
hi	21.37	–	22.92	31.38	8.63	18.50	19.71
kn	43.11	19.85	–	**70.15**	**32.83**	51.69	44.58
ml	22.28	16.10	**55.33**	–	11.07	28.29	43.32
mr	33.30	26.59	**49.07**	**22.73**	–	10.22	36.64
ta	33.63	11.59	24.97	21.37	7.51	–	18.22
te	20.44	18.15	**59.24**	0.74	24.32	36.07	–

%Imp	bn	hi	kn	ml	mr	ta	te
bn	–	–	**61.78**	26.08	23.37	23.79	25.32
hi	–	–	–	–	–	–	–
kn	29.79	–	–	**34.68**	**18.85**	40.08	54.47
ml	12.22	–	**72.33**	–	25.58	44.09	33.28
mr	15.15	–	**71.11**	**33.61**	–	24.24	37.81
ta	19.71	–	24.14	13.63	19.46	–	34.99
te	20.71	–	**76.78**	19.59	53.45	54.93	–

component of λ corresponding to $P_{\{\text{en}\}}$. The minimum and maximum values of $\lambda_{\{\text{en}\}}$ were 0.51 and 0.81, and the mean and median values were both 0.65.

We tried AUX-COMB with two[21] auxiliary languages to study the impact of using more languages (Table 3). The results are much better than when a single auxiliary language is used (we see upto **124%** improvement). For example, for *mr*-*ml*, the improvement obtained using *en* and *fr* were 39% and 22%, and using both was 83%. We see similar results for *kn*-*te*, *te*-*mr*, etc. We see robust performance for most of the 21 language pairs and for both directions.

Asymmetric Performance. As anticipated in Section 3.2, we see an asymmetry in performance for a single language pair, e.g. MRR for *te*–*ml* is 0.3543, while MRR for *ml*–*te* is 0.2381. Since the auxiliary models also have the same property, we see that the performance improvement is also not symmetric—even if the baseline performance happens to be symmetric. For example, MRR values for *ta*–*te* are 0.25 and 0.26, while the improvements are 21% and 42%.

Examples from kn-te. Table 4 shows some examples for *kn*-*te*. For each *kn* word, we take the translation correspondences using TI+Cue and AUX-COMB (with

[21] The model allows the inclusion of any number of auxiliary languages. However, our experimental setup requires the training pairs to be present in every auxiliary language corpus, so as to accurately measure the contribution of each auxiliary language. This restriction resulted in very small training sets when using three or more auxiliary languages, e.g. $|G(\{\text{en}, \text{fr}, \text{hi}\})| = 37$ for *kn*-*ml*. Due to this reason, we did not try with more auxiliary langauges for our chosen set of language pairs.

Table 3. Percentage improvement (over baseline MRR) of AUX-COMB using $S_{\{en,fr\}}()$ on $G(\{en, fr\})$

%Imp	bn	hi	kn	ml	mr	ta	te
bn	–	36.05	**92.45**	42.59	26.95	41.55	46.90
hi	24.96	–	31.77	28.94	34.75	25.95	43.81
kn	53.36	27.27	–	**82.33**	**89.51**	52.03	94.75
ml	13.98	22.83	**51.72**	–	23.26	18.77	68.03
mr	32.10	35.66	**95.94**	**83.78**	–	12.48	42.36
ta	39.64	17.78	23.22	15.50	19.12	–	45.39
te	33.60	38.21	**124.54**	10.24	70.74	55.37	–

Table 4. Examples: for each source *kn* word, we generate the translation correspondence using TI+Cue, and using AUX-COMB (with $S_{\{en,fr\}}$) and show (a) the top-ranked *te* word, and (b) the rank of the *te* translation

Source word		**TI+Cue**			$S_{\{en,fr,hi\}}$		
kn word	Meaning	*te* word at rank 1	Meaning	Rank of transln.	*te* word at rank 1	Meaning	Rank of transln.
ಎಲೆಕ್ಟ್ರಾನ್	electron	చక్కెర	sugar	20	ఎలక్ట్రాన్	electron	1
ರಸವಿದ್ಯೆ	chemistry	గ్రీకీక్	Greek	24	శాస్త్రం	science	3
ಶನಿ	saturn	సాహిత్యము	literature	9	శని	saturn	1
ಶಿಲೀಂಧ್ರ	fungus	పర్యావరణ	environment	32	లైకెన్	lichen	4
ಗುರುತ್ವ	gravitation	గురుత్వాకర్షణ	gravitation	1	కృష్ణ	dark	3
ಜೀವಸತ್ವಗಳು	vitamins	వ్యాధి	disease	55	విటమిన్	vitamin	1
ಜೀವವೈವಿಧ್ಯ	biodiversity	పర్యావరణ	environment	9	పర్యావరణ	environment	4
ಗುಲಾಮಗಿರಿ	slavery	కౌన్సిల్	council	2	బానిసత్వం	slavery	1

$S_{\{en,fr\}}()$) and show the *te* word at rank 1 and the rank of the correct *te* translation. We found that the top-ranked terms from both approaches were topically related but the translation was not usually at rank 1. However, AUX-COMB is able to use additional evidence from multiple languages and boost the probability of the translation so that it is ranked higher.

4.3 Further Analysis for AUX-COMB

Small Training Sets are Enough. We analyzed how sensitive our method was to the size of the training set used for learning λ. We chose the language pair *mr*-*te* since it had a sufficiently large gold set to allow training set size ablation, and sufficiently high performance to allow both positive and negative variation. In Figure 3 (left), we see the performance of AUX-COMB for different training set sizes. The overall trend suggests a very gradual increase in performance as training set size increases. For just 10 pairs, the performance is nearly as good as the performance for 70 pairs. The trend for *te*-*mr* was very similar.

Both Rare and Frequent Words Do Better. We analyzed how our method performed on words with different collection frequencies. For the language pair *te*-*mr*, we plotted the collection frequency of *te* words vs. percent improvement

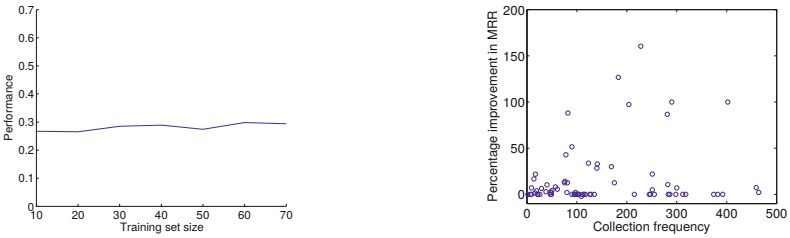


Fig. 3. *Left*: MRR for different training set sizes for *mr-te*. *Right*: Improvement in MRR for *te* terms with different collection frequencies, for *te-mr* with $S_{\text{en}}()$.

in MRR (Figure 3 (right)). We observe improvement over a wide range of frequencies, suggesting that the method is suitable for both rare as well as frequent words. The observations were similar for *mr* terms as well. We performed similar analyses for other term properties, viz. document frequency and average document count, and observed similar behavior.

4.4 Wikipedia Title Suggestion—User Study

We performed a user study on the *WikiTSu* system for the language pair Kannada-Hindi to assess the quality of the cross-lingual titles suggested. The quality of suggestions for source words that are Wikipedia titles has been studied in Section 4.2. In the user study, we focused on source words that are *not* Wikipedia titles. Since the Kannada Wikipedia (∼14,000 articles) is much smaller than the Hindi Wikipedia (∼100,000 articles), we chose Kannada as the source language.

Study Methodology. We randomly selected 3200 words from the *kn* corpus that were not titles, and removed common verbs, adjectives, parts of names, very common nouns, and noise words—these are unlikely to be article titles in *hi* (or any other language), giving a final list of 512 words. For each *kn* word k, we scored the *hi* vocabulary, and presented to the user the top-scoring *hi* word h that is also a Wikipedia title, with the following instructions: Suppose the user sees k in an article, and wants to know more about the concept K represented by the word k. Let H be the article corresponding to h. Score h as 1 if H is about the concept K, 0.5 if H contains information about concept K, and 0 otherwise. The above exercise was performed independently by two users.

Results. For each scoring method (TI+Cue and AUX-COMB), for each k, we averaged the relevance score given by the two users, and then averaged that over all k. The results (Table 5 (left)) show that using AUX-COMB leads to a **20% improvement** in the quality of titles. The Cohen's κ agreement between the users is good, but does not take the ordering the scores into account—a disagreement of 0 *vs.* 1 is worse than 0 *vs.* 0.5. We computed the weighted κ [55] using the weight matrix W [22] shown in Table 5 and found very good agreement.

[22] W_{ab} is the penalty when a title is given the score a by User 1, and b by User 2.

Table 5. User study on *WikiTSu*: Average relevance score of suggested titles and user agreement metrics (left), and the weight matrix for weighted κ (right)

	TI+Cue	AUX-COMB
Avg. relevance score	0.298	**0.360**
Agreement	83%	81%
Cohen's κ	0.69	0.68
Weighted κ	0.83	0.81

		User 2		
	W	1	0.5	0
User 1	1	0	1	3
	0.5	1	0	1
	0	3	1	0

5 Conclusions and Future Work

In this paper, we explored using auxiliary language corpora for CC-TCI. Using no resources other than comparable corpora, we demonstrated remarkable improvements in performance for 21 language pairs and applied the method to the crosslingual Wikipedia title suggestion task. This study raises interesting questions regarding the effect of the number of languages, language family, and corpus characteristics and quality. The model combination framework allows easy introduction of other cues besides auxiliary language corpora, e.g. transliteration models for names. We plan to explore these ideas in future work.

Acknowledgements. We thank Srivaths Ranganathan for the initial experiments, and Chaitra Shankar for help with the annotation. This work was supported by grants from Infosys Technologies Ltd. and the Department of Science and Technology, Government of India.

References

1. Schafer, C., Yarowsky, D.: Inducing translation lexicons via diverse similarity measures and bridge languages. In: COLING 2002 (2002)
2. Gaussier, E., Renders, J.M., Matveeva, I., Goutte, C., Djean, H.: A geometric view on bilingual lexicon extraction from comparable corpora. In: ACL 2004 (2004)
3. Andrade, D., Tsuchida, M., Onishi, T., Ishikawa, K.: Translation acquisition using synonym sets. In: NAACL-HLT (2013)
4. Tamura, A., Watanabe, T., Sumita, E.: Bilingual lexicon extraction from comparable corpora using label propagation. In: EMNLP-CoNLL (2012)
5. Haghighi, A., Liang, P., Berg-Kirkpatrick, T., Klein, D.: Learning bilingual lexicons from monolingual corpora. In: ACL-HLT (2008)
6. Koehn, P., Knight, K.: Learning a translation lexicon from monolingual corpora. In: ULA 2002 (2002)
7. Ismail, A., Manandhar, S.: Bilingual lexicon extraction from comparable corpora using in-domain terms. In: COLING (2010)
8. Vulić, I., De Smet, W., Moens, M.: Identifying word translations from comparable corpora using latent topic models. In: ACL-HLT (2011)
9. Udupa, R., Khapra, M.: Improving the multilingual user experience of wikipedia using cross-language name search. In: HLT 2010 (2010)

10. Bao, P., Hecht, B., Carton, S., Quaderi, M., Horn, M.S., Gergle, D.: Omnipedia: bridging the wikipedia language gap. In: CHI (2012)
11. Rapp, R.: Identifying word translations in non-parallel texts. In: ACL 1995 (1995)
12. Boyd-Graber, J., Blei, D.M.: Multilingual topic models for unaligned text. In: UAI (2009)
13. Lee, L., Aw, A., Zhang, M., Li, H.: Em-based hybrid model for bilingual terminology extraction from comparable corpora. In: COLING 2010 (2010)
14. Prochasson, E., Fung, P.: Rare word translation extraction from aligned comparable documents. In: HLT 2011 (2011)
15. Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., Gornostay, T.: Term extraction, tagging, and mapping tools for under-resourced languages. In: TKE (2012)
16. Laws, F., Michelbacher, L., Dorow, B., Scheible, C., Heid, U., Schütze, H.: A linguistically grounded graph model for bilingual lexicon extraction. In: COLING (2010)
17. Qian, L., Wang, H., Zhou, G., Zhu, Q.: Bilingual lexicon construction from comparable corpora via dependency mapping. In: COLING (2012)
18. Delpech, E., Daille, B., Morin, E., Lemaire, C.: Extraction of domain-specific bilingual lexicon from comparable corpora: Compositional translation and ranking. In: COLING (2012)
19. Yu, K., Tsujii, J.: Extracting bilingual dictionary from comparable corpora with dependency heterogeneity. In: HLT-NAACL (2009)
20. Shao, L., Ng, H.T.: Mining new word translations from comparable corpora. In: COLING 2004 (2004)
21. Déjean, H., Gaussier, É., Sadat, F.: An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In: COLING (2002)
22. Rapp, R.: Automatic identification of word translations from unrelated english and german corpora. In: ACL 1999 (1999)
23. Laroche, A., Langlais, P.: Revisiting context-based projection methods for term-translation spotting in comparable corpora. In: COLING (2010)
24. Morin, E., Daille, B., Takeuchi, K., Kageura, K.: Brains, not brawn: The use of smart comparable corpora in bilingual terminology mining. ACM Trans. Speech Lang. Process. (2008)
25. Rubino, R., Linarès, G.: A multi-view approach for term translation spotting. In: CLITP (2011)
26. Fišer, D., Ljubešic, N.: Bilingual lexicon extraction from comparable corpora for closely related languages. In: RANLP (2011)
27. Mausam, S.S., Etzioni, O., Weld, D.S., Skinner, M., Bilmes, J.: Compiling a massive, multilingual dictionary via probabilistic inference. In: ACL 2009 (2009)
28. Kaji, H., Tamamura, S., Erdenebat, D.: Automatic construction of a japanese-chinese dictionary via english. In: LREC (2008)
29. Udupa, R., Saravanan, K., Kumaran, A., Jagarlamudi, J.: Mint: a method for effective and scalable mining of named entity transliterations from large comparable corpora. In: EACL 2009 (2009)
30. Li, L., Wang, P., Huang, D., Zhao, L.: Mining english-chinese named entity pairs from comparable corpora. In: TALIP (2011)
31. Ji, H.: Mining name translations from comparable corpora by creating bilingual information networks. In: BUCC 2009 (2009)
32. Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: Improving the extraction of bilingual terminology from wikipedia. ACM TMCCA (2009)

33. Fung, P.: Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In: VLC (1995)
34. Vulić, I., Moens, M.-F.: Cross-lingual semantic similarity of words as the similarity of their semantic word responses. In: NAACL-HLT (2013)
35. Li, B., Gaussier, E.: Improving corpus comparability for bilingual lexicon extraction from comparable corpora. In: COLING (2010)
36. Su, F., Babych, B.: Development and application of a cross-language document comparability metric. In: LREC (2012)
37. Shezaf, D., Rappoport, A.: Bilingual lexicon generation using non-aligned signatures. In: ACL 2010 (2010)
38. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: KDD 2002 (2002)
39. Borin, L.: You'll take the high road and i'll take the low road: using a third language to improve bilingual word alignment. In: COLING (2000)
40. Mann, G.S., Yarowsky, D.: Multipath translation lexicon induction via bridge languages. In: NAACL 2001 (2001)
41. Tsunakawa, T., Okazaki, N.: ichi Tsujii, J.: Building bilingual lexicons using lexical translation probabilities via pivot languages. In: LREC (2008)
42. Wu, H., Wang, H.: Pivot language approach for phrase-based statistical machine translation. Machine Translation (2007)
43. Cohn, T., Lapata, M.: Machine translation by triangulation: Making effective use of multi-parallel corpora. In: ACL (2007)
44. Utiyama, M., Isahara, H.: A comparison of pivot methods for phrase-based statistical machine translation. In: HLT-NAACL (2007)
45. Kumar, S., Och, F.J., Macherey, W.: Improving word alignment with bridge languages. In: EMNLP-CoNLL (2007)
46. Khapra, M.M., Kumaran, A., Bhattacharyya, P.: Everybody loves a rich cousin: an empirical study of transliteration through bridge languages. In: HLT 2010 (2010)
47. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: ACL 2005 (2005)
48. Kim, W., Khudanpur, S.: Lexical triggers and latent semantic analysis for cross-lingual language model adaptation. ACM Transactions on Asian Language Information Processing (TALIP) 3, 94–112 (2004)
49. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist (1993)
50. Picard, R.R., Cook, R.D.: Cross-validation of regression models. JASA (1984)
51. Voorhees, E.M.: et al.: The trec-8 question answering track report. In: TREC (1999)
52. Mimno, D., Wallach, H.M., Naradowsky, J., Smith, D.A., McCallum, A.: Polylingual topic models. In: EMNLP 2009 (2009)
53. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu`
54. Heinrich, G.: Parameter estimation for text analysis. Technical report (2009)
55. Cohen, J.: Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit.. Psychological Bulletin (1968)

English-Arabic Statistical Machine Translation: State of the Art

Sara Ebrahim[1], Doaa Hegazy[1], Mostafa G.M. Mostafa[2], and Samhaa R. El-Beltagy[3]

[1] Scientific Computing Department, Ain Shams University, Cairo, Egypt
sara.elkafrawy@gmail.com, doaa_hegazy@fcis.asu.edu.eg
[2] Computer Science Department, Ain Shams University, Cairo, Egypt
mgmostafa@cis.asu.edu.eg
[3] Center for Informatics Science, Nile University, 6 October, Egypt
samhaa@computer.org

Abstract. This paper presents state of the art of the statistical methods that enhance English to Arabic (En-Ar) Machine Translation (MT). First, the paper introduces a brief history of the machine translation by clarifying the obstacles it faced; as exploring the history shows that research can develop new ideas. Second, the paper discusses the Statistical Machine Translation (SMT) method as an effective state of the art in the MT field. Moreover, it presents the SMT pipeline in brief and explores the En-Ar MT enhancements that have been applied by processing both sides of the parallel corpus before, after and within the pipeline. The paper explores Arabic linguistic challenges in MT such as: orthographic, morphological and syntactical issues. The purpose of surveying only En-Ar translation direction in the SMT is to help transferring the knowledge and science to the Arabic language and spreading the information to all who are interested in the Arabic language.

1 Introduction

Statistical Machine Translation (SMT) has become the most vital technique towards a comprehensive system for automatic translation. SMT has statistically significant good results compared to other techniques. SMT includes a training phase that uses statistical models to predict the most appropriate translation and considered to have similarities with the human way of learning.

The aim of focusing on the translation in the direction from English to Arabic is to transfer the knowledge to the Arab world. Revitalizing the Arabic language is the long term goal of this research; most of the studies concerning education claimed that learning with the native language is important and more productive than learning with a second language.

To the best of our knowledge, there is no survey dissected the En-Ar SMT. From the early beginning, there are survey papers that explained different techniques for MT in general such as: [Slo85], [Som92] and [DJB99]. After that period of time, survey papers focused on one of the methods such as: Example based

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 520–533, 2015.
DOI: 10.1007/978-3-319-18111-0_39

MT in [Som99] and SMT in [Lop08]. Arwa et al. in [AOS12] explored some of the Arabic linguistic topics then stated what has been done in the following techniques: Rule based, Statistical, Example based, Knowledge based and Hybrid Machine Translation. The survey was concerned with the Arabic language and they explored most recent research papers in both directions En-Ar and Ar-En translations.

The paper is organized as follows: section two scans a brief history of MT problem. Section three discusses SMT and gives the overview of the SMT pipeline /architecture. Then, section four explores important Arabic linguistic issues concerning translation in SMT technique. After that, section five presents the bilingual data and pipeline processing for En-Ar SMT. Section six gives an overview of the most common datasets and tools used in the discussed research papers. Finally, conclusions are drawn in section seven.

2 Machine Translation Brief History

Hutchins in [Hut95] presented an intensive historical research for the suggested solutions for of the automatic translation problem over the previous ages, as he mentioned in his paper the 17th century was the beginning; scientists started to develop mechanical dictionaries aiming to translate single words. Late in 20th century, two scientists proposed interdependency two ideas for translation: in 1933 George Artsouni, a French-Armenian and Peter Smirnov-Troyanskii. Artsrouni designed a storage device on paper tape which could be used to find the equivalent of any word in another language. Troyanskii envisioned three stages of mechanical translation.

Warren Weaver was one of the pioneers in the machine translation field. A message wrote by Weaver in 1949 was the start that described the need and possibility for computers to translate text. The message was: "I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need do is strip off the code in order to retrieve the information contained in the text." as Zughoul et. al. mentioned in their survey paper [ZAA05].

Weaver outlined the methods we are investigating now, he suggested useful methods and theories such as statistical methods, Shannon's information theory, and the exploration of the underlying logic and universal features of language [Hut95]. Research was growing in those days in many centers and mainly for political reasons; the United States (US) research community focused on Russian-English translation, and the Soviet research community focused on English-Russian translation.

Expectations and optimism were high in 1950s about the MT field. In 1964 the US formed the Automatic Language Processing Advisory Committee (ALPAC) which had a role to examine the future of the field. In 1966 ALPAC announced its famous report about investing in the MT field. The report stated that: it is useless and expensive to invest in the MT field as it needs more cost, and time than human translators and it is useful instead to help translators by developing

dictionaries and other helper tools for them. One of the consequences of the report was the down funding for research in MT field. [Hut95].

Without going any further in history, the main point for this brief history is that current online translation systems like Google and Bing are obvious evidence to the lake of efficiency and short insight of the ALPAC report. Research in MT field is open as long as there are an open minded researchers that can introduce new and innovative solutions to MT.

3 Statistical Machine Translation

SMT requires a large parallel corpus. The basic idea is to use this large parallel corpus in the training phase to produce translation examples by dividing it into controllable smaller pieces (sentences). The idea goes back to [Wea55] ; he suggested applying the statistical and cryptanalytic techniques to translation. He stated in[Wea55]:

"One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'"

3.1 Pipeline

First, SMT requires a large bilingual translated text often called a parallel corpus. Second, text segmentation and alignment processes are done for both sides of the parallel corpus. Third, text segments are trained to build the statistical models needed in the translation. Fourth, a decoding process is done by taking a phrase translation model and a language model to finally produce the most appropriate translation for the input source text that is needed to be translated. Finally, the output translation from the system is evaluated whether manually or automatically.

3.2 Relation to the Noisy Channel Model

Translation process in SMT has been formulated as a noisy channel model. Claude Shannon established the noisy channel to model the communication between transmitter and receiver in certain medium known as noisy channel [Sha48]. In communication theory the source message is unknown and a process is done to guess the original message.

In SMT we actually know the source message (the source language sentence) and we want to guess the target message (the translation of the message), unlike in communication theory. Consider that we want to translate an English (source s) into Arabic (target sentence t).[1]

A noisy channel model has two components: Language Model (LM) and Translation Model. Language Model $p(t)$ means that how likely the sentence is really

[1] Notations for source and target languages are s and t for the rest of this paper.

Arabic. LM can be a trigram model, a factored model (which will be referred to later in this paper) or other types. LM is trained from monolingual corpus; no need for a parallel corpus. Translation model $p(s|t)$ is trained from En-Ar parallel corpus. The parameters of this model will be estimated from the translated language pair. The noisy-channel approach uses Bayes rule:

$$p(t|s) = \frac{p(t,s)}{p(s)} = \frac{p(t)p(s|t)}{\sum_t p(t)p(s|t)} \tag{1}$$

Hence,

$$argmax\ _{t\epsilon A}\, p(t|s) = argmax\ _{t\epsilon A}\, \frac{p(t)p(s|t)}{\sum_t p(t)p(s|t)} \tag{2}$$

The task can be formulated as searching for an Arabic sentence that maximizes the product of language model and translation model. The output of the translation model on a new Arabic sentence t is:

$$t^* = argmax_{t\epsilon A}\ \ p(t) \times p(s|t) \tag{3}$$

where A is the set of all sentences in Arabic. Thus the score for a potential translation t is the product of two scores. First, the language-model score $p(t)$, which gives a prior distribution over which sentences are likely in Arabic. Second, the translation-model score $p(s|t)$, which indicates how likely we are to see the English sentence s as a translation of t. The architecture with the Bayes rule decision used in this section is illustrated in Figure 1.

4 Arabic Linguistic Challenges

Translation is a hard task for a computer to produce automatically. Difficulties arise in MT task when the two languages are close to each other such as lexical ambiguity and pronoun resolution. However, in a two distant languages like English and Arabic there are more difficulties; mostly because Arabic is a complex language to be understood by a computer. We will highlight in this section the Arabic linguistic issues orthographically, morphologically and syntactically.

4.1 Orthographic Issues

Certain letters in Arabic script are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words). In particular, variants of Hamzated Alif (آ، أ، إ) are often written without their Hamza (ا), also the Alif-Maqsura/dotless Ya (ى) and the regular dotted Ya (ي) are often used interchangeably in word final position [EKH10].[2]

[2] All Arabic transliteration are provided in the Habash-Soudi-Buckwalter transliteration scheme [HSB07].

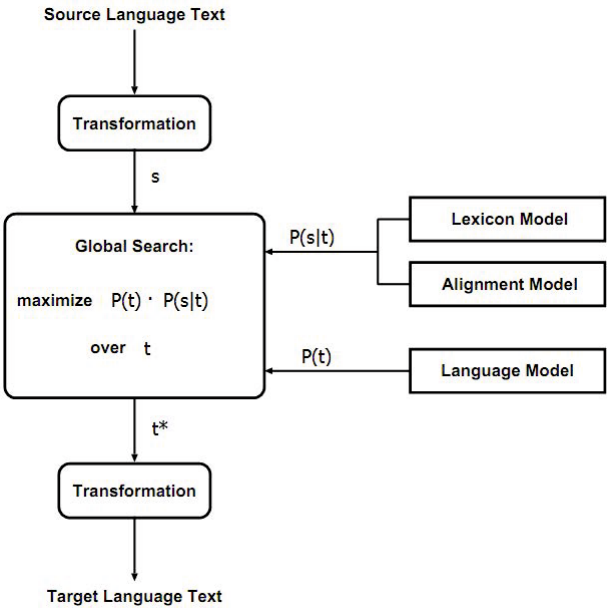


Fig. 1. Architecture of the translation approach based on Bayes decision rule. Adapted from: [ON00].

4.2 Morphological Issues

Arabic has a rich morphology compared with English language. Richness is a fact because words are being inflected for gender and number for example, and also because words attach to various clitics for: conjunction(*w + 'and'*), the definite article(*Al+ 'the'*), preposition(e.g. *b+ 'by/with'*, *l+ 'for'*, *k+ 'as'*), possessive pronouns and object pronouns(e.g. *+ ny 'me/my'*, *+hum 'their/them'*). For example, the verbal form *wsnqAblhum* وسنُقَابلهُم and the nominal form *wbsyyArAtnA* وبسيَّارَاتنَا can be decomposed as follows:

```
1.    w+    s+     n+    qAbl+  hum
      and+  will+  we+   meet+  them
2.    w+    b+     syyAr+   At+ nA
      and+  with+  car+     PL+ our
```

This richness causes Arabic corpus to have more surface forms than an equivalent English corpus and the problem of sparsity appears. El Kholy and Habash in [EKH12] mentioned that "While the number of (morphologically untokenized) Arabic words in a parallel corpus is 20% less than the number of corresponding

English words, the number of unique Arabic word types is over twice the number of unique English word types over the same corpus size." [3]

4.3 Syntactical Issues

Arabic is more complex than English syntactically. There are three dominant issues in Arabic syntax: verb-subject order, adjectives and *Idafa* construct which is the equivalent of the English possessive, compound nouns and the *of*-relationship.

Verb-Subject Order. An Arabic sentence usually has the order Verb-Subject-Object (VSO). An English sentence usually has the order Subject-Verb-Object (SVO) which also occurs in Arabic but less frequently than in English. Examples (1,2) show the different ordering in Arabic. Example(3) shows the verb-subject gender agreement in VSO order and example(4) shows the verb-subject gender and number agreement in SVO order.

1. `ktb Alwld Aldrs`
   `wrote the boy the lesson`
   `En: The boy wrote the lesson.`
   Ar: كتب الولد الدرس
2. `Alwld ktb Aldrs`
   `the boy wrote the lesson`
   `En: The boy wrote the lesson.`
   Ar: الولد كتب الدرس
3. `ktb AlA'wlAd Aldrws`
   `wrote the boys the lessons`
   `En: The boys wrote the lessons.`
   Ar: كتب الأَولَاد الدروس
4. `AlA'wlAd ktbw Aldrws`
   `the boys wrote the lessons`
   `En: The boys wrote the lessons.`
   Ar: الأَولَاد كتبُوا الدروس

Arabic Adjectives. Arabic noun phrase has different structure from English. The adjective in Arabic which modifies a noun follows the noun in definiteness, so it is added to the definite article if the noun is definite and vice versa:

1. `Alyd Alkbyra`
   `the hand the big`
   `En: The big hand.`
   Ar: اليد الكبيرة

[3] Examples and problem definition in this section are adapted from Badr et al. [BZG08].

2.
```
yd      kbyra
hand    big
En: A big hand.
```
Ar: يد كبيرة

***Idafa* Construct.** *Idafa* construct is the Arabic equivalent to English possessive, noun compounding and the *of*-relationship. The three English structures are translated to the *Idafa* which has a one or more indefinite nouns followed by a definite noun. For example the English phrases (*the student books*, *the student's books* and *the books of the student*) all translated to one Arabic sentence which is (ktb AlTAlb - كتُب الطَالب)

5 English to Arabic Modifications in SMT

Organizations interested in research, have been increasing funds for Arabic Natural Language Processing (ANLP) in general and in Arabic-to-English translation in the MT field. Farghaly and Shaalan mentioned in [FS09] that the fund has been raising since 11 September 2001 and [Koe05] previously stated part of the fact: "Due to the involvement of US funding agencies, most research groups focus on the translation from Arabic to English and Chinese to English. Next to text-to-text translation, there is increasing interest in speech-to-text translation."

Despite this fact, a number of scientists have been trying to improve the translation from English-to-Arabic; they made modifications as preprocessing and postprocessing to both sides of the parallel corpus. They also made modifications in the used LM within the SMT pipeline. Current preprocessing techniques for Arabic are: orthographic normalization, morphological tokenization /decomposition and syntactic reordering. Current post-processing techniques for Arabic are: orthographic enrichment and morphological detokenization /recombination. Current preprocessing techniques for English are: down-casing, separating punctuation from words and splitting off ('s). Current in-process enhancements are inside the language model; they used Factored Language Model (FLM) which integrates additional features to each word: Part Of Speech (POS), number, gender, semantic class, etc.

5.1 Preprocessing Techniques

Orthographic Normalization. Badr et al. in [BZG08] named this process as a Normalization process. However, El Kholy and Habash in [EKH12] named it as Orthographic normalization. Orthographic normalization has two forms: Enriched (ENR) and Reduced (RED) forms. RED form is a reductive process that converts all Hamzated Alif forms(آ، أ، إ) to bare Alif (ا) and dotless Ya or Alif Maqsura (ى) to dotted Ya (ي). ENR form has one difference than RED form which is selecting appropriate form of Alif. El Kholy and Habash in [EKH12]

proposed the two forms and named them. But actually one of them which is the RED form was used by [BZG08]. It is well known that ENR Arabic is the desired form to generate and to evaluate against.

Morphological Tokenization. Separating the Arabic word to its parts is an important process before training the data. The problem is that an Arabic corpus will have more surface forms than an English corpus of the same size and the purpose of tokenizing Arabic text is to reduce sparsity and decrease the number of out-of-vocabulary (OOV) words [BZG08]. Badr et al. in [BZG08] named this process as Segmentation of the text. El Kholy and Habash in [EKH12] claimed that there is a slight difference between segmentation and tokenization. For example the segmentation of the word (maktbthom - مَكتبتهُم) is split off to be (maktbt + hum - مَكتَبت + هُم) and not the correct word (maktaba - مَكتبَة). That is because [EKH12] consider orthographic/morphological as an adjustment to the segmentation process. They call the whole process a morphological tokenization. Figure 2 illustrates some of the adjustment rules as El Kholy and Habash mentioned in [EKH12].

Rule Name	Tokenized	Untokenized	Example		
			Tokenized	Untokenized	Gloss
Definite Article	?ل+ال *l+Al+l ?*	+لل *ll+*	ل+ال+مكتب *l+Al+mktb*	للمكتب *llmktb*	'for the office'
			ل+ال+الجنة *l+Al+ljnħ*	للجنة *lljnħ*	'for the committee'
Ta-Marbuta	ة- *-ħ* +pron	ت- *-t* +pron	مكتبة+هم *mktbħ+hm*	مكتبتهم *mktbthm*	'their library'
Alif-Maqsura	ى- *-ý* +pron	ا- *-A* +pron	روى+ه *rwý+h*	رواه *rwAh*	'he watered it'
	exceptionally	ي- *-y* +pron	على+ه *ςlý+h*	عليه *ςlyh*	'on him'
Hamza	ء- *-'* +pron	ئ- *-ŷ* +pron	بهاء+ه *bhA'+h*	بهائه *bhAŷh*	'his glory [gen.]'
	less frequently	ؤ- *-ŵ* +pron	بهاء+ه *bhA'+h*	بهاؤه *bhAŵh*	'his glory [nom.]'
	less frequently	ء- *-'* +pron	بهاء+ه *bhA'+h*	بهاءه *bhA'h*	'his glory [acc.]'

Fig. 2. Examples of some Arabic morphological adjustment rules. Source: [EKH12].

El Kholy and Habash in [EKH12] proposed six schemes for morphological tokenization. Figure 3 illustrates the six schemes with a sentence example. The schemes are (D0, D1, D2, TB, S2, and D3), Where D0 is the surface word form. Their work was an extension to [BZG08]. They made two extended contribution. First they presented a comparison of a larger number of tokenization schemes, while [BZG08] used only S2 and D3 schemes. Second is that they discussed the issue of producing unnormalized Arabic output, while [BZG08] reported their work only on normalized Arabic. El Kholy and Habash in [EKH12] reported that their results were consistent with [BZG08]'s results regarding D0 and D3 but TB result outperform S2. They noticed that training over RED Arabic followed by enriching its output sometimes yields better results than training on ENR directly which is the case with TB tokenization scheme.

Arabic	وسينهى الرئيس جولته بزيارة الى تركيا.						
	wsynhý	Alrŷys	jwlth	bzyArħ	Alý	trkyA	.
Gloss	and will finish	the president	tour his	with visit	to	Turkey	.
English	The president will finish his tour with a visit to Turkey.						
Scheme							
D0	wsynhy	Alrŷys	jwlth	bzyArħ	Ălý	trkyA	.
D1	w+ synhy	Alrŷys	jwlth	bzyArħ	Ălý	trkyA	.
D2	w+ s+ ynhy	Alrŷys	jwlth	b+ zyArħ	Ălý	trkyA	.
TB	w+ s+ ynhy	Alrŷys	jwlħ +h	b+ zyArħ	Ălý	trkyA	.
S2	w+s+ ynhy	Al+ rŷys	jwlħ +h	b+ zyArħ	Ălý	trkyA	.
D3	w+ s+ ynhy	Al+ rŷys	jwlħ +h	b+ zyArħ	Ălý	trkyA	.
LEM	Ânhý	rŷys	jwlħ	zyArħ	Ălý	trkyA	.

Fig. 3. A sentence in the various tokenization schemes. Source: [EKH12].

Syntactic Reordering. Badr et al. in [BZG09] proposed set of rules on the English source to align better with Arabic translation. The implementation was straight forward by using parse tree. The rules are: (1) Noun Phrase(NP): all nouns, adjectives and adverbs are inverted in a NP. (2) Prepositional Phrase (PP): prepositional phrases of form N1 of N2 of Nn are transformed to N1 N2 Nn. (3) Definite article (the): "the" is replicated before adjectives. (4) Verb Phrases(VP): transforms SVO order to VSO. First they tag English source text with Stanford log-linear POS tagger. Then they split the tagged text into small sentences and tag them by maximum entropy tagger [R+96], parse them using Collins parser [Col97]. Finally they tag person, location and organization names by Stanford Named Entity Recognition (NER) tagger for English side [FGM05] . For Arabic side they normalize the text, then segment the text using Morphological Analysis and Disambiguation for Arabic (MADA) toolkit. They found that replicating "the" before adjectives hurts the scores. They proved that the previous rules made significant gain. Moreover, they integrated syntactic reordering with morphological decomposition and recombination techniques. Habash in [Hab07] has made a trial for syntactic reordering in Arabic-to-English translation and also reordering the source language which here is Arabic. He showed that if the parse quality is not good it will reflect the translation as well; that is because Arabic parsing has a limited work compared to English parsing.

Habash and Elming in [EH09] proposed a syntactic reordering method before translating English-to-Arabic using statistical methods. The technique was produced firstly by [Elm08] but on a close language pair English-Danish. They proved that the technique was viable for distant language pair like English-Arabic.

5.2 Post-processing Techniques

Morphological Detokenization. It is the reverse process for morphological tokenization. Badr et al. in [BZG08] proposed four schemes for recombining Arabic translated text: Simple(S), Rule-based(R), Table-based (T) and Table+Rule(T+R). They call the process morphological recombination while

El Kholy and Habash in [EKH10] named it as Morphological Detokenization. El Kholy and Habash in [EKH10] extended two more schemes for detokenization/recombination for Arabic translated text. The six schemes are: Simple(S), Rule-based(R), Table-based (T), Table+Rule(T+R), Table+Language Modeling (T+LM) and (T+R+LM). Badr et al. in [BZG08] reported that (T+R) technique was the best performer in the experiments. Moreover El Kholy and Habash in [EKH10] reported that (T+R+LM) technique was the best one in all conditions; (T+R+LM) technique was first experienced with [EKH10].

Orthographic Enrichment and Detokenization. El Kholy and Habash in [EKH10] proposed two techniques for orthographic enrichment and detokenization. Reduced tokenized output should be enriched and detokenized to produce proper Arabic. First technique is by using Morphological Analysis and Disambiguation for Arabic (MADA) toolkit [HR05] to enrich detokenized reduced text (MADA-ENR). The other technique is detokenizing and enriching in one joint step (Joint-DETOK-ENR). [EKH10]'s joint technique was better than performing the two tasks in two separate steps. The best setup for the MT as a whole in [EKH10]'s experiments is on RED text and then apply the joint technique; enriching and detokenizing in one step.

5.3 Factored Language Model

Sarikaya and Deng in [SD07] proposed a Joint Morphological-Lexical Language Modeling (JMLLM) for MT. The process begins with a morphological segmentation for Arabic text. They proposed a tree structure called Morphological-Lexical Parse Tree (MLPT) to combine the morphological information with lexical information in a single JMLLM as illustrated in Figure 4. The idea is to split word into segments to form meaningful lexical unit. An example of the MLPT is shown in Figure 4; each word has three attributes in the first level (type, gender, number). Type is considered to be the POS tag and here Noun(N) and Verb(V) are only considered. Gender can be Masculine(M) of Feminine(F) and number can be Singular(S) or Plural(P). They do not use the first level of the tree in Figure 4 (NFS, VFP, NFS and NMP) because lexical attributes are not available. The authors mentioned that they are in the process of labeling the data. Training the JMLLM has two degrees; (loose integration) and (tight integration). Loose integration is used in their experiments because the training time was faster than using tight integration. Badr et al. in [BZG08] used factored models; they put factors on both sides of the parallel corpus. An English word factors are: the surface word and the POS tag, while the factors on an Arabic word are: the surface word, stem and POS tag concatenated with the segmented clitics. For example for the word *wlAwlAdh* (and for his kids), the factored words are: *AwlAd* and *w+l+N+P:3MS*.

Khemakhem et al. in [KJ13] formulated the importance of factored models: "one of the problems of statistical language models is to consider that the word is depending only on its previous history (words or classes). But in fact, in

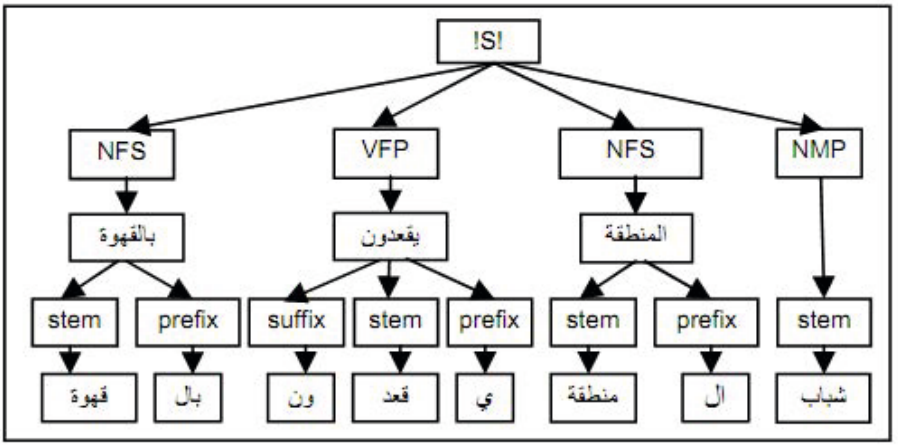


Fig. 4. Morphological-Lexical Parse Tree. Source: [SD07].

natural language the appearance of a word depends not only on its history but also on some other features". They mentions an example for two words having the same surface form but with different meaning in contexts. The word *katab* (write) and the word *kotob* (books). Perhaps if the two words have the correct diacritics it will not be a problem to identify the meaning of them. Moreover, they introduced two features to attach them for each Arabic word: the word itself and the syntactic class (noun, verb, particle and proper noun).

6 Datasets and Tools

Datasets consist of monolingual and parallel corpus for both language model and translation model. Eisele et al. in [EC10] described the acquisition of a multilingual corpus extracted from the official documents of the United Nations (UN). The multilingual corpus which they named it multiUN has six languages: Arabic, Chinese, English, French, Russian and Spanish. They made it available to the research community through the website of the EuroMatrixPlus project [4]. Another free En-Ar parallel corpus is STRAND corpus and it needs processing to be ready for SMT. It is a system for automatically acquiring pairs of documents in parallel translation on the World Wide Web. The corpus itself is not available due to copyrights restrictions of the online pages. Instead, this URL [5] provides databases of URL pairs acquired by STRAND, which you can download yourself for personal use [RS03]. One of the non-free corpus is distributed by Linguistics Data Consortium (LDC). It is an organization that creates and distributes a wide array of language resources [6].

[4] `http://www.euromatrixplus.eu/downloads`
[5] `http://www.umiacs.umd.edu/~resnik/strand/`
[6] `https://www.ldc.upenn.edu/`

Morphological Analysis and Disambiguation for Arabic (MADA) toolkit [HR05] is widely used in the previously discussed papers for producing various enriched forms and tokenization schemes. Pasha et al. in [PABK+14] introduced MADAMIRA toolkit; it is an integration of the features of MADA and AMIRA [DHJ07] and it is public for research use. For English processing the most common used tools are: Stanford Named Entity Recognizer, Stanford Log-Linear POS tagger [TKMS03] and a maximum entropy inspired parser [Cha00].

For the SMT pipeline, SRILM toolkit is widely used for language modeling [S+02]. GIZA++ [ON03] is widely used in word alignment and decoding is done using the phrase-based open source SMT system [KHB+07] and it is almost used in conducting experiments in the previously discussed papers.

7 Conclusions

English-to-Arabic translation direction is highly under-represented in MT research compared to the opposite direction. Limited work has been done since 2007. This work shows that there are number of ways to enhance this direction. MT in Arabic needs researchers and junior scientists as long as professors with experience in order to level up the field and enrich the Arabic content in the world.

References

[AOS12] Alqudsi, A., Omar, N., Shaker, K.: Arabic machine translation: a survey. Artificial Intelligence Review, 1–24 (2012)

[BZG08] Badr, I., Zbib, R., Glass, J.: Segmentation for english-to-arabic statistical machine translation. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 153–156. Association for Computational Linguistics (2008)

[BZG09] Badr, I., Zbib, R., Glass, J.: Syntactic phrase reordering for english-to-arabic statistical machine translation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 86–93. Association for Computational Linguistics (2009)

[Cha00] Charniak, E.: A maximum-entropy-inspired parser. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference, pp. 132–139. Association for Computational Linguistics (2000)

[Col97] Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 16–23. Association for Computational Linguistics (1997)

[DHJ07] Diab, M., Hacioglu, K., Jurafsky, D.: Automated methods for processing arabic text: from tokenization to base phrase chunking. In: Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer (2007)

[DJB99] Dorr, B.J., Jordan, P.W., Benoit, J.W.: A survey of current paradigms in machine translation. Advances in Computers 49, 1–68 (1999)

[EC10] Eisele, A., Chen, Y.: Multiun: A multilingual corpus from united nation documents. In: LREC (2010)

[EH09] Elming, J., Habash, N.: Syntactic reordering for english-arabic phrase-based machine translation. In: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages, pp. 69–77. Association for Computational Linguistics (2009)

[EKH10] El Kholy, A., Habash, N.: Techniques for arabic morphological detokenization and orthographic denormalization. In: Editors & Workshop Chairs, p. 45 (2010)

[EKH12] El Kholy, A., Habash, N.: Orthographic and morphological processing for english–arabic statistical machine translation. Machine Translation 26(1-2), 25–45 (2012)

[Elm08] Elming, J.: Syntactic reordering integrated with phrase-based smt. In: Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation, pp. 46–54. Association for Computational Linguistics (2008)

[FGM05] Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 363–370. Association for Computational Linguistics (2005)

[FS09] Farghaly, A., Shaalan, K.: Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP) 8(4), 14 (2009)

[Hab07] Habash, N.: Syntactic preprocessing for statistical machine translation. MT Summit XI, 215–222 (2007)

[HR05] Habash, N., Rambow, O.: Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 573–580. Association for Computational Linguistics (2005)

[HSB07] Habash, N., Soudi, A., Buckwalter, T.: On arabic transliteration. In: Arabic Computational Morphology, pp. 15–22. Springer (2007)

[Hut95] John Hutchins, W.: Machine translation: A brief history. In: Concise History of the Language Sciences: from the Sumerians to the Cognitivists, pp. 431–445 (1995)

[KHB+07] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Richard, Zens, o.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. Association for Computational Linguistics (2007)

[KJ13] Khemakhem, I.T., Jamoussi, S.: Integrating morpho-syntactic features in english-arabic statistical machine translation. In: ACL 2013, p. 74 (2013)

[Koe05] Koehn, P.: Europarl: A parallel corpus for statistical machine translation. MT Summit 5, 79–86 (2005)

[Lop08] Lopez, A.: Statistical machine translation. ACM Computing Surveys (CSUR) 40(3), 8 (2008)

[ON00] Och, F.J., Ney, H.: Statistical machine translation. In: EAMT Workshop, pp. 39–46 (2000)

[ON03] Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)

[PABK+14] Pasha, A., Al-Badrashiny, M., Kholy, A.E., Eskander, R., Diab, M., Habash, N., Pooleery, M., Rambow, O., Roth, R.: Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In: Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland (2014)

[R+96] Ratnaparkhi, A., et al.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, vol. 1, pp. 133–142 (1996)

[RS03] Resnik, P., Smith, N.A.: The web as a parallel corpus. Computational Linguistics 29(3), 349–380 (2003)

[S+02] Andreas, S., et al.: Srilm-an extensible language modeling toolkit. In: INTERSPEECH (2002)

[SD07] Sarikaya, R., Deng, Y.: Joint morphological-lexical language modeling for machine translation. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pp. 145–148. Association for Computational Linguistics (2007)

[Sha48] Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27 (1948)

[Slo85] Slocum, J.: A survey of machine translation: its history, current status, and future prospects. Computational Linguistics 11(1), 1–17 (1985)

[Som92] Somers, H.L.: Current research in machine translation. Machine Translation 7(4), 231–246 (1992)

[Som99] Somers, H.: Review article: Example-based machine translation. Machine Translation 14(2), 113–157 (1999)

[TKMS03] Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: 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, vol. 1, pp. 173–180. Association for Computational Linguistics (2003)

[Wea55] Weaver, W.: Translation. Machine Translation of Languages 14, 15–23 (1955)

[ZAA05] Zughoul, M.R.: English/arabic/english machine translation: A historical perspective. Meta: Journal des traducteursMeta:/Translators' Journal 50(3), 1022–1041 (2005)

Mining Parallel Resources for Machine Translation from Comparable Corpora

Santanu Pal[1], Partha Pakray[2], Alexander Gelbukh[3], and Josef van Genabith[1]

[1] Universität Des Saarlandes, Saarbrucken, Germany
[2] National Institute of Technology, Mizoram, India
[3] Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
{santanu.pal,josef.vangenabith}@uni-saarland.de,
www.parthapakray.com, www.gelbukh.com

Abstract. Good performance of Statistical Machine Translation (SMT) is usually achieved with huge parallel bilingual training corpora, because the translations of words or phrases are computed basing on bilingual data. However, in case of low-resource language pairs such as English-Bengali, the performance is affected by insufficient amount of bilingual training data. Recently, comparable corpora became widely considered as valuable resources for machine translation. Though very few cases of sub-sentential level parallelism are found between two comparable documents, there are still potential parallel phrases in comparable corpora. Mining parallel data from comparable corpora is a promising approach to collect more parallel training data for SMT. In this paper, we propose an automatic alignment of English-Bengali comparable sentences from comparable documents. We use a novel textual entailment method and distributional semantics for text similarity. Subsequently, we apply template-based phrase extraction technique to aligned parallel phrases from comparable sentence pairs. The effectiveness of our approach is demonstrated by using parallel phrases as additional training examples for an English-Bengali phrase-based SMT system. Our system achieves significant improvement in terms of translation quality over the baseline system.

1 Introduction

Many natural language processing tasks such as corpus-based machine translation (MT) heavily rely on bilingual parallel corpora. Statistical Machine Translation (SMT) is a kind of corpus-based MT based on probabilistic translation models. The model is learned from sentence-aligned parallel corpora. However, a major problem of SMT is scarcity of available parallel data. Many language pairs, such as English to Indian languages, suffer from the scarcity of parallel data.

Comparable corpora provide a possible solution to this data scarceness problem to some extent. Comparable documents are not strictly parallel: the corpus consists of bilingual documents but they are not sentence-aligned; more precisely, they are rough translations of each other. The sentences of comparable corpora are not really

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 534–544, 2015.
DOI: 10.1007/978-3-319-18111-0_40

translations, but they convey the same information and hence there must exist some sentential or sub-sentential level of parallelism.

Recently, comparable corpora are considered as a valuable resource for acquiring parallel data, which can play an important role in improving the quality of machine translation (MT) (Smith et al. 2010). The extracted parallel texts from comparable corpora are typically added with the training corpus as additional training material that is expected to improve performance of SMT systems, specifically for low-density language pairs.

In this paper, we describe a methodology for extracting English-Bengali parallel resource from comparable corpora. We did it in three steps. At the first step, we clustered the both side of bilingual comparable corpus into several groups using a textual entailment (TE) method. At the second step, we established cross-lingual link between the groups using n-best list of probabilistic bilingual lexicon. The bilingual lexicons are extracted from bilingual training data of the same domain by using a statistical word alignment tool GIZA++. At the final step, we used a template-based phrase extraction technique between each aligned groups. The extracted phrases were aligned using a baseline PB-SMT system, which was trained on the same-domain English-Bengali parallel corpus.

We collected document-aligned comparable corpus of English-Bengali document pairs from Wikipedia, which provides a huge collection of documents in many different languages.

Typically, there are two approaches can be applied for grouping corpus according to their text similarity: textual entailment and semantic textual similarity. Textual Entailment is defined by Dagan et al. (2004) as follows: text T is said to entail hypothesis H if H can be inferred from T. The task of textual entailment is to decide whether the meaning of H can be inferred from the meaning of T. For example, the text T = "*John's assassin is in jail*" entails the hypothesis H = "*John is dead*"; indeed, if there exists one's assassin, then this person is dead. However, T = "*Mary lives in Europe*" does not entail H = "*Mary lives in US*".

On the other hand, Semantic Textual Similarity (STS)[1] task measures the degree of semantic equivalence between a pair of texts, e.g. sentences. Four STS evaluation tasks were organized in 2012, 2013, 2014, and 2015 at SemEval workshops. STS is related to both Textual Entailment (TE) and paraphrasing, but differs in a number of ways and it is more directly applicable to a number of NLP tasks. STS is different from TE inasmuch as it assumes bidirectional graded equivalence between the pair of textual snippets. In case of TE the equivalence is directional, e.g. a car is a vehicle, but a vehicle is not necessarily a car. STS also differs from both TE and Paraphrase in that, rather than being a binary yes/no decision (e.g. a vehicle is not a car), STS is a graded similarity notion (e.g. a vehicle and a car are more similar than a wave and a car). This graded bidirectional nature of STS is useful for NLP tasks such as MT evaluation, information extraction, question answering, and summarization. The Textual Entailment system is unidirectional but Semantic Textual Similarity is mainly bidirectional. Therefore, we will also use Sematic Textual Similarity technique also.

[1] http://ixa2.si.ehu.es/stswiki/index.php/Main_Page

The main goal of Semantic Textual Similarity (STS) task (Agirre et al., 2014) is to measure the degree of semantic equivalence between a pair of texts, e.g. sentences.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the mining process of the comparable corpora. The TE system architecture is described in Section 4. Section 5 describes the automatic alignment technique of parallel fragment of texts. Section 6 describes the Dataset used for this work. The baseline system setup is demonstrated in Section 7. Experiments and evaluation results are presented in section 8. Section 9 concludes and presents avenues for future work.

2 Related Works

Comparable corpora have been used in many research areas in NLP, especially in machine translation. Several earlier works have studied the use of comparable corpora in machine translation. However, most of these approaches (Fung and McKeown, 1997; Fung and Yee, 1998; Rapp, 1999; Chiao and Zweigenbaum, 2002; Dejean et al., 2002; Otero, 2007; Saralegui et al., 2008; Gupta et al., 2013) are specifically focused on extracting word translations from comparable corpora. Most of the strategies follow a standard method based on the context vector similarity measure such as finding the target words that have the most similar distributions with a given source word. In the majority of the cases, a starting list contains the "seed expressions" and this list is required to build the context vectors of the words in both the languages. A bilingual dictionary can be used as a starting list. The bilingual list can also be prepared form parallel corpus using bilingual correlation method (Otero, 2007). Instead of a bilingual list, multilingual thesaurus could also be used for this purpose (Dejean, 2002). Pal et al., 2014 applied TE method for extracting parallel text from comparable corpora.

Wikipedia is a multilingual encyclopedia available in different languages and it can be used as a source of comparable corpora. Otero et al. (2010) stored the entire Wikipedia for any two languages and transformed it into a new collection: CorpusPedia. Our work shows that only a small ad-hoc corpus containing Wikipedia articles could prove to be beneficial for existing MT systems.

The main objective of the present work is to investigate whether textual entailment can be used to establish alignments between text fragments in comparable corpora and whether the parallel text fragments extracted thus can improve MT system performance.

Textual similarity problem may be tackled by various techniques at lexical, syntactic, and semantic levels, as usual during NLP processing. Among lexical techniques, there are word overlap metrics or n-gram matching. Another way is to compare dependency relations of two texts. For computation, one can use synonyms, hypernyms, etc. The higher processing level, the better performance is usually achieved. There always remain some examples that cannot be decided by lexical, syntactic, or semantical analysis, because full knowledge and meaning representation is needed for it. There is semantic gap between lexical surface of the text and its meaning because same concepts are represented in different vocabulary, languages,

formalisms, and notations. Updating knowledge databases with all dialectical possibilities in supervised way is doomed to failure. In distributional semantics approaches (Blei et al., 2003), similarities between linguistic items could be computed from their collocativity and distributional properties in large samples of language data in unsupervised way, as clearly seen from visualization experiments (Chaney et al., 2012). Especially convincing are recent experiments computed by Gensim framework (Rehurek and Sojka, 2010) where words and phrases are computed by Word2Vec (Mikolov et al., 2013) language model.

3 Comparable Corpora Collection

Wikipedia is a huge collection of large varieties of topics of articles in a wide variety of languages and it is available in various domains. Wikipedia links articles on the same topic in different languages using "interwiki" linking facility. Thus, document alignment for multi-lingual documents on similar topics is already provided in Wikipedia, which can be directly applied for sentence or sub-sentential phrase extraction step.

We designed a crawler to collect comparable corpora for English-Bengali document pairs. Based on an initial *"seed keyword list"*, the crawler first visits each English page of Wikipedia, saves the raw text (in HTML format), and then follows the cross-lingual link for each English page and collects the corresponding Bengali document. We keep only the textual information and all the other details are discarded. The "*seed keyword list"* is mainly named entity (NE) list, collected from English tourism domain corpus. We extract English and Bengali sentences from each document; those are not parallel. Moreover, Bengali documents are encompassed limited information compare to the English document.

Initially, we make cluster on the both side of comparable document with the help of TE method. The TE system provides entailment score by comparing every sentence of the document to other sentences within the same document. Thus, $n\,(n–1)$ comparisons have been occurred. The TE system is operated on monolingual data. A cut-off entailment score has been considered for grouping entailed sentences to be a member of the same cluster. The TE system divides the complete set of comparable resources list into some smaller sets of clusters. Each cluster contains at least two sentences. Since, TE system is served on monolingual data, The English clusters are not exactly one to one correspondence with comparable Bengali clusters. To established cross-lingual link between the English and Bengali clusters, we use a probabilistic bilingual lexicon. The lexicon has been prepared by using statistical word alignment tool, in this case, we have used GIZA++ word alignment tool. We trained bilingual English-Bengali parallel corpus of tourism domain with GIZA++, which produce a probabilistic bilingual word alignment list. Five most probable target words with respect to the source word have been considered to retain in the bilingual lexicon.

4 Textual Entailment and Distributional Semantic Similarity

Our system called TESim has shown in Figure 1. TESim system contains semantic textual similarity and textual entailment modules.

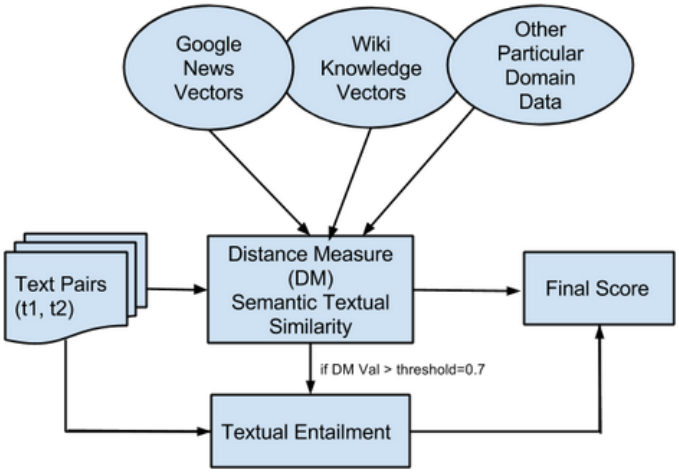


Fig. 1. System Architecture

We have used already pre-trained word and phrase vectors available as part of Google News dataset (Mikolov et al., 2013) (about 100 billion words). The LSA word-vector mappings model contains 300-dimensional vectors for 3 million words and phrases.

Wikipedia is an online encyclopedia that contains millions of articles on a wide variety of topics with quality comparable to that of traditional encyclopedias. We built word and phrase vectors from Wikipedia articles and other crawled corpora from web both Bengali and English. In this experiment, we generated 300 dimensional word and phrase vectors by word2vec[2] tool from Wikipedia articles. Gensim (Rehurek and Sojka, 2010) is a Python framework for vector space modeling.

We used Gensim[3] for this experiment, and computed the cosine distance between vectors representing text chunks. We have experimented on Sematic Textual Similarity data from SemEval to build TESim System.

To see why we use both Semantic Textual Similarity and Textual Entailment, consider an example:

(1) **Text 1 (t1)**: A brown dog is attacking another animal in front of the man in pants.
Text 2 (t2): A brown dog is helping another animal in front of the man in pants.

[2] https://code.google.com/p/word2vec/
[3] https://radimrehurek.com/gensim/

Semantic Textual Similarity Score: 0.95 (by using Wiki Vector and Google News Vector and Cosine Distance)
Textual Entailment Decision: Not entailment

Semantic Textual Similarity system has given high score of 0.95 but the meaning of text 1 and text 2 is very different. In this case, dependency structure weights verbs as main decision factor to solve the problem. For those pair got high score i.e. above threshold value 0.7 by Semantic Textual Similarity, we have checked that pair by using Textual Entailment and then system provided final score.

5 Automatic Alignment Technique of Parallel Fragment of Texts

We extracted bilingual phrase from comparable sentence using template based phrase extraction method (see Section 5.1). Although the template based approach works well in case of parallel corpus, it can also be applied to the comparable corpus. In this case, template based method can only able to align atomic translation (see Section 5.1). Other phrases extracted by template-based method are aligned by baseline PB-SMT system (see Section 7) that is trained on the tourism domain parallel corpus. The English phrases are translated into Bengali using English-Bengali baseline PB-SMT system that we have already developed. This is the same machine translation system whose performance we want to improve. We have also analyzed the same to the other direction, i.e., Bengali-English.

5.1 Template-Based Phrase Extraction

We extract phrase pairs based on the work described in Cicekli and Guvenir (2001). They automatically extract translation templates from sentence-aligned bilingual text by observing the similarities and differences between two example pairs. Their approach produces two types of translation templates i.e. generalized and atomic translation templates. A generalized translation template replaces the similar or differing sequences with variables while an atomic translation template does not contain any variable. We extract the atomic translation template as an additional phrase pair for our Hybrid MT system. Consider the following two English–Bengali translation pairs from the tourism domain data:

(2) a. visitors feel happiness: *darsakera ananda onuvab kore*
b. visitors feel restlessness : *darsakera klanti onuvab kore*

These two examples share the word sequence "*visitors feel"* and differ in the word sequence happiness and restlessness on the source side. Similarly, on the target side, the differing fragments are *"ananda"* and *"klanti"*. Based on these differing fragments, we extract the following sub-sentential phrase pairs in (3).

(3) a. happiness : ananda b. restlessness : klanti

We apply this process recursively to extract sub-sentential phrase pairs when more than one differing sequence is present in between a pair of sentences. The details of the algorithm can be found in Cicekli and Guvenir (2001).

This particular approach has a cubic runtime complexity with respect to the number of sentences in the bilingual corpus. This takes significant amount of time to extract phrase pairs even from a small corpus. Therefore, we used the heuristics to reduce the time complexity. We grouped the entire corpus into *n* clusters based on the sentence similarity such that similar sentences belong to the same cluster. We extract atomic translations from each of these clusters.

6 Dataset

In our experiment, we used an English-Bengali parallel corpus containing 23,492 parallel sentences comprising of 488,026 word tokens from the travel and tourism domain. We randomly selected 500 sentences each for the development set and the test set from the initial parallel corpus. The rest of the sentences were used as the training corpus. The training corpus was filtered with the maximum allowable sentence length of 100 words and sentence length ratio of 1:2 (either way). The corpus has been collected from the "*Development of English to Indian Languages Machine Translation (EILMT) System"* project funded by the Department of Electronics and Information Technology (DEITY), Ministry of Communications and Information Technology (MCIT), Government of India.

7 System Setup

The effectiveness of the present work is demonstrated by using the standard log-linear PB-SMT model as our baseline system. For building baseline PB-SMT system, we use the maximum phrase length of 7 and a 5-gram language model. The other experimental settings were GIZA++ implementation of IBM word alignment model 4 with grow-diagonal-final-and heuristics for performing word alignment and phrase-extraction (Koehn et al., 2003). The reordering model was trained msd-bidirectional (i.e. using both forward and backward models) and conditioned on both source and target languages. The reordering model is built by calculating the probabilities of the phrase pair associated with the given orientation such as monotone (m), swap(s) and discontinuous (d). We use Minimum Error Rate Training (MERT) (Och, 2003) on a held-out development set of 500 sentences, and a target language model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with SRILM (Stolcke, 2002).

8 Experiments and Results

Our experiments have been carried out in two directions. First, we improved the baseline model using the aligned sentiment phrases. Then, we automatically post-edited the translation output by using the sentiment knowledge of the source input test sentence.

The evaluation results are reported in Table 1. The evaluation was carried out using well-known automatic MT evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2006).

Table 1. Statistics of the Comparable Corpus

	Total (English)	Total (Bengali)
Extraction from Comparable corpora	579037 Sentences	169978 Sentences
Extracted comparable sentence pairs using TE	4613 Sentences	4613 Sentences
Aligned parallel phrases	5937 Phrases	5937 Phrases

The collected comparable corpus consisted of 6825 English-Bengali document pairs. It is evident from Table 1 that English documents are more informative than the Bengali documents, as the number of sentences in English documents is much higher than the number of sentences in the Bengali documents. The TE system was able to establish cross-lingual entailment for 4,613 English-Bengali comparable sentence pairs. When the Bengali phrases were passed to the Bengali-English translation module, some of them could not be translated into English and some of them could be translated only partially. Therefore, some of the tokens were translated while some were not. Those partially translated phrases were discarded. Manual inspection of the parallel list revealed that most of the aligned texts were of good quality.

Table 2. Evaluation Result

Exp. No.	Experiments	BLEU	NIST	METEOR	TER
1	Baseline (B)	10.92	4.16	0.3073	75.34
2	B + Extracted Parallel Phrase	13.83	4.44	0.3311	72.33

Table 2 shows the performance of the PB-SMT systems built on the initial training corpus and the larger training corpus containing parallel phrases extracted from the comparable corpora. In the second experiment, the extracted parallel phrases are incorporated with the existing baseline phrase table and the resulting model performs better than the baseline system.

Treating the parallel phrases extracted from the comparable corpora as additional training material results in significant improvement in terms of BLEU (2.92 points, 26.64% relative) over the baseline system. Similar improvements are also obtained for the other metrics.

9 Conclusion and Future Works

In this paper, we successfully presented how textual entailment can help to extract parallel phrases from comparable corpora. For low-resource language pairs, this approach can help to improve the quality of the MT system. The low evaluation scores could be attributed to the fact that Bengali is a morphologically rich language and has a relatively free phrase order; besides there was only one set of reference translations for the test set. Manual inspection of a subset of the output revealed that the additional training examples extracted from comparable corpora effectively resulted in better lexical choice and less out-of-vocabulary compared to the baseline PB-SMT output

In the future, we would like to explore the parallel phrase extraction technique from comparable corpora by combining TE System with hybrid word alignments or hybrid MT method. We will also integrate the knowledge about parallel phrases into the word alignment models as well as with in the MT workflow; this is another future direction for this work. We would also investigate into whether this approach can bring improvements of similar magnitude for larger training data. Finally, richer text representations is yet another direction of our future work (Alonso-Rorís et al., 2014; Das et al., 2014, Sidorov, 2014).

Acknowledgements. The third author acknowledges support from Mexican Government through the Instituto Politécnico Nacional (SIP 20150028, COFAA-IPN) and CONACYT-DST India grant 122030. The research leading to these results has received funding from the EU project EXPERT –the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013<tel:2007-2013>/ under REA grant agreement no. [317471]

References

1. Agirre, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirrea, A., Guof, W., Mihalcea, R., Rigau, G., Wiebeg, J.: Semeval-2014 task 10: Multilingual semantic textual similarity. In: Proceedings of SemEval 2014, p. 81 (2014)
2. Alonso-Rorís, V.M., Gago, J.M.S., Rodríguez, R.P., Costa, C.R., Carballa, M.A.G., Rifón, L.A.: Information Extraction in Semantic, Highly-Structured, and Semi-Structured Web Sources. Polibits 49, 69–75 (2014)
3. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
4. Chaney, A.J., Blei, D.M.: Visualizing topic models. In: International AAAI Conference on Social Media and Weblogs. Department of Computer Science, Princeton University Princeton, NJ, USA (March 2012)
5. Chiao, Y.-C., Zweigenbaum, P.: Looking for candidate translational equivalents in specialized, comparable corpora. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 2, pp. 1–5. Association for Computational Linguistics (2002)

6. Dagan, I., Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. In: Proceedings of PASCAL Workshop on Learning Methods for Text Understanding and Mining, p. 6. Grenoble (2004)
7. Das, N., Ghosh, S., Gonçalves, T., Quaresma, P.: Comparison of Different Graph Distance Metrics for Semantic Text Based Classification. Polibits 49, 51–57 (2014)
8. Déjean, H., Gaussier, É., Sadat, F.: Bilingual terminology extraction: an approach based on a multilingual thesaurus applicable to comparable corpora. In: Proceedings of the 19th International Conference on Computational Linguistics COLING, pp. 218–224 (2002)
9. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)
10. Fung, P., McKeown, K.: Finding terminology translations from non-parallel corpora. In: Proceedings of the 5th Annual Workshop on Very Large Corpora, pp. 192–202 (1997)
11. Fung, P., Yee, L.Y.: An IR approach for translating new words from nonparallel, comparable texts. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 414–420. Association for Computational Linguistics (1998)
12. Gupta, R., Pal, S., Bandyopadhyay, S.: Improving MT System Using Extracted Parallel Fragments of Text from Comparable Corpora. In: Proceedings of 6th Workshop of Building and Using Comparable Corpora (BUCC), pp. 69–76. ACL, Sofia (2013)
13. Cicekli, I., Guvenir, H.A.: Learning Translation Templates from Bilingual Translation Examples. Applied Intelligence 15(1), 57–76 (2001)
14. Kneser, R., Ney, H.: Improved backing-off for n-gram language modeling. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. I, pp. 181–184 (1995)
15. Koehn, P., Och, F.J., Marcu, D.: 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, pp. 48–54. Association for Computational Linguistics (2003)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distribute Representations of Words and Phrases and their Compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
17. Otero, P.G.: Learning bilingual lexicons from comparable english and spanish corpora. In: Proceedings of MT Summit xI, pp. 191–198 (2007)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
19. Pakray, P., Sojka, P.: An Architecture for Scientific Document Retrieval Using Textual and Math Entailment Modules. In: RASLAN 2014: Recent Advances in Slavonic Natural Language Processing, Karlova Studánka, Czech Republic, December 5-7 (2014)
20. Rapp, R.: Automatic identification of word translations from unrelated English and German corpora. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 519–526. Association for Computational Linguistics (1999)
21. Rehurek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of LREC 2010 Workshop New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta (2010)

22. Pal, S., Pakray, P., Naskar, S.K.: Automatic Building and Using Parallel Resources for SMT from Comparable Corpora. In: Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra) @ EACL 2014, April 27, pp. 47–56. ssociation for Computational Linguistics, Gothenburg (2014)
23. Saralegui, X., San Vicente, I., Gurrutxaga, A.: Automatic generation of bilingual lexicons from comparable corpora in a popular science domain. In: LREC 2008 Workshop on Building and Using Comparable Corpora (2008)
24. Sidorov, G.: Should syntactic n-grams contain names of syntactic relations? International Journal of Computational Linguistics and Applications 5(1), 139–158 (2014)
25. Smith, J.R., Quirk, C., Toutanova, K.: Extracting parallel sentence from comparable corpora using document level alignment. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 403–411. Association for Computational Linguistics (2010)
26. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp. 223–231 (2006)
27. Stolcke, A.: SRILM-an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904 (2002)

Statistical Machine Translation from and into Morphologically Rich and Low Resourced Languages

Randil Pushpananda[1], Ruvan Weerasinghe[1], and Mahesan Niranjan[2]

[1] Language Technology Research Laboratory,
University of Colombo School of Computing, Sri Lanka
{rpn,arw}@ucsc.lk
[2] School of Electronics and Computer Science,
University of Southampton, Highfield, Southampton SO17 1BJ, UK
M.Niranjan@Southampton.ac.uk

Abstract. In this paper, we consider the challenging problem of automatic machine translation between a language pair which is both morphologically rich and low resourced: Sinhala and Tamil. We build a phrase based Statistical Machine Translation (SMT) system and attempt to enhance it by unsupervised morphological analysis. When translating across this pair of languages, morphological changes result in large numbers of out-of-vocabulary (OOV) terms between training and test sets leading to reduced BLEU scores in evaluation. This early work shows that unsupervised morphological analysis using the Morfessor algorithm, extracting morpheme-like units is able to significantly reduce the OOV problem and help in improved translation.

1 Introduction

Recent developments in machine translation are dominated by statistical and machine learning methodologies [1] over rule based approaches [2]. SMT relies on the availability of large corpora of parallel text in the source and target languages. The success of practical machine translation systems such as *Google Translate*[1] and similar systems is somewhat restricted to European languages and Chinese and Arabic in which such large collections of data are available [3]. An additional challenge faced by SMT comes from morphological richness in either the source or target language, or both [4–6]. Morphological modifications amplify the effective vocabulary size at the word and phrase levels resulting in an increased size of corpus needed to estimate their statistics reliably. The challenge is most pronounced when both the source and target languages are morphologically rich, and are minority languages in the sense that there are no readily available large corpora with which SMT systems may be trained.

In this paper, we consider one such language pair, Sinhala and Tamil, the national languages of Sri Lanka. Sinhala is spoken almost exclusively in Sri

[1] https://translate.google.com

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 545–556, 2015.
DOI: 10.1007/978-3-319-18111-0_41

Lanka, while Tamil is found (on a much larger scale) in India as well. Sinhala largely belongs to the Indo-European family of languages while Tamil is from the Dravidian family, mostly found in Southern India. Both are morphologically rich. There are 110 noun word forms and up to 240 verb word forms in Sinhala [7] and about 40 noun forms and up to 240 verb forms in Tamil [8]

Another issue to consider is that, written Tamil and colloquial Tamil differ, and this difference is much more pronounced in the usage of this language in India than in Sri Lanka. This causes particular problems in acquiring parallel corpora to translate between Sinhala and Tamil, a point also noted in [9]. Thus, though some development in natural language processing tools, such as part of speech taggers and morphological analyzers for the Tamil language have been developed in Indian research institutions [10, 11], these are not readily applicable on Sinhala-Tamil parallel corpora that we have collected. Hence we resort to an unsupervised learning approach (see Section 3, Methodology).

There is some early research in SMT between Sinhala and Tamil. One of us [12], showed that a Sinhala-Tamil machine translation is easier than Sinhala-English, and attributed the difference to closer relationships between Sinhala and Tamil due to co-evolution between them that has taken place in Sri Lanka. In that research, 4064 Sinhala and Tamil parallel sentences were used as the training data and 167 Tamil sentences were used for testing. The best BLEU score achieved for that Tamil-Sinhala translation was 13.62% while for English-Sinhala translation it was just 6.18%. In our earlier work [13], we quantified limitations of phrase based statistical translation between Sinhala and Tamil. In particular, we explored the increase in translation accuracy as a function of increasing corpus size. As we expected OOV (unique word) rate is reduced by 8% - 10%, when the parallel dataset size was increased from 5000 to 25000 parallel sentences. However, the error analysis showed that most of the untranslated words were inflections and derivatives. Further, in undergraduate work, attempts were made to translate between Sinhala and Tamil using kernel ridge regression [14], a machine learning method using small corpora of up to 3000 parallel sentences [15, 16]. In this formulation, the mapping between phrases is formulated as a regression with the context in which they appear in a sentence being inputs to the kernel model.

A particular advantage of this language pair is that the primary syntactic structures of both languages are of the Subject-Object-Verb (SOV) form. However their grammars allow substantial differences in word order. We illustrate this in Figure 1. The related English sentences for the given examples of Sinhala and Tamil languages are "We are going to watch a film in a little while", "Altogether there are sixty fruits in that room" and "Asian Development Bank has provided 1250 million rupees" respectively.

Such word / phrase movements have been modeled as regression problems in Ni *et al.* [17] to enhance SMT between grammatically very different languages. We note from the example in Figure 1 that, this may not be a serious issue to consider between Sinhala and Tamil.

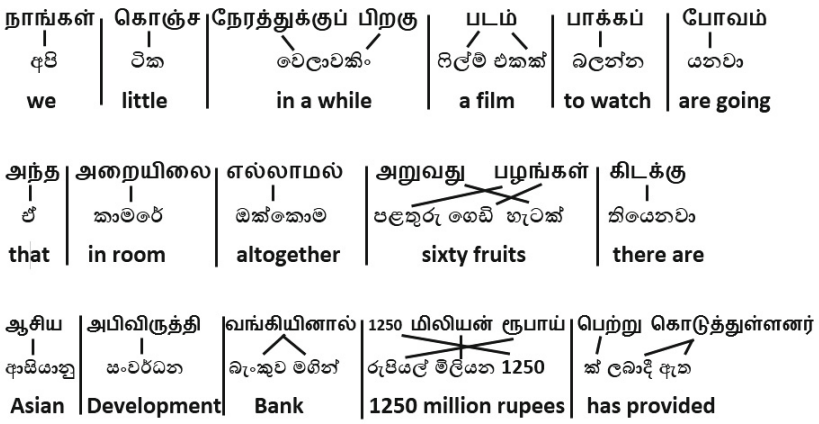


Fig. 1. Example of a pair of Sinhala and Tamil sentences and their relative alignment to the corresponding English sentence taken from the dataset used. Some obvious grammatical errors apparent to readers of Tamil also highlight the challenge of acquiring parallel data for this endeavor.

The use of morphological information integrated into SMT, however, has not been attempted before, both due to a lack of suitable natural language processing tools for Tamil (more specifically, Sri Lankan Tamil) and the datasets used so far being of very limited sizes. The empirical work we report in this paper is a first step in that direction.

Table 1. Examples of some words in common usage in Sinhala with their root (probably) in Tamil

Tamil Word	Sinhala Word	Meaning
அம்மா (/amma:/)	අම්මා (/amma:/)	Mother
அக்கா (/akka:/)	අක්කා (/akka:/)	Elder Sister
பருப்பு (/paruppu/)	පරිප්පු (/parippu/)	Dhal
ஆண்டு (/a:nndu/)	ආණ්ඩු(/a:nndu/)	Government
இடம் (/idam/)	ඉඩම් (/idam/)	Land
வினாடி (/wina:di/)	විනාඩි (/wina:di/)	Minute

In addition there are aspects of Tamil influence on the structure of the Sinhala language. The most significant impact of Tamil on Sinhala has been at the lexical level [18]. Table 1 lists some loan words, of more than a thousand identified as borrowed from Tamil to the Sinhala language [19]. According to [20] people of South India continuously visited Sri Lanka and had close connections with the Sinhala community. Such a close relationship affected the Sinhala language and brought further change to its lexicon.

The remainder of this paper is organized as follows. In Section Two, we give some brief background to statistical translation and morphological analysis that is relevant to this work. In Section Three, we discuss methodological details. Section Four describes experimental evaluations carried out, and we conclude with a discussion of results obtained in Section Five.

2 Background

Translation between two morphologically rich languages is still uncommon. However translating from English to a morphologically rich language and vice versa are widely studied problems. According to the literature, various approaches have been applied for the translation between morphologically rich languages. Most of the researchers have used morphological analyzers and part of speech (POS) taggers to integrate the morphological information to machine translation research.

Popović *et al.* [21] have showed that there are some significant improvements to be achieved by considering morpho-syntactic information even considering only the base forms of the Serbian language when translating from Serbian to English. Also translating from English to Serbian can be improved by removing some of the articles in English. However, it clearly shows that translation in the English to Serbian direction has higher word error rate rather than in the other direction. This proves the difficulty of translating into a morphologically rich language due to the large number of inflections. Similarly, Oflazer *et al.* [22] have investigated phrase based SMT from English to Turkish. They have used lexical morphemes instead of surface morphemes. Results obtained by them show that somewhere in between full word forms and fully morphologically segmented representations provide a significant BLEU score improvement. Nießen and Ney [23] have shown the importance of morphology for scarcely resource languages. Popvic and Ney [24] have proposed two methods to improve the quality of translation from Serbian, Spanish and Catalan languages to English by using POS tags, words stems and suffixes. Their work resulted in a significant reduction of error rates for Serbian, Spanish and Catalan languages. Segalovich [25] showed that an algorithm which can be used to get the morphology of wide lexical coverage using only a limited dictionary.

The above studies investigated translating from English to a morphologically rich language or from a morphologically rich language to English. There are only a limited number of research carried out for translation between two morphologically rich languages without having full morphological analyzers and POS taggers. In such a case an unsupervised morphology learning preprocess to SMT is one of the approaches to be explored. Virpioja *et al.* [26] have applied the Morfessor [27] algorithm to extract the morpheme-like units in an unsupervised manner. They have shown that longer n-grams and longer phrase lengths result in better values for morph-based translations. Even though BLEU score values were slightly lower than the word based approach, it showed promising results for the two morphologically rich languages.

However, there has been no integration of morphology into SMT reported in the literature for the Sinhala-Tamil language pair. Therefore this empirical research is expected to be helpful to identify better morphological approaches to build a successful MT system for translating between two morphologically rich and resource poor languages.

3 Methodology

In this research we have used Morfessor, an unsupervised learning algorithm, to find morpheme-like units of the source and target languages in order to train the language and translation models. Since Morfessor Categories-MAP algorithm [28] has a better segmentation accuracy and handles OOV words in the training data [26], we have used it in our work. Using this approach words have been divided as multiple prefixes followed by stem(s) and multiple suffixes. In rare cases we have found some multiple stems as well.

First we trained the Sinhala and Tamil datasets separately using Morfessor and extracted morpheme-like units as shown in the Figure 2.

සහෝදරියන්ට (To sisters)	සහ/PRE + ෝ/STM + දර/SUF + ි/SUF + ය/SUF + න්/SUF + ට/SUF
කෙටිකාලීනව (short period)	කෙටි/PRE + කා/PRE + ලී/STM + න/SUF + ව/SUF
அகதிகளின் (Refugees)	அ/PRE + க/STM + தி/SUF + களின்/SUF
அகமதாபாத்தில் (In Ahmedabad)	அ/PRE + கம/STM + தா/SUF + பா/SUF + த்/SUF + தில்/SUF

Fig. 2. Examples of unsupervised morphological decomposition

Then we performed three sets of experiments, one with a word based (Baseline system) and two others with two different morphological representations (fully morpheme-like and semi morpheme-like segmentation systems) for the Sinhala-Tamil language pair.

Baseline System. In this experiment, we have used the standard phrase-based SMT modelling approach where words were used as the smallest unit. We have done this experiment to compare performance against the two morph-based approach. A sample parallel sentence from the data is shown below. Here Tamil (TA) sentence length is 7 and Sinhala (SI) sentence length is 10

TA: அவர் கண்ணீர் மல்கிய கண்களுடன் தனது மனைவியை பார்த்துக்கொண்டிருந்தார்
SI: ඔහු කඳුළු පිරුණු දෙනෙතින් යුතුව තම බිරිඳ දෙස බලා සිටියේය

To develop the baseline system, the open source SMT system MOSES [29] was used with GIZA++ [30] using the standard alignment heuristic grow-diag-final

for word alignments. Language models were trained using the Stanford Research Institute language Modeling (SRILM) toolkit [31] with Kneser-Ney smoothing. The systems were tuned using a small extracted parallel dataset (500 sentences) with Minimum Error Rate Training (MERT)[32] and then tested with a randomly extracted test dataset (10% of training data). Finally, the Bilingual Evaluation Understudy (BLEU) [33] evaluation metric was used to evaluate the output produced by the translation system.

Fully Morpheme-Like Segmentation System. In this method morpheme-like units used as the smallest unit and phrase based SMT modelling approach was used similar to the baseline system. As in Figure 2, resulting surface morphemes consist of tags such as Prefixes (PRE), Stems (STM) and Suffixes (SUF). However before training the translation model and the language model, we have removed these tags from the data. Then words in the parallel sentences (training, tuning, testing) and monolingual corpus were replaced with these morpheme-like units. A sample of the split morpheme-like parallel sentence is shown below. Here the Tamil (TA) sentence length is 19 and Sinhala (SI) sentence length is 21

TA: அவர் | கண்ண ீ ர் | ம ல் கி ய | கண் களுடன் | தன து | மனைவி யை | பார்த்த ா க் கொண்டிருந்த ார்
SI: ඔහු | කඳු ළු | පි රු ණු | දෙ නෙ ත ින් | යුතු ව | තම | බි රි ඳ | දෙස | බලා | සිටි යේ ය

Then as mentioned in the baseline system, training, testing and tuning were done. Finally the evaluation was done after performing some post processing. In the post processing stage, the longest matching morpheme-like units were merged to extract readable translated sentences.

Semi Morpheme-Like Segmentation System. In this approach we have combined all the prefixes and stems together and separately merged the suffixes. Similarly, we have built the translation model and language model as before. A sample parallel sentence of this form is shown below. Here the Tamil (TA) sentence length is 12 and Sinhala (SI) sentence length is 15

TA: அவர் | கண்ண ீர் | மல் கிய | கண் களுடன் | தன து | மனைவியை | பார்த்துக்கொண்டி-
ருந்த ார்
SI: ඔහු | කඳු ළු | පිරුණු | දෙනෙ තින් | යුතු ව | තම | බිරි ඳ | දෙස | බලා | සිටියේ ය

Similar to the fully morpheme-like segmentation approach, evaluation was done after post-processing the resulting output. Finally all the results were compared with the baseline system.

4 Experimental Conditions

4.1 Data

We have conducted our experiments for the Sinhala-Tamil language pair. Since Sinhala and Tamil parallel data is limited, we have used two methods to collect the parallel data. The first approach was identifying Sinhala-Tamil parallel documents such as magazines, books, articles, etc. Then we checked the availability of parallel data in electronic format. Most of the documents were available in electronic form but in pdf format not encoded in Unicode. Therefore we had to convert proprietary encoding into Unicode. However we had to align the sentences manually since they were not pre-aligned. Most of the sentences were aligned in a one to many form and some were not aligned at all. Also we have found a large number of figures and tables inside these documents which made the sentence alignment harder.

The second approach to collect parallel data was by translating sentences available electronically in one language to the other with the help of professional translators. We first extracted Sinhala sentences from the *UCSC* [2] *10M words Sinhala Corpus* [34] with word lengths between 8 and 12. Professional translators then translated these Sinhala sentences to Tamil. From both these approaches, we managed to collect 25,500 Sinhala-Tamil parallel sentences. Detailed statistics of the parallel corpus collected are given in Table 2.

Table 2. Characteristics of parallel dataset

Language	Total Words(TW)	Unique Words(UW)	UW/TW
Sinhala	252,101	37,128	15%
Tamil	219,017	53,024	24%

We used the *UCSC 10M words Sinhala Corpus* to build the Sinhala language model. The characteristics of the Sinhala corpus is shown in Table 3.

Table 3. Characteristics of Sinhala Monolingual Corpus

Language	Characteristics		
	Total Words	Unique Words	Sentences
Sinhala	10,142,501	448,651	850,000

4.2 Experiments and Results

As mentioned in the section 3, we have carried out three sets of experiments separately. All the experiments were done in the Tamil to Sinhala translation direction. Fully morpheme-like segmentation and semi morpheme-like segmentation were done repeatedly for three different language models (3-gram, 5-gram

[2] University of Colombo School of Computing.

and 7-gram) without changing the default phrase length. The word-based baseline approach was carried out only for the default settings (i.e. phrase length: 7 and 3-gram language model). Results obtained for the experiments are shown in Table 4.

Table 4. BLEU Score values of the word-based, fully segmented and semi segmented approaches

Description	Word Based	Fully Segmented			Semi Segmented		
	3-gram	3-gram	5-gram	7-gram	3-gram	5-gram	7-gram
BLEU Score (%)	12.99	8.50	12.06	12.53	9.7	10.68	10.29

By comparing the columns in Table 4, we can clearly see that the word-based baseline system gives better BLEU score results overall. However, when we consider the fully-segmented approach, we can see that the BLEU score values increases while increasing the language model size upto 7-gram. According to Table 4, the BLEU improvement rate of the fully-segmented model has been reduced when increasing the language model size from 3 to 7. When considering the semi-segmented approach, results of the 3-gram language model gave better results than the fully-segmented approach. However when the language model size increases upto 7-gram, 7-gram semi-segmented approach resulted in a lower BLEU score value compared to the 5-gram semi-segmented approach.

Since the semi-segmented approach resulted in lower values compared to the fully-segmented approach, further investigations were conducted only using fully-segmented approach. Further investigations were done by changing the maximum phrase length size to 10 in both 5-gram and 7-gram language models. The evaluation results are shown in Table 5.

Table 5. BLEU score (%) values obtained for the experiments done with 5-gram and 7-gram language model and maximum phrase length 10

Description	Fully Segmented (Phrase Length:10)	
	5-gram LM	7-gram LM
BLEU Score(%)	12.15	13.11

Finally the best BLEU score resulted from the fully-segmented approach with language model size 7-gram and maximum phrase length size 10. However, it is not a significant improvement compared to the results of the baseline system. When we consider the translated output, we can rarely see any untranslated words, unlike in the baseline system. Even though the untranslated words are rare, we could achieve only lower BLEU score values for the fully-segmented approach. Visual inspection of the translated output clearly sehows that the first half of sentences have been translated more accurately rather than the second half of sentences. The Table 6 shows the evaluation of the first and second half

of the sentences in 3-gram and 7-gram language models with maximum phrase length 7.

Table 6. Evaluation of first and second half of the sentences resulted in 3-gram and 7-gram language models

Description	BLEU Score(%)	
	3-gram Language Model	7-gram Language Model
Overall	8.50%	12.53%
First Half	9.93%	17.28%
Second Half	3.21%	4.31%

According to Table 6, we can clearly see that the average BLEU score value of the first half of the sentences are much higher than those of the second half of the sentences. Table 7 shows an average word/morpheme-like unit length of the sentences including the maximum and minimum sentence lengths. According to Table 7, we can clearly see that the Tamil morpheme-like sentence length is 3 times larger than the word based sentence length whereas for Sinhala, it is only twice as large.

Table 7. Average and maximum sentence lengths of each word based and morpheme based sentences. Word based approach words used as the smallest unit and morpheme based approach morpheme-like units (MOR) considered as the smallest unit

Description	Sentence Length	
	Words	MOR
Average (Tamil)	10	27
Maximum(Tamil)	23	60
Average (Sinhala)	11	22
Maximum (Sinhala)	27	59

5 Discussion and Conclusions

Experimental results support the suggestion that integrating morphological information into SMT is a way around the data sparseness issue for language pairs that are morphologically rich. However, in the comparisons we make, the BLEU scores are not significantly higher than those of the baseline. In our earlier study [13]), we noted that the traditional word-based approach was unable to translate 25% of words in the test set, and out of this 68% was owing to words in the test set being OOV. This suggests that the rest of the words (32%) are untranslated even when they were present in the training set. Morphological analysis via unsupervised learning was able to reduce this to less than 1% of the total input words, which is a significant result observed in the experiments.

As seen in Table 7, as larger words in the text gets decomposed into smaller morpheme-like units, sentence lengths increase. Since phrase-based systems work better with short word alignments, this length bias introduced by the morph decomposition needs to be improved on. In the experiments we conducted, the longest matching morphemes were merged as words. We will explore ways of correcting the resulting errors by post-processing methods.

Another important observation we made is the influence of errors in the training dataset. In morphologically rich languages, a certain amount of variability in suffixes and merging of words into compounds is tolerated. Writers are often not consistent in the way they use such variations and do not stay within strict grammatical rules of the language. In our database we see this difficulty in sentences that have been translated from a Sinhala original to Tamil by translators. Manually cleaning the training data is important to address this issue.

In future work, we will also concentrate on enhancements to the morph decomposition approach by focusing on suffixes, as in both languages the main morphological modifications are in this part. We believe the decomposition algorithm could be improved to allow supervised segmentation to achieve this.

Acknowledgments. The authors would like to acknowledge the National Research Council (NRC), ICT Agency and LK Domain Registry of Sri Lanka for funding this research. They are also indebted to the research team members of the Language Technology Research Laboratory of the University of Colombo School of Computing, Sri Lanka, for assisting in numerous ways.

References

1. Koehn, P.: Statistical Machine Translation. Cambridge University Press (2009)
2. Chéragui, M.A.: Theoretical Overview of Machine Translation. In: Proceedings ICWIT, p. 160 (2012)
3. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: MT Summit, vol. 5 (2005)
4. Koehn, P., Hoang, H.: Factored Translation Models. In: EMNLP-CoNLL, pp. 868–876 (2007)
5. Goldwater, S., McClosky, D.: Improving Statistical MT through Morphological Analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 676–683. Association for Computational Linguistics (2005)
6. Davis, E.H., Lavie, P.A., Vogel, S.: Integration of Morphology into Statistical Machine Translation (2008)
7. Welgama, V., Herath, D.L., Liyanage, C., Udalamatta, N., Weerasinghe, R., Jayawardana, T.: Towards a Sinhala Wordnet. In: Proceedings of the Conference on Human Language Technology for Development (2011)
8. Lushanthan, S., Weerasinghe, R., Herath, D.: Morphological Analyzer and Generator for Tamil Language. In: Proceedings of the 14th International Conference on Advances in ICT for Emerging Regions, Colombo, Sri Lanka, pp. 190–196 (2014)

9. Germann, U.: Building a Statistical Machine Translation System from Scratch: How much bang for the buck can we expect? In: Proceedings of the Workshop on Data-Driven Methods in Machine Translation, vol. 14, pp. 1–8. Association for Computational Linguistics (2001)
10. Parameshwari, K.: An Implementation of Apertium Morphological Analyzer and Generator for Tamil. An E-Journal of Language in India (2011), http://www.languageinindia.com
11. Anand Kumar, M., Dhanalakshmi, V., Soman, K., Rajendran, S.: A Sequence Labeling Approach to Morphological Analyzer for Tamil Language. IJCSE) International Journal on Computer Science and Engineering 2, 1944–195 (2010)
12. Weerasinghe, R.: A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation. Towards an ICT Enabled Society 136 (2003)
13. Pushpananda, R., Weerasinghe, R., Niranjan, M.: Sinhala-Tamil Machine Translation: Towards better Translation Quality. In: Proceedings of the Australasian Language Technology Association Workshop 2014, Brisbane, Australia, pp. 129–133 (2014)
14. Wang, Z., Shawe-Taylor, J., Szedmak, S.: Kernel Regression Based Machine Translation. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pp. 185–188. Association for Computational Linguistics (2007)
15. Jeyakaran, M.: A Novel Kernel Regression Based Machine Translation System for Sinhala-Tamil Translation. Unpublished BSc Thesis (2011)
16. Sakthithasan, S.: Statistical Machine Translation for Sinhala and Tamil. Unpublished BSc Thesis (2010)
17. Ni, Y., Saunders, C., Szedmak, S., Niranjan, M.: Exploitation of Machine Learning Techniques in Modelling Phrase Movements for Machine Translation. Journal of Machine Learning Research 12, 1–30 (2011)
18. Karunatilaka, W.: Link. Godage International Publishers, Sri Lanka (2011)
19. Coperahewa, S., Arunachalam, S.: A Dictionary of Tamil Word in Sinhala, vol. 2. Godage International Publishers, Sri Lanka (2011)
20. Chandralal, D.: Sinhala, vol. 15. John Benjamins Publishing (2010)
21. Popović, M., Vilar, D., Ney, H., Jovičić, S., Šarić, Z.: Augmenting a Small Parallel Text with Morpho-Syntactic Language Resources for Serbian-English Statistical Machine Translation. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, pp. 41–48. Association for Computational Linguistics (2005)
22. Oflazer, K., El-Kahlout, I.D.: Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, pp. 25–32. Association for Computational Linguistics (2007)
23. Nießen, S., Ney, H.: Statistical Machine Translation with Scarce Resources using Morpho-Syntactic Information. Computational Linguistics 30, 181–204 (2004)
24. Popovic, M., Ney, H.: Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In: LREC (2004)
25. Segalovich, I.: A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In: MLMTA, CiteSeer, pp. 273–280 (2003)
26. Virpioja, S., Väyrynen, J.J., Creutz, M., Sadeniemi, M.: Morphology-Aware Statistical Machine Translation based on Morphs Induced in an Unsupervised Manner. In: Machine Translation Summit XI, pp. 491–498 (2007)

27. Creutz, M., Lagus, K.: Unsupervised Models for Morpheme Segmentation and Morphology Learning. ACM Transactions on Speech and Language Processing (TSLP) 4, 3 (2007)
28. Creutz, M., Lagus, K.: Inducing the Morphological Lexicon of a Natural Language from Unannotated Text (2005)
29. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: MOSES: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. Association for Computational Linguistics (2007)
30. Och, F.J., Ney, H.: The Alignment Template Approach to Statistical Machine Translation. Computational Linguistics 30, 417–449 (2004)
31. Stolcke, A., et al.: SRILM-An Extensible Language Modeling Toolkit. In: INTERSPEECH (2002)
32. Och, F.J.: Minimum Error Rate Training in Statistical Machine Translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 160–167. Association for Computational Linguistics (2003)
33. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
34. Weerasinghe, R., Herath, D., Welgama, V., Medagoda, N., Wasala, A., Jayalatharachchi, E.: UCSC Sinhala Corpus - PAN Localization Project-Phase I (2007)

Adaptive Tuning for Statistical Machine Translation (AdapT)

Mohamed A. Zahran[1] and Ahmed Y. Tawfik[2]

[1] Computer Engineering Department, Cairo University, Egypt
[2] Microsoft Advanced Technology Lab, Cairo, Egypt
moh.a.zahran@gmail.com, atawfik@microsoft.com

Abstract. In statistical machine translation systems, it is a common practice to use one set of weighting parameters in scoring the candidate translations from a source language to a target language. In this paper, we challenge the assumption that only one set of weights is sufficient to pick the best candidate translation for all source language sentences. We propose a new technique that generates a different set of weights for each input sentence. Our technique outperforms the popular tuning algorithm MERT on different datasets using different language pairs.

Keywords: Statistical Machine Translation, Adaptive Tuning, Sentence Representation, Per-sentence Translation.

1 Introduction

Tuning statistical machine translation systems (SMT) is a crucial step that has a significant impact on the overall performance of the system. Tuning is the process of finding optimal weights used to pick the best translation among the generated candidate translations. These weights reflect the relative importance of the SMT building models such as language model, translation model, word penalty, distortion, and any other additional features affecting the quality of translation.

Minimum error rate training (MERT) [4] is the popular tuning algorithm for many statistical machine translation systems. Given a parallel corpus $\{F, E\}$ of source language sentences $F = \{f_1, f_2, f_3 \ldots\}$ and target language sentences $E = \{e_1, e_2, e_3 \ldots\}$, a typical phrasal SMT system undergoes three main steps: training, tuning and testing. The training phase uses the source language and its parallel target language sentences to learn phrase translations and compute translation probabilities to them. These translations from source to target are stored with their probabilities and some additional information in a phrase table. The translation task requires building a language model for the target language to favor the translations obeying the language structure of the target language. The language model can be built with the target language side of the parallel training data, or it can be built using any additional target language text. The tuning phase is concerned with generating candidate

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 557–569, 2015.
DOI: 10.1007/978-3-319-18111-0_42

translations $(e_{i1}, e_{i2}, e_{i3} \ldots)$ for source language sentence f_i and picking the best candidate (e_{i^*}) as the final translation.

$$e_{\mathrm{i}^*} = argmax_{\mathrm{e}}\, S(e,f) \tag{1}$$

The scoring function $S(e,f)$ combines the conditional log likelihood probabilities in the translation model $TM(e|f)$ the language model $LM(e)$ score, a distortion model $D(f,e)$ score, and a word penalty $W(e)$ term. The distortion model controls the amount of reordering of the translated phrases to suite the target language requirements. Word penalty ensures that the translations do not get too long or too short.

$$\begin{aligned} S(e,f) &= \lambda_{\mathrm{LM}} LM(e) + \lambda_{\mathrm{TM}} TM(e|f) + \lambda_{\mathrm{D}} D(f,e) + \lambda_{\mathrm{W}} W(e) \\ &= \lambda \,.\, \Psi(e,f) \end{aligned} \tag{2}$$

The goal of tuning algorithms like MERT is to find a set of optimal weighting parameters $(\lambda_{\mathrm{LM}},\ \lambda_{\mathrm{TM}},\ \lambda_{\mathrm{D}},\ \lambda_{\mathrm{W}})$ to weight the four model listed in (2) to achieve the best translation accuracy measured against a reference translation (ê) using a measure such as the popular BLEU score [5] such that:

$$Maximize\colon\ S(e^*,f)\ if\ e^* is\ most\ similar\ to\ \hat{\mathrm{e}} \tag{3}$$

Practically, $S(e,f)$ can be viewed as the inner product of the weighting parameter vector λ with the model vector Ψ. Let E_λ be the set of translations selected by the model parameterized by the weight vector λ. MERT's goal is finding an optimal weight vector λ^* that minimizes the loss function $L(E_\lambda)$:

$$L(E_\lambda) = 1 - BLEU(E_\lambda) \tag{4}$$

$$\lambda^* = argmin_\lambda\, L(E_\lambda) \tag{5}$$

MERT explores the parameter space using either Powell's method [12] or Koehn coordinate descent as adapted by Moses the statistical machine translation package [8]. MERT finds a series of sub-optimal points (weight vectors) during its search until no new BLEU gain is achieved or the changes in the weights are less than a certain threshold. MERT's objective function is a non-convex piece-wise constant [3]. Which means that at certain critical points, small changes in the weights will change the relative ranking between candidate translations. To visualize these critical points, we will consider two weights only as shown in Figures 1&2, these critical points are on the boundaries of the shaded regions.

If we have two source sentences f_1 and f_2 to translate. We will examine the effect of changing two weights only while holding the rest of weights constant on changing the relative order of the candidate translations of both $f_1$andf_2.

By examining Figure 1, if e_{11} is in fact the best translation for f_1,then the two weights $(\lambda_1,\ \lambda_2)$ should be assigned values in the dotted region to make e_{11} the dominant candidate. Since the same weight values will be used in the translation of all

source sentences in F, there is no guarantee that the best candidate translation of any other sentence will be in the dotted region of f_1 . In other words, if the candidate regions of f_2 is as shown in Figure 2, then one set of values for $\lambda_1$and λ_2 will not translate both f_1 and f_2 optimally, because the dotted regions of f_1 and f_2 do not overlap.

For one set of weights to translate all source sentences optimally, all the dotted regions for all source sentences must overlap, which is not a practical assumption as we explained. In the next sections, we propose new techniques to generate a set of weighting parameters to be used per input source sentence.

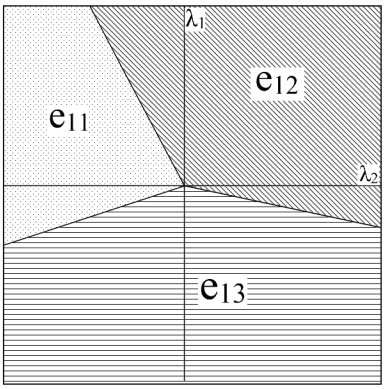


Fig. 1. Changing the relative order between three candidate translations (e_{11}, e_{12}, e_{13}) for the source sentence f_1 with the change of two weights only (λ_1, λ_2). Each region is labeled with its dominant candidate.

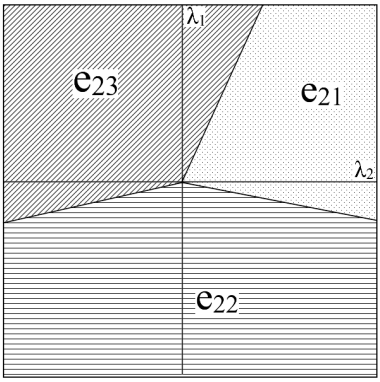


Fig. 2. Changing the relative order between three candidate translations (e_{21}, e_{22}, e_{23}) for the source sentence f_2 with the change of two weights only (λ_1, λ_2). Each region is labeled with its dominant candidate.

2 Related Work

While MERT is used broadly in many SMT systems, no research has been made –to the best of our knowledge- that discusses weight adaptation as presented here. Liu et al. [7] proposed a local training scheme, where the system retunes using a tailored tuning set for the input test sentence. A number of training sentences most similar to the input test sentence are appended to the default tuning set then retuning is performed, and finally the resulting weights are used to decode the test sentence.

Li et al. [6] presented an adaptive data selection, where given a test set, an iterative algorithm will select sentences from the tuning set most similar to the test set and using in tuning weights for this test set. Although our technique AdapT shares the same spirit as these two techniques, unlike them, our sentence specific weights are obtained without re-tuning. This is a major and important difference as re-tuning is a time consuming operation that cannot be done on the fly in real-time. Our methods ensure that decoding happens in real-time and the tuning phase happens only once.

There have been several attempts to enhance upon MERT directly, or enhancing SMT models. Hildebrand et al. [1] developed a method to adapt the translation model for the test set. For each test sentence, the corresponding top n similar sentences are selected from the training data. This selection results in a new subset of the training data used to build a translation model adapted to this particular test set. They represented the sentences as vectors using TF- IDF and used cosine the angle between vectors as the similarity measure between sentences. The amount of data to be selected from the train data per test sentence (n) is determined by minimizing the perplexity (PPL) of the language model built by the selected data.

Hildebrand and Vogel [2] introduced a scheme to leverage the individual strength of different machine translation systems. For a test sentence, they pool the N-best list of all machine translation systems together forming a joint N-best list. The best hypothesis is selected depending on the features scores. These features are based solely on the hypothesis without any prior knowledge of the corresponding machine translation systems. Linear combination weighting between features scores is optimized using MERT. To optimize MERT, Cer et al. [3] presented two alterations to MERT's search techniques. The first is introducing a new simple stochastic search strategy that outperformed Powell's method and coordinate descent. The second is presenting a regularization scheme for both Powell's method and coordinate descent that lead to performance gain.

3 Adaptive Tuning

Using hypothetical examples, we introduced that using one weight vector cannot guarantee choosing the best candidate for all source language sentences, so using different weight vectors in translating different source sentences can –in the best case scenario- translate all source sentences optimally. Optimal translation in this context is choosing the best candidate translation. In this paper, we introduce new methods to generate tailored weight vectors influenced by the input sentence so that the weights change adaptively according to this particular input sentence. Changing the weight vector from one sentence to another reveals the relative importance of the SMT

models (LM, TM, D, W) in translating different sentences. The basic idea is to have a pool of weight vectors preferably diverse enough to explore different relative importance for the SMT models. Then using features from the input sentence we choose the appropriate weight vector from the pool specifically for this sentence. Our technique asks three main questions. First, how to formulate the weights pool. Second, how to represent the input sentence. Third, how to map a sentence to its suitable weight vector. We propose answers to these questions in the next sections.

3.1 Input Representation

The idea is to give similar sentences similar representation. Similarity of sentences can be measured in terms of use of words, topic, length … etc. In the literature term frequency-inverse document frequency (tf-idf) with cosine similarity were used in the context of machine translation to compare sentences. However, tf-idf vectors can be too sparse; instead, we used two techniques to represent input source language sentences. The first is using Latent Semantic Analysis (LSA) [10] which performs SVD over tf-idf bag of words representation for the sentences projecting them into lower dimensionality (200~300 dimensions generally works fine for LSA).

The second representation technique combines semantic projections of individual word representations to form a sentence representation. Several attempts have been made in the literature to generate a continuous vectorized representation (embeddings) for individual words for a given language. Mikolov et al. [11] proposed new technique for learning vectorized representation for individual words called continuous bag of words (CBOW). It is a neural network that predicts a word by using a context window from its history and future. The hidden layer is replaced with a projection layer into which the context words are projected by averaging their vector representations, thus decreasing the computational complexity.

We used the source language side of the parallel training data to learn the CBOW model. Then using the learnt word vectors we represented the source side language sentences of the development set by combining the individual words representation per sentence. A simple combining technique is just adding the words vectors together, but this will not highlight the relative importance of words in the final sentence representation, therefore we used tf-idf weighting as a measure to stress the impact of words over others.

The same data used to learn the CBOW is also used to build the tf-idf model so that the final representation of an N-words sentence is:

$$\sum_{i=1}^{N} \frac{tfidf(w_i) .* vec(w_i)}{\sum_{j=1}^{N} tfidf(w_j)} \tag{6}$$

Where $vec(w_i)$ is the CBOW vector representation for the word w_i of size M. The $tfidf(w_i)$ is a scalar weight for word w_i. The operator $.*$ is an element-wise multiplication operator for each element in the vector. The denominator is a normalizing factor. The result of equation (6) is the vectorized sentence representation of size M.

3.2 Weight Vector Pool

The core idea of our technique is that the impact of the SMT models varies from one sentence to another, these variations are reflected in the choice of weight vector for each sentence. Since eventually the vectors in the pool will be used as reference weights to translate sentences, they should satisfy two conditions: diversity and performance. The first method to fill the pool is via MERT. While MERT's goal is to find a single weight vector that maximizes the BLEU score for the whole development set, it also finds sets of sub-optimal weight vectors along its search for a single global optimum. These sub-optimal vectors can outperform the final optimal vector for individual sentences, which means that we can use these sub-optimal vectors together with the optimal vector in the weights pool. This technique treats the development set as a whole and finds a series of weight vectors performing globally well over the whole development set. We will refer to this technique as "**MERTprogress**".

Another idea is to divide the development set into clusters, and then we run MERT on each cluster so that the final optimal MERT weight vector for each cluster is used in the weights pool. We used kmeans clustering over the development set sentence representations. We will refer to this technique as "**MERTclusters**".

3.3 Mapping Sentences to Weight Vectors

When decoding a new input test sentence, there should be a method to map this sentence to an appropriate weight vector from the pool. Here we present two mapping techniques. The first technique is only applicable to MERTclusters. Using the clusters formed by MERTclusters on the development set sentences, we can uses the test sentence representation to assign it to one of these clusters and uses the assigned cluster's weight vector in its translation.

The second technique is applicable to both MERTprogress and MERTclusters. It is building a regression neural network with objective of mapping the representation of the development set sentences with their corresponding weight vectors. Typical regression neural networks minimize the mean square error (MSE) between outputs and reference values. For the problem at hand, we propose the objective function to maximize the cosine similarity between the predicted weights with the reference weights. The intuition behind this choice for the objective function is as shown in Figures 1 & 2 that the optimal candidate is top ranked when the weights allow the best candidate to dominate (the dotted region). This region extends to infinity, which means that the relative order of the candidate translations is scale invariant with respect to the weight values. Consider equation (2) if we multiply the whole equation by a constant α then the relative ranking of the candidates will remain unchanged.

$$S(e,f) = \alpha\,\lambda\,.\,\Psi(e,f) = \emptyset\,.\,\Psi(e,f) \qquad (7)$$
$$where\ \emptyset\ =\ \alpha\,\lambda$$

This property illustrates that any scaled version of the weight vector will not tamper with the relative order of the candidate translations, which means that it is not important how close in values the predicted weights are from the reference weights as long as they form a scaled version of the reference weights. This directly follows the intuition behind maximizing the cosine similarity between the predicted weights and the reference weights. We used back propagation as a learning algorithm for the neural network. (Check appendix). It is worth noting that minimizing the Cosine error between two normalized vectors is equivalent to minimizing half the square error between them in terms of the objective function.

For MERTclusters, choosing the appropriate weight vector to each sentence in the development set in order to train the neural network is straight forward; the neural network will be trained to map sentences representations with their corresponding cluster weight vector.

On the other hand, for MERTprogress, choosing the best weight vector per sentence requires some calculations. We used all the MERTprogress weight vectors to translate the development set, then for each sentence in the development set we choose which weight vector with the best translation for this particular sentence. The best translation is one with the highest BLEU score. Calculating the BLEU score per sentence usually equals zero. This happens when one of the n-gram scores is zero. To avoid this problem we use equation 8 as a per sentence gauge of translation quality (s) inspired by BLEU.

$$s = BP \sum_{\mathrm{i}=1}^{\mathrm{n}} b_{\mathrm{i}} \tag{8}$$

Where s is the score, BP is the brevity penalty, b_{i} are the i-gram score. When MERT converges after k iterations it finds k points[1] in the weight space one after each iteration. Earlier vectors tend to be associated with lower BP value. Let $P = \{p_1, p_2 \dots p_k\}$ be the set of points ordered by MERT iterations, and $S = \{s_1, s_2 \dots s_k\}$ are the scores computed for the translation of a given sentence using the weights in set P respectively. The set of points that correspond to best scoring translations is:

$$P^* = argmax_{\mathrm{i}}\, s_{\mathrm{i}} \tag{9}$$

If P^* is a singleton, i.e. there is only one point that gives, the best score, then the corresponding point is optimal for this sentence. If not singleton, i.e. if there is a tie, selecting one of the candidates affects the results, and in the experiments we considered two tie resolution mechanisms: the first favors the first or earliest point in P^* (lower index) that produces the best score. The other tie resolution strategy favors the latest point (higher index) in the set P^* that produces the best score. Experiments show, tie resolution, either early or late, affects the performance of the system.

[1] A point in the space is a vector of weights.

4 Phrasal SMT Systems Using AdapT

Using AdapT as tuning algorithm for SMT systems, we will follow the same steps as normal SMT training with few exceptions. The first is the need to build representations for the source language (LSA/CBOW), which can be built using the source language side of the parallel training data or any other source language data. The second difference is the need to have a relatively larger development set, because small development set will not be enough to fit the mapping neural network, and it will not be enough to give neither adequate number of clusters nor representative cluster members.

If the development data is small it will be essential to remove parallel sentences from the training set to be added to the development set, an adequate development set in the order of 10,000 to 20,000 sentences.

We performed a number of experiments to utilize both sentence representation schemes (LSA & CBOW), both weight vectors pooling schemes (MERTprogress & MERTclusters) and finally both mapping schemes (Neural Network & Clustering).

The first experiment using LSA representation with MERTprogress pooling technique and the regression neural network as the mapping scheme (LSA_MERT progress_NN). The second experiment is using the CBOW representations with MERTclusters and clustering to map representation to suitable weight vectors (CBOW_MERTclusters_clustering). The third experiment is using the CBOW representations with MERTclusters and the regression neural network as the mapping scheme (CBOW_MERTclusters_NN). We choose those three experiments to report their results exhaustively. Other possible experiments were carried out using different combinations, but their initial results were redundant. For example, one possible experiment is (CBOW_MERT progress_NN) has almost the same results as (LSA_MERT progress_NN). Testing the SMT system after tuning using one of the pervious settings is shown in Figure 3.

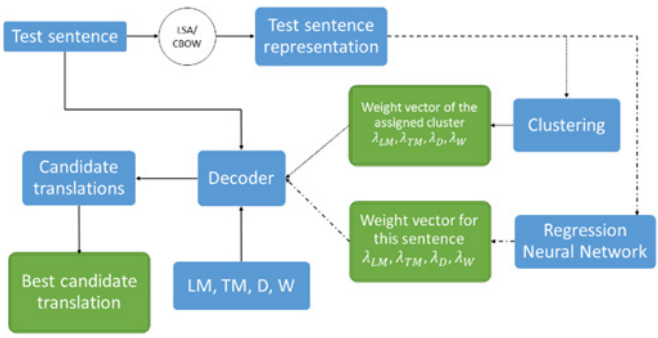


Fig. 3. The decoding of a test sentence using either the clustering or the neural network mapping techniques

5 Results and Evaluation

We applied AdapT in comparison with MERT on different language pairs French-English, Spanish-English, and German-English using different datasets. We used the European Parliament Proceeding Parallel Corpus (europarl-v7) as the training set[2] and used the target side of the parallel data to build the language model and the source side to build the sentence representations for either LSA or CBOW models.

We used the news test 2008 as the development set. Standard development sets are usually small around 3000 sentences, so we removed 15000 sentences from the training set and appended them to the default development set. We build the LSA model using 300 topics to represent the sentence using gensim [9]. On the other hand, CBOW model requires large data to be built accurately, since we intended not to use external data to the WMT datasets, that left us to use the source language side of the training set (around 1.5~2 million lines). This relatively small dataset suggested to use smaller vector size to be learnt by CBOW. Thus, we used vectors of size 200. For MERTclusters pooling technique we used kmeans with k=50 clusters over the development set. Too few clusters will result in big non-homogeneous clusters, which can set back exploiting enough weight vectors variations. On the other hand, too many clusters will result in sparse clusters.

We manually examined the clusters of the development set for the Parliament Spanish-English dataset. Sentences in each cluster share some characteristics as semantic scope, topic and domain (Economics, Energy, Money, Middle-east and conflicts), sentence structure (use of articles, commas, dots and quotations), use of words (named entities, numbers and quantifications), length (short questions and long questions) and type (questions, objections, opening statements, closing statements, question numbers, lists, thank-you sentences, applause and conclusion). Next, in Table 1 we show sample clusters with representative sentences:

Table 1. Sentences and their clusters for the English-Spanish development set

Short Questions:
Why not?
Whose turn is next?
Any comments?
Can you?
Question Numbers:
question No 28 by (H-0781 / 99).
question No 29 by (H-0786 / 99).
Opening statements:
I would like , first of all , to thank the rapporteur for his exceptionally accurate …
Mr President , Commissioners , first of all , I cannot help but reflect upon …
I call upon you , ladies and gentlemen , to vote in favour of this report …
Closing statements:
the debate is closed.
that concludes Question Time.
the vote will take place tomorrow at 12 p.m.

[2] Data available at: `http://www.statmt.org/wmt14/`

Table 1. (*Continued*)

Energy:
we need real cost-effectiveness for our entire energy supply system.
we must reduce CO2 emissions , employ renewable energies , and generally make …
promoting the use of renewables is especially important for the environment.
Conclusion:
Parliament approved the Commission proposal.
the President declared the common position approved (as amended).
Parliament rejected the proposal.
Money:
a special provision of up to EUR 50 million for Greece.
Kyrgyzstan received EUR 17 million.
between EUR 1500 and 2000 billion are traded every day on the financial markets.
Thank-you:
thank you, Mr Poettering.
thank you very much.
thank you, Commissioner, for your statement.
I thank him for that.

Table 2. BLEU scores for MERT and AdapT for different language pairs on different test sets

Test set	MERT	LSA_MERT progress_NN _Early	LSA_MERT progress_NN _Late	CBOW_ MERT clusters_NN	CBOW_MERT clusters_ Clustering
fr-en 2010	22.07	22.31 (+0.24)	22.08 (+0.01)	**22.37 (+0.3)**	22.22 (+0.15)
fr-en 2011	23.07	23.30 (+0.23)	23.10 (+0.03)	23.52 (+0.45)	**23.55 (+0.48)**
es-en 2010	23.73	23.95 (+0.22)	23.97 (+0.24)	**24.25 (+0.52)**	24.05 (+0.32)
es-en 2011	23.25	23.43 (+0.18)	23.54 (+0.29)	23.69 (+0.44)	**23.78 (+0.53)**
de-en 2010	17.00	16.83 (-0.17)	17.25 (+0.25)	17.34 (+0.34)	**17.53 (+0.53)**
de-en 2011	15.82	15.80 (-0.02)	16.17 (+0.35)	16.09 (+0.27)	**16.50 (+0.68)**

Table 2 shows the BLEU score for different data sets. We can notice the loss in BLEU score for the "LSA_MERTprogress_NN_Early" in the last two test sets due to the low BP associated with the "early" configuration. On the other hand, "CBOW_MERTclusters_Clustering" shows the best performance in most of the test sets, which suggests that the SMT systems can leverage information from multiple domains by considering each cluster as a separate domain and classifying the new test sentence to one of the domains. Table 3 shows the statistical significance for our results against MERT using bootstrap resampling techniques [13]. The test aims to estimate the degree to which the true translations quality lies within a certain confidence interval (q) around the measurement on test sets. The commonly used confidence interval is 95%. For a test set of m BLEU score, this test finds an interval $[m-d, m+d]$ in which the true BLEU score lies with probability $q = 0.95$. This test shows that AdapT performs better than MERT with a 95% confidence, and indicates that they are two independent systems as evidenced by the P-value.

Table 3. Statistical significance results for AdapT in comparison to MERT against different datasets at $q = 0.95$, subsampling size equals to the whole test set, and repeating the subsampling for 1000 times. The P-value is the probability that both MERT and AdapT translations are generated from the same system.

Test set		LSA_MERTprogress_NN Late	CBOW_MERT clusters_NN	CBOW_MERT clusters_Clustering
fr-en 2010	MERT	21.3645 +/- 0.5976	21.3885+/-0.5756	21.3556+/-0.5976
	AdapT	21.3671 +/- 0.6041	21.6333+/-0.5802	21.4953+/-0.5755
	P-value	0.391	0.013	0.11
fr-en 2011	MERT	22.1985 +/- 0.5763	22.2038+/-0.5730	22.2299+/0.55699
	AdapT	22.25489+/-0.5754	22.6409+/-0.5881	22.7129+/-0.5706
	P-value	0.168	≈ 0	≈ 0
es-en 2010	MERT	23.1678 +/- 0.6198	23.1409+/-0.5969	23.1546+/-0.6275
	AdapT	23.3848 +/- 0.6238	23.4080+/-0.6140	23.3988+/-0.6010
	P-value	≈ 0	0.01	0.008
es-en 2011	MERT	22.5934 +/-0.5863	22.5888+/-0.5798	22.61975+/0.5818
	AdapT	22.8795 +/- 0.5879	23.0219 +/- 0.577	23.135 +/- 0.5988
	P-value	≈ 0	≈ 0	≈ 0
de-en 2010	MERT	16.3774 +/- 0.5064	16.3504 +/- 0.479	16.3605+/-0.4864
	AdapT	16.5735 +/- 0.494	16.5996+/-0.5202	16.8447+/-0.4822
	P-value	0.001	0.045	≈ 0
de-en 2011	MERT	15.2275 +/- 0.446	15.2069+/-0.4389	15.2263+/-0.4457
	AdapT	15.5275 +/- 0.4473	15.4073+/-0.4965	15.7768+/-0.4525
	P-value	≈ 0	0.068	≈ 0

6 Conclusion

In this paper, we showed the limitations of using one set of weighting parameters in the SMT systems. We presented a number of experiments to choose weight vectors adaptively according to the input sentence. Our preliminary results show that AdapT is a promising approach that has outperformed standard MERT on different language pairs using different datasets. The results of our analysis suggest that there still more room for improvements. We believe that AdapT can influence future research to target the area of adaptive tuning. One possible extension is using AdapT with other tuning algorithm like MIRA or PRO.

References

1. Hildebrand, A., Eck, M., Vogel, S., Waibel, A.: Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In: EAMT: Proceedings of the Tenth, European Association for Machine Translation in Budapest, Hungary, May 30-31, pp. 133–142 (2005)
2. Hildebrand, A., Vogel, S.: Combination of Machine Translation Systems via Hypothesis Selection from Combined N-Best Lists. In: AMTA: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas, Hawaii, pp. 254–261 (October 2008)

3. Cer, D., Jurafsky, D., Manning, C.: Regularization and Search for Minimum Error Rate Training. In: WMT: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, USA, pp. 26–34 (June 2008)
4. Och, F.: Minimum Error Rate Training in Statistical Machine Translation. In: ACL: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 160–167 (2003)
5. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU a Method for Automatic Evaluation of Machine Translation. In: ACL: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 311–318 (July 2002)
6. Li, M., Zhao, Y., Zhang, D., Zhou, M.: Adaptive Development Data Selection for Log-linear Model in Statistical Machine Translation. In: COLING: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, pp. 662–670 (August 2010)
7. Liu, L., Cao, H., Watanabe, T., Zhao, T., Yu, M., Zhu, C.: Locally Training the Log-Linear Model for SMT. In: EMNLP: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, pp. 402–411 (July 2012)
8. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: ACL: Proceedings of the Association for Computational Linguistics Demo and Poster Sessions, pp. 177–180 (2007)
9. Rehurek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: LREC: Proceedings of the Language Resources and Evaluation Conference workshop on new challenges for NLP Frameworks, Valletta, Malta, pp. 45–50 (May 2010)
10. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science, 391–407 (1990)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Distributed Representation of Words and Phrases and their Compositionality. In (NIPS): Proceedings of Neural Information Processing Systems, Nevada, United States (2013)
12. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical recipes 3rd edition: The art of scientific computing. Cambridge University Press (2007)
13. Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation. In: EMNLP: Proceedings of Empirical Methods in Natural Language Processing, pp. 388–395 (2004)

Appendix

The objective function is to maximize the cosine similarity between the predicted vector (y) and the reference vector (d). This is equivalent to:

$$Minimize\ E = 1 - \cos(y, d) = 1 - \frac{y.d}{|y||d|}$$

Let the activation function be $y_i^z = F(x_i^z)$. The superscript denotes the layer number, and the subscript denotes the input number. The notation $l(z)$ refers to the number of neurons in the layer z. The derivative of the error function E with respect to weights at layer z for the training sample m:

$$\frac{\partial E}{\partial w_{\mathrm{IJ}}^{\mathrm{z}}} = \frac{\partial E}{\partial y_{\mathrm{I}}^{\mathrm{z}+1}} \times \frac{\partial y_{I}^{\mathrm{z}+1}}{\partial x_{\mathrm{I}}^{\mathrm{z}+1}} \times \frac{\partial x_{\mathrm{I}}^{\mathrm{z}+1}}{\partial w_{\mathrm{IJ}}^{\mathrm{z}}} = \delta_{\mathrm{I}}^{\mathrm{z}+1} \times \frac{\partial x_{\mathrm{I}}^{\mathrm{z}+1}}{\partial w_{\mathrm{IJ}}^{\mathrm{z}}} \tag{10}$$

$$where, \quad \delta_{\mathrm{I}}^{\mathrm{z}+1} = \frac{\partial E}{\partial y_{\mathrm{I}}^{\mathrm{z}+1}} \times \frac{\partial y_{\mathrm{I}}^{\mathrm{z}+1}}{\partial x_{\mathrm{I}}^{\mathrm{z}+1}} \tag{11}$$

$$\frac{\partial E}{\partial y_{\mathrm{I}}^{z+1}} = \frac{(y.d)y_{\mathrm{I}}^{\mathrm{z}+1} - d_{\mathrm{I}}^{\mathrm{z}+1}|y|^2}{|d||y|^3} \tag{12}$$

$$y_{\mathrm{I}}^{\mathrm{z}+1} = f(x_{\mathrm{I}}^{\mathrm{z}+1})$$

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right)$$

$$\frac{\partial y_{\mathrm{I}}^{\mathrm{z}+1}}{\partial x_{\mathrm{I}}^{\mathrm{z}+1}} = f'(x_{\mathrm{I}}^{\mathrm{z}+1}) = \left(1.7159 \times \frac{2}{3}\right)\left(1 - \left(\frac{y_{\mathrm{I}}^{\mathrm{z}+1}}{1.7159}\right)^2\right) \tag{13}$$

$$x_{\mathrm{I}}^{\mathrm{z}+1} = \sum_{\mathrm{j}=1}^{l(\mathrm{z})} w_{\mathrm{Ij}}^{\mathrm{z}} y_{\mathrm{j}}^{\mathrm{z}} \quad \Rightarrow \quad \frac{\partial x_{\mathrm{I}}^{\mathrm{z}+1}}{\partial w_{\mathrm{IJ}}^{\mathrm{z}}} = y_{\mathrm{J}}^{\mathrm{z}} \tag{14}$$

Finally $\frac{\partial E}{\partial w_{\mathrm{IJ}}^{\mathrm{z}}}$ is calculated by substituting from (11), (12), (13) and (14) in (10).

Now we will prove that optimizing for a training sample for, Cosine error is equivalent for half the square error (SE) if the both the reference (d) and the predicted vector (y) are normalized.

$$|y| = 1 \; and \; |d| = 1$$

$$E_{\mathrm{COS}} = 1 - \cos(y, d) = 1 - y.d$$

$$E_{\mathrm{SE}} = \sum_{\mathrm{i}=\mathrm{i}}^{\mathrm{K}} (d_{\mathrm{i}} - y_{\mathrm{i}})^2 = (d - y) \,.\, (d - y)$$

$$= d.d - d.y - y.d + y.y$$

$$= |d|^2 - 2d.y + |y|^2$$

$$= 2 - 2\cos(y, d)$$

$$\frac{1}{2} E_{\mathrm{SE}} = 1 - \cos(y, d) = E_{\mathrm{COS}}$$

A Hybrid Approach for Word Alignment with Statistical Modeling and Chunker

Jyoti Srivastava and Sudip Sanyal

Indian Institute of Information Technology, Allahabad,
Uttar Pradesh, India
{rs76,ssanyal}@iiita.ac.in

Abstract. This paper presents a hybrid approach to improve word alignment with Statistical Modeling and Chunker for English-Hindi language pair. We first apply the standard word alignment technique to get an approximate alignment. The source and target language sentences are divided into chunks. The approximate word alignment is then used to align the chunks. The aligned chunks are then used to improve the original word alignment.

The statistical model used here is IBM Model 1. CRF Chunker is used to break the English sentences into chunks. A shallow parser is used to break Hindi sentences into chunks. This paper demonstrates an increment in F-measure by approximately 7% and reduction in Alignment Error Rate (AER) by approximately 7% in comparison to the performance of IBM Model 1 for word alignment. Experiments of this paper are based on TDIL corpus of 1000 sentences.

Keywords: Word alignment, Chunk alignment, Natural language processing, Artificial Intelligence.

1 Introduction

Word alignment is the process of identifying correct translation relationships among the words of a bilingual parallel corpus [2,5]. The focus of this paper is on the word alignment task for English-Hindi language pair. Statistical word alignment algorithms usually compute the probability of each word of a source sentence with each word of a target sentence. Based on this probability calculation the algorithm identifies the correct alignment of words. This paper tries to use chunk information of sentences to improve word alignment of a sentence.

Chunks are non-overlapping segment of a text. Generally each chunk encloses a head, with some previous function words and modifiers. Text chunking is the process of dividing a sentence into syntactically associated segment of words. It can be better understood with the following example (taken from TDIL sample tourism corpus):

The best time to visit Bharatpur is during the months of October, November, February and March.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 570–581, 2015.
DOI: 10.1007/978-3-319-18111-0_43

The sentence above can be divided into chunks as follows:
[*The best time*]
[*to visit*]
[*Bharatpur*]
[*is*]
[*during*]
[*the months*]
[*of*]
[*October, November, February and March.*]

The basic hypothesis of this work is that if the sentence will be divided into chunks and those chunks are aligned then by using these aligned chunks, we can improve the performance of word alignment. Thus, we can think of a two phase process. In the first phase a standard word alignment model, like IBM Model 1, can be applied on the parallel corpus to get an initial (and approximate) word alignment. Words aligned with high probability can be extracted from the results of the first phase. In the present work, we refer to these high probability alignments as the dictionary. We now use the same sentence pairs and break them into chunks. The dictionary created at the end of the first phase can then be used for chunk alignment that would generate a list of parallel chunks. We can now improve the initial word alignment by applying a set of rule based filters on these aligned chunks.

IBM Model 1 is a word alignment model that is widely used for working with parallel bilingual corpus [2]. This model was initially developed to provide reasonable parameter estimates for initializing more complex word alignment models like IBM Models 2 - 5. IBM Model 1 works for one word to one word alignment; it cannot solve the problem of fertility and distortion which is solved in higher IBM Models 3-5. All the IBM models are related because all of them use expectation maximization. Output of one IBM Model works as input of other higher IBM Model. For example output of IBM Model 1 works as input for IBM Model 2 and output of IBM Model 2 works as input of IBM Model 3 and so on. So it is expected that if the result of IBM Model 1 improves, then the higher IBM Models will also improve. Thus, we will work first and foremost with IBM Model 1 and any improvement in IBM Model 1 will also get reflected in the higher IBM models.

One well known technique to improve the correctness of word alignment is to enlarge the size of the parallel corpus. However, this is a very expensive process since this involves developing a large parallel corpus. As illustrated in section 5, better performance of word alignment can be achieved with the proposed method by using a comparatively smaller corpus. So, this paper is also an attempt to deal with word alignment for languages with scarce resources.

Section 2 describes some related work. Our approach is described in detail in the section 3. Section 4 gives a brief description of the data that has been used for the experiments. In section 5, we describe the results obtained by the proposed approach and also present their analysis. The last section 6 contains our concluding remarks.

2 Related Work

Word alignment is helpful in many NLP applications. It is a very important step of statistical machine translation [2,5]. Word alignment has been used to extract multiword expressions with semantic meaning [3]. It is also useful in automatic extraction of bilingual lexicon and terminology [14]. Word alignment is also used to transfer language tools developed for one language to other languages. Many NLP applications are enhanced and can improve their performance by using word alignment of better-quality [12]. So finding better word alignment is a central issue to accomplish good quality performance in many NLP applications.

Several word alignment techniques have been proposed. The length of sentences in parallel corpus affects the performance of word alignment. It has been observed that better word alignment quality can be achieved by splitting the longer sentences into shorter ones in the training corpus [21,11]. Thus, different techniques for sentence segmentation have been introduced in some of the research papers in order to improve the performance of word alignment. Xu et. al. [21] introduced a method of sentence segmentation based on modified IBM Model 1. Hutchins and Somers used conjunct and relative clauses for segmentation in a preprocessing step [6]. Kim and Ehara [7] presented a rule-based method for breaking long Japanese sentences in Japanese-to-English translation. Some of the research papers used clauses for sentence segmentation [17,13,16].

There are some research papers which presented word alignment task especially for English-Hindi language pair. Aswani and Gaizauskas [1] used a hybrid approach based on local word grouping, cognates, nearest aligned neighbor, dictionary lookup, transliteration similarity and finally language dependent grammar rules for the alignment of English-Hindi parallel corpus. Venkataramani and Gupta [19] provide a corpus-augmented method of word alignment for English-Hindi with scarce resources. They used two existing word alignment tools: GIZA++ and NATools. Word alignment algorithm gives better results on POS tagged parallel corpus [15].

Sun et.al. [18] proposed a method for the word alignment of English-Chinese corpus based on chunks. This method first identifies the chunks of English sentences. Then chunk boundaries of Chinese sentences are identified by using the translations of English chunks and heuristic information. Thus it proposed a translation relation probability to align words with the resulting chunk aligned bilingual corpus. Watanabe et. al. [20] describe an alternative translation model based on a text chunk under the framework of SMT. The translation model suggested here first performs chunking. Then, each word in a chunk is translated. Finally, translated chunks are reordered. It experimented on a broad-coverage Japanese-English traveling corpus and achieved improved performance. Deng et. al. [4] address the problem of extracting bilingual chunk pairs from parallel text to create training sets for SMT. It discusses a modeling approach which is a first step towards a complete statistical translation model incorporating the alignment of parallel texts. It stated that chunk alignment as a first step in word alignment can significantly reduce word alignment error rate. Most researchers used chunks for reordering to improve the performance of statistical machine

translation but this paper used chunk information at the time of training in word alignment.

Research in NLP for Hindi and the other Indian languages is still in its beginning. It is not easy to find reliable linguistic resources like bilingual parallel corpus, lemmatizer, stemmer, etc. Several educational institutions are working on different projects for these resources. Hence, applying the above methods on large corpus is tricky and thus provides a strong motivation to experiment with techniques that perform well with smaller corpus.

Based on the above literature, we decided to make a hybrid model for word alignment in which a statistical model is the base model and some rule based strategies are applied to improve the performance of the word alignment task with small size of parallel corpus. Thus this paper contributes to the improvement in word alignment for languages where the resources are scarce.

3 Chunk Based Word Alignment

This paper used chunked corpus besides plain parallel corpus to improve the performance of word alignment. CRF chunker[1] is used to break English sentences into chunks. To break Hindi sentences into chunks, a shallow parser[2] is used which is developed by IIIT Hyderabad. The proposed methodology can be defined in two phases as given below:

Phase 1: Use basic word alignment technique (like IBM Model 1) to perform an initial alignment. A bilingual dictionary is created with word pairs that are aligned with high probability.

Phase 2: The sentences are chunked. Then the chunks are aligned using the dictionary. The word alignment is improved by using the aligned chunks.

3.1 Chunk Alignment Based on Initial Dictionary

IBM Model 1 is used as the statistical word alignment algorithm here. IBM Model 1 algorithm extracts initial word alignment by using plain parallel corpus. Experiments were performed on English Hindi language pair and an analysis of word alignment output for different corpus size showed that the word pairs which are aligned with each other with probability more than 0.75 are correct alignments.

The words which have translation probability less than 0.75 are not guaranteed to be correctly aligned. So to extract the dictionary from word alignment output, threshold for translation probability is set to 0.75 because only correct alignment is needed. Any incorrect alignment can lead the alignment task in a wrong direction. Thus source words which have translation probability more than 0.75 with any target word in the word alignment table are extracted. These extracted aligned words are called "dictionary" here. This dictionary is then used for chunk alignment. The process to perform chunk alignment is explained in detail in algorithm 1. If a chunk of a sentence of one language gets aligned to more

[1] http://sourceforge.net/projects/crfchunker/

[2] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/

Algorithm 1. Chunk Alignment

Input: $Plain_{Corpus}$ and $Chunked_{Corpus}$
Output: $dictionary[S_{word}][T_{word}]$, $WordAlignTable[S_{word}][T_{word}][Prob]$, $AlignedChunks[S_{Chunk}][T_{Chunk}]$

```
1: WordAlignTable ⇐ IBMModel1(Plain_Corpus);
2: for i = 0 to WordAlignTable_Length do
3:    if (Prob > 0.75) then
4:       dictionary[][] ⇐ WordAlignTable.get(i);
5:    end if
6: end for
7: for each Sen_pair in Chunked_Corpus do
8:    for each S_Chunk in S_Sen do
9:       for each S_word in S_Chunk do
10:         if dictionary contains S_word then
11:            Extract its T_word from dictionary;
12:            for each T_Chunk in T_Sen do
13:               if T_Chunk contains T_word then
14:                  AlignedChunks[][] ⇐ [S_Chunk][T_Chunk];
15:               end if
16:            end for
17:         end if
18:      end for
19:   end for
20: end for
```

than one chunk of corresponding sentence of another language then we merge all the aligned chunks into one as is the case for first chunk of Hindi sentence given below. Following example presents aligned chunks of an English-Hindi sentence pair. In this research paper the Hindi translation is followed by its transliteration.

English: [Fatehpur Sikri] [is] [an epic] [in] [red sandstone] [.]

Hindi: [फतेहपुर सीकरी लाल बलुआ पत्थर में] [एक महाकाव्य] [है] [।]

Transliteration: [fatehpur sikari laala baluaa patthara men] [eka mahaakaavya] [hai] [.]

Aligned Chunks:

[Fatehpur Sikri] [in] [red sandstone] - [फतेहपुर सीकरी लाल बलुआ पत्थर में] [fatehpur sikari laala baluaa patthara men]

[an epic] - [एक महाकाव्य] [eka mahaakaavya]

[is] - [है] [hai]

[.] - [।] [.]

3.2 Improving Word Alignment Using Aligned Chunks

Chunks, for which we did not get any alignment from algorithm 1, are ignored. Only the list of aligned chunks is used to improve the word alignment. The process to improve word alignment by using chunk alignment can be defined in four steps. These steps are applied on each aligned chunk pair. Some threshold values are used at step 2 and step 3 as well as in Algorithm 2. These values are

selected by analyzing the word alignment output of IBM Model 1. This analysis is described in following steps. These steps are:

Algorithm 2. Improve Word Alignment By Using Aligned Chunks

Input: $Plain_{Corpus}$, $dictionary[S_{word}][T_{word}]$, $WordAlignTable[S_{word}][T_{word}][Prob]$, $AlignedChunks[S_{Chunk}][T_{Chunk}]$
Output: $WordAlignment$

```
for each S_Chunk in AlignedChunks do
  for each S_word in S_Chunk do
    if dictionary contains S_word then
      Extract its T_word from dictionary;
      Remove S_word from S_Chunk;
      Remove extracted T_word from aligned T_Chunk;
    else
      for each T_word in aligned T_Chunk do
        if (Prob(S_word, T_word) in WordAlignTable) > 0.6 then
          dictionary[][] ⇐ [S_word][T_word];
          Remove S_word from S_Chunk;
          Remove T_word from T_Chunk;
        end if
      end for
    end if
  end for
  if S_Chunk contains only one S_word then
    if T_Chunk contains only one T_word then
      if (Prob(S_word, any T_word) in WordAlignTable) < 0.3 then
        dictionary[][] ⇐ [S_word][T_word];
        Remove S_word from S_Chunk;
        Remove T_word from T_Chunk;
      end if
    end if
  end if
end for
Extended_Corpus ⇐ Append(Plain_Corpus, dictionary);
WordAlignment ⇐ IBMModel1(Extended_Corpus);
```

1. Remove the source and target words from aligned chunk pairs which are present in the dictionary as translation of each other.
2. As analyzed by applying IBM Model 1 on different corpus size, if a source word has a probability more than 0.6 with any target word then this alignment might be correct most of the time but not always. So we consider it correct only if it exists in the aligned chunk pairs. Thus if a source word has probability more than 0.6 with any target word and both are present in the aligned chunk pair then we add this word alignment in the dictionary and remove them from the aligned chunks.

3. If after applying the above two steps, any aligned chunk pair (both source and target) have a single word remaining. Then check whether this source word is not having probability more than 0.3 with any target word in the translation table generated by IBM Model 1. If yes then we align those single words of aligned chunk pair. So as above, we remove them from the aligned chunk pairs and put them in the dictionary. Here we set our threshold to 0.3 for this condition because we expect that if a source word has probability less than 0.3 with any target word then it is not close to correct alignment. Thus we ignore those alignments that have an alignment probability less than 0.3.
4. At the last step we append this dictionary to plain parallel corpus and run IBM Model 1 on this extended parallel corpus and get word alignment of better quality.

These steps are explained in Algorithm 2.

4 Data and Evaluation

This approach is trained and tested on TDIL sample tourism corpus[3] for English-Hindi of 1000 sentence pairs which is freely available. The proposed method was trained on 950 sentences of TDIL corpus (English-Hindi). The remaining 50 sentences (5% of the corpus) were used for testing. The performance of the system was measured in terms of F-measure which is defined in terms of precision and recall. This measure has been frequently used in the previous word alignment literature to evaluate word alignment [10]. Och and Ney [12] defined a measure called alignment error rate (AER) which is also used to measure the quality of the word alignment systems.

Alignment A is the set of alignments formed by the alignment model under testing. With a gold standard alignment G, each such alignment set consists of two sets A_S , A_P and, G_S , G_P corresponding to Sure (S) and Probable (P) alignments. Sure (S) alignments are unambiguous alignments and Probable (P) alignments are ambiguous alignments [12]. The performance statistics are defined as

$$(Precision)P_T = \frac{|A_T \bigcap G_T|}{|A_T|} \tag{1}$$

$$(Recall)R_T = \frac{|A_T \bigcap G_T|}{|G_T|} \tag{2}$$

$$(Fmeasure)F_T = \frac{2P_T R_T}{P_T + R_T} \tag{3}$$

$$AER = 1 - \frac{|A_P \bigcap G_S| + |A_P \bigcap G_P|}{|A_P| + |G_S|} \tag{4}$$

Where T is the alignment type, and can be set to either S or P.

[3] `http://tdil-dc.in`

5 Results and Discussion

A comparison is performed between the results of the proposed method and the IBM Model 1 on plain corpus in order to evaluate the effectiveness of chunk alignment for improvement in the performance of word alignment. IBM Model 1 is well formulated by Brown et. al. [2] is described as algorithm by Koehn [8] in his book. The proposed method extends the corpus size by appending some correct word alignment in itself which are extracted by applying word alignment model on the same corpus and with the use of chunk alignment. Thus it is trying to improve itself by using its previous results with IBM Model 1 and chunk alignment.

Table 1. Comparison of results of IBM Model 1 and Chunk-based word alignment

System	Precision (%)	Recall (%)	F-Measure (%)	AER (%)
IBM Model 1	45.98	43.53	44.72	55.28
Chunk-based word alignment	53.93	49.62	51.69	48.31

The results presented in table 1, are obtained by training on 950 sentence pairs and testing on 50 sentence pair. Here, we can see that precision is increased by 8%, recall is increased by 6% approximately. There is approximately 7% improvement in the performance of F-measure and AER.

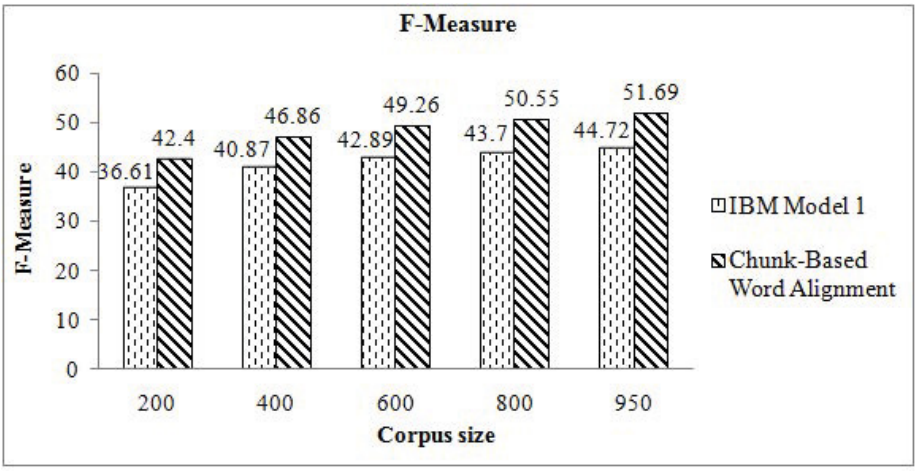


Fig. 1. Performance comparison of IBM Model 1 with the proposed method (Chunk-based Word Alignment) in terms of F-measure

The results given in Figure 1 and 2 are generated for different corpus size from 200 to 950 sentence pairs using the proposed approach and the conventional

IBM Models 1. In Figure 1 and Figure 2, it can be seen that as the corpus size increases, F-measure increases and AER decreases. Obviously it is true and expected for any statistical technique but after some time the increment in corpus size leads to only a marginal improvement in the performance [9]. The accuracy appears to approach a limiting value. So the goal of this work is not only to achieve a better performance even with the small corpus but also to improve that limiting value. This paper achieved its goal by using a hybrid technique which shifted the AER from 55.28% to 48.31% for a corpus size of 950 sentence pairs.

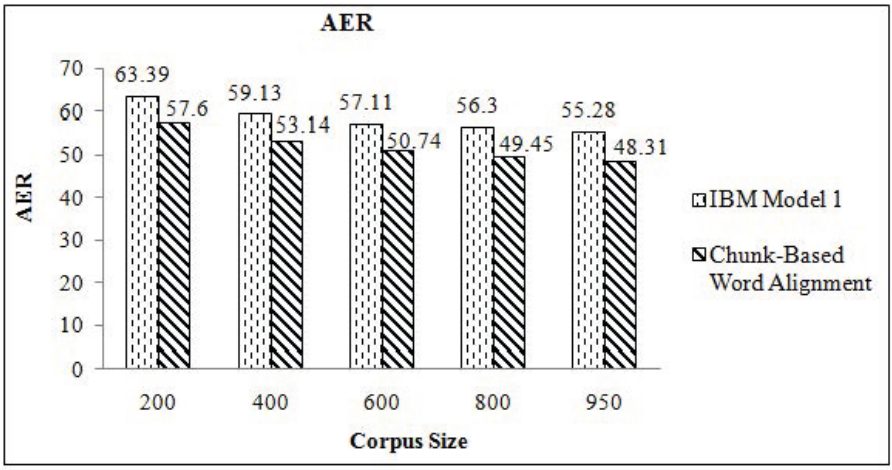


Fig. 2. Performance comparison of IBM Model 1 with the proposed method (Chunk-based Word Alignment) in terms of AER

We tried to append aligned chunk pairs in the extended corpus but that reduced the performance of word alignment. The reason behind this reduction in performance is the inaccuracy of the Chunker which is used to divide English and Hindi sentences into chunks. Obviously, wrong chunking will lead to wrong chunks being appended to the corpus. This can be better understood through the following example:

Chunked English Sentence: [*The best time*] [*to visit*] [*Bharatpur*] [*is*] [*during*] [*the months*] [*of*] [*October, November, February and March*].

Chunked Hindi Sentence: [भरतपुर के] [भ्रमण का] [सर्वोत्तम] [समय] [अक्टूबर,] [नवंबर,] [फरवरी] [और] [मार्च के] [महीनों के दौरान] [है]।

Transliteration: [bharatpur ke] [bhramana ka] [sarvottama] [samaya] [actuubar,] [navambar,] [faravarii] [aur] [march ke] [mahinon ke dauraan] [hai].

In the above example the Hindi sentence is not divided into chunks correctly. Now let us look into the details for chunk alignment. The current chunker of Hindi sentence is giving [सर्वोत्तम] ([sarvottama]) and [समय] ([samaya]) as separate chunks while it should be combined into one chunk [सर्वोत्तम समय] ([sarvottama samaya]). Same is the case for [अक्टूबर,] [नवंबर,] [फरवरी] [और] [मार्च के]

(actuubar,] [navambar,] [faravarii] [aur] [march ke]). So according to algorithm 1 the first chunk of English sentence i.e. "[The best time]" will get aligned to only "[समय]" ([samaya]) because the translation probability of "time" with "[समय]" ([samaya]) is 0.79. The translation probability of "best" with "[सर्वोत्तम]" ([sarvottama]) is less than 0.3. In fact "best" gets aligned with other Hindi words with higher probability. So if we will add the aligned chunk pair i.e. [The best time] and [समय] ([samaya]) in the extended corpus then the probability of "best" will go close to "NULL" which will not be correct. Moreover, one incorrect alignment can lead other alignments in wrong direction too. So we could not append the aligned chunk pairs to the extended corpus.

But if the Hindi chunker will divide the sentence into chunks correctly, then due to the highest alignment probability of "time" with "[समय]" ([samaya]), the chunk [The best time] will get aligned with [सर्वोत्तम समय] ([sarvottama samaya]) and we can append it to the extended corpus. So an accurate chunker for English and Hindi language is needed for improvement in the performance of word alignment.

We also tried to iterate the method four times but in each iteration results were same as in the first iteration. As in the proposed method first chunk alignment is performed and then we focus only on the aligned chunk pairs to improve the performance. So the improvement in the word alignment occurs only to the words which exist in the aligned chunk pairs. In the first iteration all possible improvements are done for the words that exist in aligned chunk pairs. In the next iteration, when we do chunk alignment then the same chunks are aligned as in first iteration because only the words inside those chunks got improved alignment and all possible improvements had already happened. Thus, no further improvements occurred in subsequent iterations.

6 Conclusion and Future Work

This paper proposes a hybrid approach for word alignment of English-Hindi language pair when the resources are scarce. We focused on using the short segments (chunks) of the sentences to improve the performance of word alignment model. This paper verified that it is possible to get better performance of IBM Model 1 in terms of F-measure and AER by about 7%, with the help of chunk alignment. All the conducted experiments provide support that the proposed approach performs better compared to the use of plain corpus with IBM Model 1, for the task of word alignment. This experiment extends our parallel corpus in such a way that it supports itself for better word alignment. As it shows improvement for English-Hindi language pair so it can also be used for other language pairs. The scarceness of the resources suggests that simply using statistical techniques may not be appropriate for word alignment. This paper focuses on developing appropriate word alignment schemes for parallel texts where the corpus is not too large. Even though this paper gives encouraging results for word alignment, it can be improved further by using a better Chunker and higher IBM Models of word alignment. Thus we can expect further improvements in the performance of word alignment.

Acknowledgments. We are thankful to IIIT Allahabad for providing the suitable infrastructure for research. This research has been funded by Tata Consultancy Services (TCS).We are really thankful to Indian Language Technology Proliferation and Deployment Centre Team for providing sample tourism parallel corpus.

References

1. Aswani, N., Gaizauskas, R.: A hybrid approach to align sentences and words in english-hindi parallel corpora. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, ParaText 2005, pp. 57–64. Association for Computational Linguistics, Stroudsburg (2005), http://dl.acm.org/citation.cfm?id=1654449.1654458
2. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics 19(2), 263–311 (1993)
3. Caseli, H., Ramisch, C., Nunes, M.G.V., Villavicencio, A.: Alignment-based extraction of multiword expressions. Language Resources and Evaluation 44(1-2), 59–77 (2010), http://dx.doi.org/10.1007/s10579-009-9097-9
4. Deng, Y., Kumar, S., Byrne, W.: Segmentation and alignment of parallel text for statistical machine translation. Natural Language Engineering 13(3), 235–260 (2007)
5. Gale, W.A., Church, K.W.: Identifying word correspondence in parallel texts. In: Proceedings of the Workshop on Speech and Natural Language, HLT 1991, pp. 152–157. Association for Computational Linguistics, Stroudsburg (1991), http://dx.doi.org/10.3115/112405.112428
6. Hutchins, W., Somers, H.: An introduction to machine translation. Academic Press (1992), http://books.google.co.in/books?id=0ZhrAAAAIAAJ
7. Kim, Y.-B., Ehara, T.: A method for partitioning of long japanese sentences with subject resolution in j/e machine translation. In: Proceedings of International Conference on Computer Processing of Oriental Languages, pp. 467–473 (1994)
8. Koehn, P.: Statistical Machine Translation. Cambridge University Press, New York (2010)
9. Kolachina, P., Cancedda, N., Dymetman, M., Venkatapathy, S.: Prediction of learning curves in machine translation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 22–30. Association for Computational Linguistics, Jeju Island (2012), http://www.aclweb.org/anthology/P12-1003
10. Lambert, P., de Gispert, A., Banchs, R.E., Mario, J.B.: Guidelines for word alignment evaluation and manual alignment. Language Resources and Evaluation 39(4), 267–285 (2005), http://dblp.uni-trier.de/db/journals/lre/lre39.html#LambertGBM05
11. Meng, B., Huang, S., Dai, X., Chen, J.: Segmenting long sentence pairs for statistical machine translation. In: International Conference on Asian Language Processing, IALP 2009, Singapore, pp. 53–58 (December 2009)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003), http://dx.doi.org/10.1162/089120103321337421

13. Ramanathan, A., Bhattacharyya, P., Visweswariah, K., Ladha, K., Gandhe, A.: Clause-based reordering constraints to improve statistical machine translation. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, pp. 1351–1355 (November 2011)
14. Smadja, F., McKeown, K.R., Hatzivassiloglou, V.: Translating collocations for bilingual lexicons: a statistical approach. Computational Linguistics 22(1), 1–38 (1996), http://dl.acm.org/citation.cfm?id=234285.234287
15. Srivastava, J., Sanyal, S.: A hybrid approach for word alignment in english-hindi parallel corpora with scarce resources. In: International Conference on Asian Language Processing (IALP), pp. 185–188 (2012)
16. Srivastava, J., Sanyal, S.: Segmenting long sentence pairs to improve word alignment in english-hindi parallel corpora. In: Isahara, H., Kanzaki, K. (eds.) JapTAL 2012. LNCS, vol. 7614, pp. 97–107. Springer, Heidelberg (2012), http://dblp.uni-trier.de/db/conf/tal/japtal2012.html#SrivastavaS12
17. Sudoh, K., Duh, K., Tsukada, H., Hirao, T., Nagata, M.: Divide and translate: Improving long distance reordering in statistical machine translation. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR, pp. 418–427. Association for Computational Linguistics, Uppsala (2010), http://www.aclweb.org/anthology/W10-1762
18. Sun, L., Jin, Y., Du, L., Sun, Y.: Word alignment of english-chinese bilingual corpus based on chunks. In: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, pp. 110–116. Association for Computational Linguistics (2000)
19. Venkataramani, E., Gupta, D.: English-hindi automatic word alignment with scarce resources. In: Dong, M., Zhou, G., Qi, H., Zhang, M. (eds.) International Conference on Asian Language Processing (IALP), pp. 253–256. IEEE Computer Society (2010), http://dblp.uni-trier.de/db/conf/ialp/ialp2010.html#VenkataramaniG10
20. Watanabe, T., Sumita, E., Okuno, H.G.: Chunk-based statistical translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 303–310. Association for Computational Linguistics (2003)
21. Xu, J., Zens, R., Ney, H.: Sentence segmentation using ibm word alignment model 1. In: In Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation), Budapest, Hungary, pp. 280–287 (May 2005)

Improving Bilingual Search Performance Using Compact Full-Text Indices

Jorge Costa[1], Luís Gomes[1], Gabriel P. Lopes[1], and Luís M.S. Russo[2]

[1] Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, Portugal
jorge.costa@campus.fct.unl.pt, luismsgomes@gmail.com, gpl@fct.unl.pt
[2] INESC-ID / Instituto Superior Técnico - Universidade de Lisboa, Portugal
lsr@kdbio.inesc-id.pt

Abstract. Machine Translation tasks must tackle the ever-increasing sizes of parallel corpora, requiring space and time efficient solutions to support them. Several approaches were developed based on full-text indices, such as suffix arrays, with important time and space achievements. However, for supporting bilingual tasks, the search time efficiency of such indices can be improved using an extra layer for the text alignment. Additionally, their space requirements can be significantly reduced using more compact indices. We propose a search procedure on top of a compact bilingual framework that improves bilingual search response time, while having a space efficient representation of aligned parallel corpora.

1 Introduction

Current statistical machine translation (MT) applications often deal with gigabytes of textual data, as parallel corpora with millions of words. To tackle these ever-increasing space demands, it is necessary to use space and time efficient techniques, to maintain or improve the efficiency of MT tasks. An example of such task is the construction of language and translation models, which are later used for translating a text from one language to another. A language model requires finding monolingual co-occurrences of terms in the same language, while translation models require, among other information, determining bilingual co-occurrence frequencies of two terms in different languages to, in turn, determine translation probabilities and extract translation tables.

Suffix arrays [14] are full-text indices with fast support for string pattern matching and have been used for several MT tasks [18,5,4,19,13], with important time and space results. However, suffix arrays demand up to four times the text size in memory, making them costly for space demanding tasks. Word-based full-text indices [7], with words as units instead of characters, reduced the space requirements of the indices, while maintaining an efficient search time response. Compressed full-text indices reduced such space consumption even further [15], resulting in a slowdown that comes from searching over compressed data, but enabling the indexing of huge amounts of textual data in main memory, without needing disk storage and access.

Full-text indices are ready to answer efficiently to searches over a corpora in one language, but for cross-language applications like determining bilingual

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 582–595, 2015.
DOI: 10.1007/978-3-319-18111-0_44

co-occurrence frequencies, these indices demand an extra layer to represent the text alignment. The alignment is essential to determine the co-occurrences, as two occurrences of a pair of terms in two languages co-occur only if they appear in aligned, or nearby, segments of the aligned parallel corpora. After getting the sets of occurrences of both terms using the indices, the co-occurrences could be determined by iterating over the set of occurrences of the least frequent term of the pair and performing binary searches over the other set. However, the number of occurrences that appear in aligned segments is commonly smaller than the total number of occurrences of the terms in each language. This means that processing all the occurrences of a term may often consume unnecessary time.

We propose a time efficient bilingual search procedure on top of a generic bilingual framework, supported by compact word-based full-text indices and an alignment layer, to determine co-occurrences of word and multi-word terms, and similar results, over aligned parallel corpora. The proposed procedure takes into consideration that in a bilingual search for terms X and Y in different languages, X can co-occur with terms different from Y, as X may have more than one valid translation in the parallel corpora, and jumps over occurrences of X that cannot co-occur with Y, using the alignment information. This strategy improves considerably the bilingual search time response of full-text indices, when compared with the alternative approach based on binary searches, without demanding additional memory consumption. Having the bilingual framework based on compact indices that demand less space than the size of the indexed corpora, enables the support of MT tasks with considerable space requirements in primary memory.

Considering its space and time performance, the full-text bilingual framework can be used for calculating translation models faster, and even on demand, without needing to build the complete translation tables, thus speeding up the machine translation task. The framework is also able to support computer-assisted translation tools, like bilingual concordancing [1], as well as word-sense disambiguation applications and extraction of translation equivalents, by taking advantage of the full-text nature of the indices to extract the context of an occurrence of a term in aligned parallel corpora.

2 Text Alignment and Bilingual Co-occurrences

Parallel text alignment is the task of identifying the corresponding segments between parallel texts. The bilingual framework used in this paper supports parallel text alignment with two granularities: coarse and fine [16,8]. We consider a paragraph or sentence as a coarse segment, while a word or multi-word phrase is considered a fine segment. We denote $coarse(off)$ and $fine(off)$ as the coarse and fine segment respectively, where an occurrence of a term at word-based position off of the text appears.

We consider a monolingual search the task of counting or locating a word or multi-word term in a corpora of the same language. Similarly, a bilingual search determines the bilingual co-occurrences (frequencies or locations) of a

pair of terms X<->Y in aligned parallel corpora. The bilingual co-occurrence frequency is an important factor to determine if X is translated by Y, as when the co-occurrence frequency is high, X and Y have more probability of being correct translations of each other. An occurrence of X co-occurs with an occurrence of Y when the following conditions are met:

Condition 1: $coarse(s_X) = coarse(s_Y)$

Condition 2: $fine(s_X) - d \leq fine(s_Y) \leq fine(e_X) + d$

where d is the maximum distance, in fine segments, for the terms starting at positions s_X and s_Y of the texts, and ending at positions e_X and e_Y, to co-occur.

An occurrence of X co-occurs with an occurrence of Y when both appear nearby in the parallel corpora. In a bilingual environment, the distance between occurrences is measured using the alignment segments. Condition 1 states that the occurrence of X must appear in an aligned coarse segment with the occurrence of Y, otherwise there is no co-occurrence. Within coarse segments, Condition 2 uses the fine segments to determine the distance between the occurrences of X and Y. If both occurrences do not appear within the maximum distance allowed, there is no co-occurrence.

A finely aligned parallel corpora can have segments with one word length, thus an occurrence of a multi-word term will span through more than one fine segment. To tackle such cases, Condition 2 uses the offset where one of the terms in the bilingual search ends to check for a co-occurrence, covering as well situations where s_X is not nearby s_Y (the starting offsets), considering distance d, but e_X (the ending offset of X) and s_Y appear in nearby, or aligned segments. This way, a term may also co-occur with the other term in a parallel corpora if its occurrence ends before the other one starts.

3 Word-Based Full-Text Framework

The full-text bilingual framework is supported by two word-based full-text indices, one for each language, to index the parallel corpora in main memory, and an alignment layer to represent the alignment of the indexed parallel corpora.

3.1 Full-Text Indexing Layer

The bilingual framework can be supported by any word-based full-text index that follows the *Pizza&Chili* API[1]. This API generalizes the basic functionality of word-based full-text indices, consisting in operations such as: $count(ind, t, len)$, to count all the occurrences of term t with word length len in index ind; $locate(ind, t, len)$, to find the word-based offsets where t occurs in the text; and $extract(ind, from, to)$, to obtain the snippet of text from position $from$ until position to.

The word-based full-text framework does not remove any stop-words, to answer accurately to multi-word term searches that may have such words, as "*in*

[1] `http://pizzachili.dcc.uchile.cl/api.html`

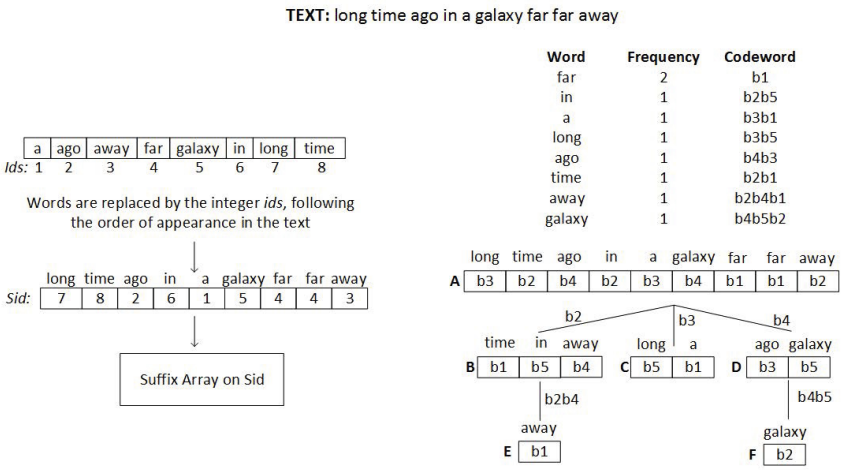


Fig. 1. General Structure of a Word-Based Suffix Array and a Byte-Oriented Wavelet Tree

spite of" or "*is set out below*". In this paper, we experiment the framework and the search procedure using a word-based suffix array [7] and a compressed byte-oriented wavelet tree [3].

Word-Based Suffix Array. A suffix array [14] is a full-text index that represents all the suffixes of a text following a lexicographical order. For a text T indexed in suffix array SA, $SA[i]$ represents the ith substring of T, the suffix of T starting at position i and finishing at the end of T. Suffix arrays have an efficient search time response and may occupy up to four times the text size in memory.

Ferragina and Fischer [7] proposed a word-based suffix array, considering the word as unit instead of the character. The proposal consists on replacing every word in T with an unique integer identifier id. Every id is concatenated following the text order, resulting in an integer sequence sid, with each character (id) referring to a word in the text. Searching for a word or multi-word string pattern starts by decoding every word in the pattern to their integer ids, and then search for the sequence of integers in the suffix array. This proposal led to a considerable gain in space consumption, occupying less than two times the text size in memory, while maintaining a fast query response time. Figure 1 on the left shows the structure of the index.

Byte-Oriented Wavelet Tree. A common form of word compression uses statistical methods to assign a variable-length codeword of bytes to each word in the text, resulting in competitive compression rates. Brisaboa et. al. [3] proposed a reordering of the bytes in the codewords to follow a wavelet tree [9] structure, enabling self-synchronization and providing interesting search time response over the compressed text, for a compressed index, resulting in the byte-oriented wavelet tree. Self-synchronization allows a search to start, or resume, from a given position of the text, instead of always starting from the beginning of

the text. This characteristic is extremely useful to filter the search to particular points of the text, but also to filter the search to certain texts or documents of the indexed corpora.

Figure 1 on the right shows the structure of a byte-oriented wavelet tree, where $b_1 ... b_5$ represent different bytes. Every word in the text has a different codeword, with the more frequent words having smaller codewords to save space. The root node (node A) stores all the first bytes of all the codewords, following the text order. The second bytes are represented in the second level of the tree, while the remaining bytes follow the same procedure for the remaining levels, until all the bytes are represented. The word "*galaxy*" has the codeword $b_4 b_5 b_2$. The root node stores byte b_4. Following the branch b_4, the 2nd byte (b_5) is stored in node D with all the 2nd bytes of the codewords that start with b_4. Continuing through branch $b_4 b_5$, the 3rd byte (b_2) is represented in F.

As all the occurrences of a term are represented in the lower levels of the tree, but the absolute location of the occurrences in the text are codified in the root node, locating a term, or counting its frequency, in the text involves *descending* and *ascending* the tree, using the bytes of the codeword of the term. For multi-word terms, the search is made using the least frequent word of the term and then, for each occurrence of that word, the index checks if the remaining words appear contiguously in the text.

3.2 Alignment Layer

To represent the parallel text alignment, the framework needs an extra layer on top of the text indices. The alignment layer is supported by arrays of integer offsets, two for the $fine$ and two for the $coarse$ alignment, one of each per language. Each position in the arrays stores the word-based offset of the text, in which the alignment segments begin. Considering $fine_{pt}$ the array for a fine alignment of a corpora in Portuguese, $fine_{pt}[0] = 0$, as the first segment starts at the beginning of the text, $fine_{pt}[1] = len(fine_{pt}[0])$, $fine_{pt}[2] = len(fine_{pt}[0]) + len(fine_{pt}[1])$, and so on, with $len(X)$ representing the word length of X. The procedure is similar for the coarse segments.

This layer returns the $coarse$ and $fine$ alignment information, crucial for obtaining the information for the co-occurrence conditions (§2).

4 Bilingual Search Procedure

To determine the bilingual co-occurrences of two terms, a baseline approach would start by obtaining the occurrences of both terms from the indices. Then, iterate over the set of occurrences of the least frequent term, which is the shorter set, and for each one of those offsets, perform binary searches over the longer set of occurrences to find which ones meet Condition 1 and 2, using the alignment layer.

When translating from one language to another, it is common to encounter ambiguous situations. For instance, "*plant*" in English can be translated by "*planta*", "*fábrica*" or "*instalação*" in Portuguese. When determining bilingual co-occurrences for the pair "*plant*"<->"*fábrica*", it would not be uncommon to

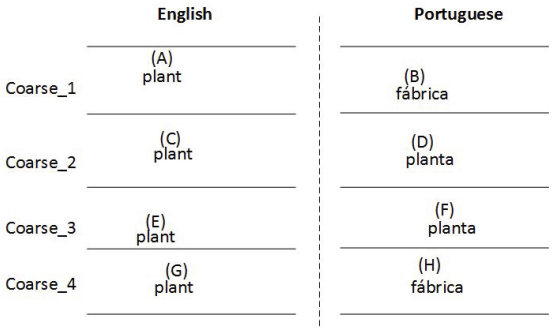


Fig. 2. Occurrences of "*plant*", "*planta*" and "*fábrica*" in aligned parallel corpora

find occurrences of "*plant*" co-occurring with "*planta*" for instance. This means that it may not be necessary to process all the occurrences of "*plant*" or "*fábrica*" to find all the co-occurrences of the pair.

Knowing that two occurrences must appear in aligned coarse segments (Condition 1) and in nearby fine segments (Condition 2) to be a valid bilingual co-occurrence, we propose a different approach for searching for a pair of terms over parallel corpora. Using the information provided by the multi-grain parallel text alignment, the search proposal jumps over occurrences of the terms in the search pair that do not meet the mentioned conditions for co-occurring in the parallel corpora.

Figure 2 shows an abstract representation of four aligned coarse segments of an English-Portuguese parallel corpora, with occurrences of "*plant*", "*planta*" and "*fábrica*" located in several aligned coarse segments from $coarse_1$ to $coarse_4$, in positions (A) to (H). Considering the bilingual search pair "*plant*"<->"*fábrica*", Figure 2 shows two possible co-occurrences in $coarse_1$ and $coarse_4$. The remaining two occurrences of "*plant*" co-occur with a different term and can be jumped over to save time.

4.1 Jump-Based Bilingual Search

Condition 1 is the main basis to support the jump-based bilingual search, thus the operation $coarse(off)$ is essential to determine the jumps. Considering the pair of terms X<->Y, when two occurrences off_X and off_Y, of X and Y respectively, do not appear in aligned coarse segments, $coarse(off_X) \neq coarse(off_Y)$, a jump will take place. The search procedure then selects the term with the occurrence that appears in the coarse segment that meets $min(coarse(off_X), coarse(off_Y))$, and jumps to the first occurrence of the term that appears after the beginning of the coarse segment that meets $max(coarse(off_X), coarse(off_Y))$, the segment where the other term is located.

Considering the example in Figure 2, with X = "*plant*" and Y = "*fábrica*", the search begins with $off_X = (A)$ and $off_Y = (B)$. With these occurrences, Condition 1 is met, as $coarse(off_X) = coarse(off_Y)$. If Condition 2 is also met, then these occurrences are considered a valid bilingual co-occurrence of terms X and Y.

Moving to the next occurrences of both terms would result in $off_X = (C)$ and $off_Y = (H)$. These occurrences appear in non-aligned coarse segments, as (C) is in $coarse_2$ while (H) is in $coarse_4$, thus Condition 1 is not met and there is no bilingual co-occurrence. At this point and considering the text alignment, it is possible to infer two situations: $coarse_2 < coarse_4$ and no occurrence of X will co-occur with an occurrence of Y before $coarse_4$. For that matter, the search can jump to the first occurrence of X appearing after the beginning of $coarse_4$. The jump results in $off_X = (G)$ and $off_Y = (H)$, where $coarse(off_X) = coarse(off_Y)$ and thus Condition 1 is met.

In the example, the jump guided the search procedure to the occurrence in $coarse_4$ as expected. However, the occurrence of "*plant*" might not appear in $coarse_4$, but in a coarse segment after $coarse_4$, leading to the same situation of $coarse(off_X) \neq coarse(off_Y)$. As Condition 1 would not be met, the search would perform another jump, this time with term Y, as $coarse(off_X) > coarse(off_Y)$. This process is repeated until Condition 1 is met or until there are no more occurrences left.

When more than one occurrence of a term appears in the same coarse segment and meet Condition 2, the procedure selects the one which is closest to the other term, with the distance being measured as well in fine segments.

4.2 Jump-Based Search in the Full-Text Framework

In a scenario of parallel corpora with millions of words, the ambiguous situations that cause jumps can occur very often and the procedure can move over a considerable number of occurrences, leading to significant gains in time performance, as demonstrated in the experiments (§5). This procedure does not demand extra space consumption, as it only uses the information reported by the indices and the alignment layer, and it is not applicable to monolingual queries, where all the occurrences need to be reported.

For supporting the jump-based search, the *locate* operation of the Pizza&Chili API (§3) must return one occurrence at a time, mostly like in a *next* operation of an iterator, and take into account the position given by the jump. This lead to a change in the API, replacing the *locate* operation with $\{off, next\} = locate\_next(ind, it, t, jump)$. This operation obtains the occurrence off of term t in index ind, that appears after, or in position $jump$, using an iterator of occurrences it. It also returns the occurrence of t immediately after off, $next$, which is extremely helpful to determine the next jump. If there was no jump, then $jump$ is just the position of the next occurrence of t, $next$, in the following iteration. To initialize the iterator and the search for a term t with word length len, the API also needs the operation $it = start\_locate(ind, t, len)$.

The byte-oriented wavelet tree returns the occurrences following the order they appear in the corpora, so instead of continuing the search to the next occurrence of a term, until there are no more occurrences left, the index uses the position of the jump and its self-synchronous nature, to guide the search to a farther stage of the corpora. With such strategy, this index avoids processing and returning all the occurrences of the terms, as well as additional operations

over the sets of occurrences to determine the co-occurrences. With suffix arrays, the sets of occurrences are returned in a lexicographical order. To support the jumps, the word-based suffix arrays determine all the occurrences of both terms. Then, the sets of occurrences are sorted following the order of appearance in the text. Finally, both sets are transversed following the jump-based procedure, which may avoid processing all the occurrences of the terms.

Algorithm 1 presents the main part of the jump-based search procedure in pseudo-code, using the additional operations: $more\_matches(t_x, t_y)$, to determine if there are more matches of the terms t_x and t_y; $is\_match(off_x, off_y)$, to determine if the occurrences at positions off_x and off_y meet Condition 2; and $coarse\_beg(coarse(off))$, to determine the starting offset of the coarse segment where occurrence off appears.

```
Data: ind_x, ind_y: indices. it_x, it_y: search iterators
next_x = 0; next_y = 0;
while (more_matches(t_x, t_y)) do
    {off_x, next_x}=locate_next(ind_x, it_x, t_x, next_x);
    {off_y, next_y}=locate_next(ind_y, it_y, t_y, next_y);
    while ((coarse(off_x) != coarse(off_y)) & (more_matches(t_x, t_y))) do
        c = max(coarse(off_x), coarse(off_y));
        if (c == coarse(off_x)) then
            jump = max(next_y, coarse_beg(c));
            {off_y, next_y} = locate_next(ind_y, it_y, t_y, jump);
        else
            jump = max(next_x, coarse_beg(c));
            {off_x, next_x} = locate_next(ind_x, it_x, t_x, jump);
        end
    end
    if (is_match(off_x, off_y)) then
        num_matches++ ;
    end
end
```

Algorithm 1. Bilingual search procedure

5 Experimental Results

To perform the experiments, we used four English (EN)-Portuguese (PT) parallel corpora, indexed in main memory using the word-based full-text bilingual framework: 1) EMEA [17] with 196 Megabytes (Mb); 2) Europarl [10] with 394 Mb; 3) DGT[2] with 804 Mb; and 4) Eurlex[3] with 1126 Mb. These sizes consider the parallel corpora in both languages and the information about the alignment.

To measure the query time response of the framework and the proposed bilingual search algorithm, we used 700,000 bilingual lexicon entries in EN-PT.

[2] http://ipsc.jrc.ec.europa.eu/index.php?id=197
[3] http://eur-lex.europa.eu/en/index.htm

Table 1. Memory consumption results

Corpora	Size (Mb)	Tokens (M)	WSA (Mb)			BOWT (Mb)		
			Index	Total	Ratio	Index	Total	Ratio
EMEA	196	8.912 (EN) 10.633 (PT)	250.02	315.12	1.61	86.25	151.35	0.77
Europarl	394	16.052 (EN) 20.715 (PT)	520.76	628.60	1.59	176.96	284.81	0.72
DGT	806	35.614 (EN) 46.127 (PT)	1,062.56	1,305.40	1.62	365.58	606.36	0.75
Eurlex	1,126	47.537 (EN) 62.509 (PT)	1,428.69	1,748.27	1.55	493.91	813.49	0.72

Table 2. Bilingual search time results

Corpora	Nr. Entries	Time WSA (s)			Time BOWT (s)		
		Jump	Base	Imp. Ratio	Jump	Base	Imp. Ratio
EMEA	190,600	30	71	2.37x	275	382	1.39x
Europarl	455,122	87	264	3.03x	1,850	3,042	1.64x
DGT	502,770	169	560	3.31x	8,830	14,628	1.66x
Eurlex	553,279	225	780	3.47x	16,546	27,225	1.65x

Each bilingual lexicon entry consists on two terms that are correct translations in EN-PT, where each term has a variable length of 1 to 8 words. These entries were automatically extracted [8] from the four corpora used, and later validated by human translators and classified as correct, under the three-year research project ISTRION[4], funded by FCT/MEC.

Before the experiments, the 700,000 bilingual entries were filtered for each corpora, maintaining only the subset of entries that have at least one occurrence in the corpora, either in EN or PT. The number of entries used per corpora are shown in Table 2. The experiments were developed using a machine with 16 Gigabytes Memory RAM, a Intel Core I7, 3.4 GHz processor, using Ubuntu Linux Kernel 3.2.0.

5.1 Memory Consumption

Table 1 shows the results for the bilingual framework memory consumption in Mb, using word-based suffix arrays (**WSA**) and byte-oriented wavelet trees (**BOWT**), alongside with the space consumption ratio (**Ratio**) of the framework in comparison with the corpora size. Table 1 also shows the number of tokens of each corpora in millions (**M**), in EN and PT. The memory consumption is not affected by the proposed bilingual search procedure, as the structure of the framework does not change.

The bilingual framework supported by BOWT is the most compact approach, occupying 72% of the text size for Eurlex, due to the compressed nature of the

[4] ref. PTDC/EIA-EIA/114521/2009.

index, which requires 43% of the size of Eurlex to index it in main memory. The approach using WSA is also space efficient, requiring 1.5 to 1.6 of the corpora size. These ratios have a tendency to improve for larger corpora, in particular with the compressed index, due to an increase of word repetition, which is already visible when comparing the index values obtained with EMEA and Eurlex, making the framework ready to support space demanding MT tasks that require corpora with millions of words.

The alignment layer by itself is space demanding, however it does not makes the framework space inefficient. On the other hand, a more compact approach for supporting the alignment could lead to an undesired slowdown.

5.2 Bilingual Search Time Performance

Table 2 presents the bilingual search results for the same indices, using the jump-based proposal (**Jump**) and the baseline approach (**Base**), which consists on iterating over the smaller set of occurrences in the search pair and using a binary search on the larger set of occurrences. Table 2 also shows the number of entries processed per corpora, which are the subsets of entries that have at least one occurrence in the respective corpora, and the improvement ratio (**Imp. Ratio**) obtained with the jump-based approach.

The jump-based search improves the bilingual search time response up to 3.5 times using WSA and 1.7 times using BOWT for Eurlex, when compared with the baseline approach. Table 2 shows that the improvement ratio increases with the size of the corpora, in particular for WSA. When searching for pairs of terms over larger corpora, the cases of ambiguity and the number of occurrences of the terms tend to be higher, thus benefiting the jumps behavior and resulting in more significant time savings.

Using BOWT the framework is considerably slower, when compared with WSA. The compressed structure of BOWT, with all the words codified to save space, demands more time to find and decode a term in the corpora. Even without needing to determine all the occurrences of the terms, which is always necessary with WSA, the restart of the search from a given point is time consuming. Every jump demands extra *descends* and *ascends* that are not normally performed with the index standard search behavior, thus the gains are less significant in comparison with WSA. This slowdown is typical of compressed indices and it depends on the level of compression rate, where less space requirements typically lead to slower time responses. However, considering the index compression ratio, this solution is important to index huge corpora in main memory, without needing secondary storage and access.

Figure 3 shows two graphs that represent the variation of the jump-based bilingual search time performance, using both indices, for the lexicon entries with higher co-occurrence frequency per corpora. We filtered the 10,000, 30,000, 50,000, 70,000, and 100,000 more frequent entries for Europarl, DGT and Eurlex. The most frequent entries are chosen by corpora, thus they may not be the same for the three corpora.

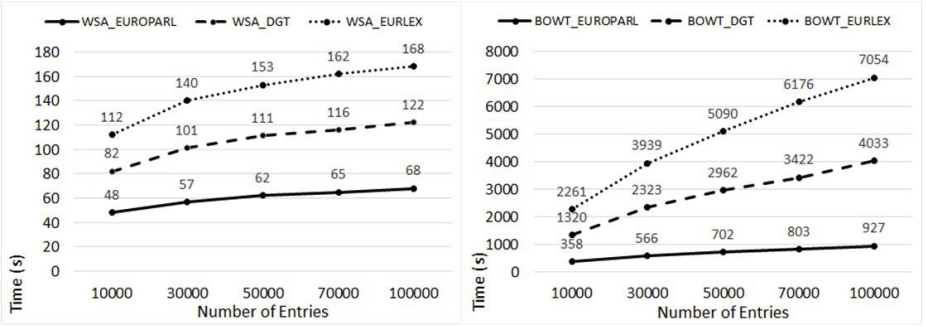


Fig. 3. Bilingual search time variation for answering the 10,000 to 100,000 most frequent entries

The results obtained using WSA demonstrate that answering the 30,000 most frequent entries occupy around 60% of the total time presented in Table 2. Considering Eurlex, 30,000 entries are only 5% of the total of entries processed, showing the weight of these highly frequent pairs of terms to the bilingual search procedure. With BOWT, the 30,000 most frequent entries demand around 25%-30% of the total time showed in Table 2. This way, besides the 30,000 most frequent entries, and even the 100,000 most frequent ones, BOWT also needs a considerable amount of time to answer the remaining entries, due to its slower search procedure that makes even less frequent entries also time demanding. The most frequent entries can be determined in a pre-processing stage and stored in a cache to avoid processing them at runtime, but in this paper we intended to test the performance of the algorithm for the most frequent entries as well.

6 Related Work

Full-text indices have been commonly used in MT tasks, in particular suffix arrays, due to their ability to fully index the text and provide fast search functionalities over the text information. Yamamoto and Church [18] proposed suffix arrays for determining term and document frequency of substrings over large corpora. Callison-Burch et.al. [5] used suffix arrays for retrieving translations of large phrases, with important space and time improvements, and also for indexing and searching translation memories, using one index per language and an array to associate the pairs to their alignments. Zhang et. al. [19] also used suffix arrays to determine phrase-to-phrase alignment models, while Lopez [13] used them to support translation by computing translation rules only as needed, instead of pre-calculating the complete model. Lopez [12] used as well suffix arrays, together with additional structures, to extract hierarchical phrase-based rules.

The full-text framework in this paper can be used as support for concordancing tools, like TransSearch, [1], a context analysis tool of significant importance for human translators, as it allows them to analyze the context of the occurrences of a term or pair of terms in parallel corpora. TransSearch is based on

Lucene[5], a text retrieval framework supported by inverted indices [20]. We opted for a framework based on suffix arrays and full-text indices as such indices are arranged in a way that the frequency of multi-word terms is readily available, instead of having to be computed on the fly, and also contain information of arbitrary sub-strings of words, making them extremely useful in MT.

Brisaboa et. al. [2] proposed a word-based compressed suffix array that demonstrated to be faster than compact inverted indices for compression ratios lower than 40% and for word and multi-word terms without too many occurrences. This index also presents better time performance for phrases with three or more words, when compared with the byte-oriented wavelet tree. On the other hand, being a suffix array, the results are obtained in lexicographical order, demanding the sorting of both set of occurrences of a pair of terms to apply the jumps. Nevertheless, it is an interesting alternative for supporting the more compressed version of the bilingual framework.

7 Conclusions and Future Work

In this paper, we propose a bilingual search procedure on top of a bilingual framework supported by word-based full-text indices, to determine bilingual co-occurrences of two word or multi-word terms over finely aligned parallel corpora, with efficient time and space performance. Such functionality is important for several machine translation tasks, as the translation task itself by supporting the construction of translation models, context analysis tools, among others. The bilingual search procedure jumps over occurrences of a term that cannot co-occur with any occurrence of the other term, using the alignment layer, instead of considering all the occurrences of at least one of the terms to determine the co-occurrences. The jump-based search improves the bilingual search response time for all the indices and corpora used in the experiments, when compared to the latter approach, without increasing the memory requirements of the framework. With a compressed index, the framework may occupy 70% of the text size, thus enabling the support for space demanding tasks in main memory.

The bilingual framework can also be adapted to support hierarchical machine translation, a translation model proposed by Chiang [6], with support for discontiguous language constructions, which improved the translation quality when compared with phrase-based models [11]. A hierarchical phrase has gaps between subphrases that can be replaced by other subphrases, tackling difficult language constructions as the EN-PT pair "*in* X_1 *and* X_2 *alike*" <-> "*tanto em* X_1 *como em* X_2", where "*alike*" is translated by "*tanto em ... como em*", and X_1 and X_2 are gaps. The jump-based search can be adapted to solve the collocation problem described by Lopez [12], without using any additional data structures. For example, the search may not consider occurrences of "*in*" or "*and*" that are not collocated with '*alike*" and vice-versa, saving space by not using auxiliary structures and time by jumping over occurrences of frequent terms.

[5] `http://lucene.apache.org/core/`

Acknowledgments. We thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by FCT/MEC grants, with references SFRH/BD/78390/2011 and SFRH/BD/64371/2009, and by ISTRION project (ref. PTDC/EIA-EIA/114521/2009).

References

1. Bourdaillet, J., Huet, S., Langlais, P., Lapalme, G.: Transsearch: from a bilingual concordancer to a translation finder. Machine Translation 24(3-4), 241–271 (2010)
2. Brisaboa, N.R., Fariña, A., Navarro, G., Places, Á.S., Rodríguez, E.: Self-indexing natural language. In: Amir, A., Turpin, A., Moffat, A. (eds.) SPIRE 2008. LNCS, vol. 5280, pp. 121–132. Springer, Heidelberg (2008)
3. Brisaboa, N.R., Fariña, A., Ladra, S., Navarro, G.: Reorganizing compressed text. In: Proceedings of the 31th Annual International ACM SIGIR Conference, pp. 139–146 (2008)
4. Callison-Burch, C., Bannard, C., Schroeder, J.: A compact data structure for searchable translation memories. In: European Association for Machine Translation (2005)
5. Callison-Burch, C., Bannard, C., Schroeder, J.: Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 255–262 (2005)
6. Chiang, D.: Hierarchical phrase-based translation. Computational Linguistics 33(2) (2007)
7. Ferragina, P., Fischer, J.: Suffix arrays on words. In: Ma, B., Zhang, K. (eds.) CPM 2007. LNCS, vol. 4580, pp. 328–339. Springer, Heidelberg (2007)
8. Gomes, L., Aires, J., Lopes, G.P.: Parallel Texts Alignment. In: New Trends in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence, pp. 513–524 (2009)
9. Grossi, R., Gupta, A., Vitter, J.S.: High-order entropy-compressed text indexes. In: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, pp. 841–850 (2003)
10. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Machine Translation Summit X, vol. 5 (2005)
11. Koehn, P., Och, F.J., Marcu, D.: 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, vol. 1, pp. 48–54 (2003)
12. Lopez, A.: Hierarchical phrase-based translation with suffix arrays. In: Proceedings of the Joint Meeting EMNLP-CoNLL, pp. 976–985 (2007)
13. Lopez, A.: Tera-scale translation models via pattern matching. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 505–512 (2008)
14. Manber, U., Myers, G.: Suffix arrays: a new method for on-line string searches. SIAM Journal on Computing 22(5), 935–948 (1993)
15. Navarro, G., Mäkinen, V.: Compressed full-text indexes. ACM Computing Surveys 39(1) (2007)
16. Ribeiro, A., Dias, G., Lopes, G., Mexia, J.: Cognates alignment. In: Machine Translation Summit VIII, Machine Translation in The Information Age, pp. 287–292 (2001)

17. Tiedemann, J.: News from opus-a collection of multilingual parallel corpora with tools and interfaces. In: Recent Advances in Natural Language Processing, vol. 5, pp. 237–248 (2009)
18. Yamamoto, M., Church, K.W.: Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. Computational Linguistics 27(1), 1–30 (2001)
19. Zhang, Y., Vogel, S.: An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In: Proceedings of the 10th European Association for Machine Translation Conference, pp. 294–301 (2005)
20. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Computing Surveys 38(2), 6 (2006)

Neutralizing the Effect of Translation Shifts on Automatic Machine Translation Evaluation

Marina Fomicheva, Núria Bel, and Iria da Cunha

Institute for Applied Linguistics, Universitat Pompeu Fabra, Barcelona, Spain
{marina.fomicheva,iria.dacunha,nuria.bel}@upf.edu

Abstract. State-of-the-art automatic Machine Translation [MT] evaluation is based on the idea that the closer MT output is to Human Translation [HT], the higher its quality. Thus, automatic evaluation is typically approached by measuring some sort of similarity between machine and human translations. Most widely used evaluation systems calculate similarity at surface level, for example, by computing the number of shared word n-grams. The correlation between automatic and manual evaluation scores at sentence level is still not satisfactory. One of the main reasons is that metrics underscore acceptable candidate translations due to their inability to tackle lexical and syntactic variation between possible translation options. Acceptable differences between candidate and reference translations are frequently due to optional *translation shifts*. It is common practice in HT to paraphrase what could be viewed as close version of the source text in order to adapt it to target language use. When a reference translation contains such changes, using it as the only point of comparison is less informative, as the differences are not indicative of MT errors. To alleviate this problem, we design a paraphrase generation system based on a set of rules that model prototypical optional shifts that may have been applied by human translators. Applying the rules to the available human reference, the system generates additional references in a principled and controlled way. We show how using linguistic rules for the generation of additional references neutralizes the negative effect of optional translation shifts on n-gram-based MT evaluation.

Keywords: Translation shifts, Machine Translation Evaluation, Paraphrase Generation.

1 Introduction[1]

One of the most important observations in the field of translation studies is that a translated text can differ from the original at any linguistic level – lexical, syntactic, discourse – and still be considered perfectly acceptable. The departures from theoretical formal correspondence between source and target language units for the sake of

[1] This work was funded by the UPF-IULA PhD grant program and the FI-DGR grant program of the Generalitat de Catalunya.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 596–607, 2015.
DOI: 10.1007/978-3-319-18111-0_45

textual equivalence are denominated *translation shifts* [1]. It is one of the key concepts in translation theory. Apart from the obvious transformations necessary for grammatical well-formedness, it is common practice in translation to introduce optional changes to the way information is presented in the source text. Although such changes are not strictly necessary, they are part and parcel of Human Translation [HT], as professional translators are expected to adapt the original to the norms and conventions of target language use depending on the text genre, text type, register, means of communication, etc.

The distinctive properties of translated texts have been extensively studied both in the field of translation theory and in computational linguistics. Surprisingly, they have rarely been discussed in the field of automatic Machine Translation [MT] evaluation, although the vast majority of evaluation systems are actually based on the degree of similarity between MT and HT. As similarity is normally calculated at surface level, the performance of the metrics depends on the availability of a heterogeneous set of human reference translations. In practice, however, only one reference is available and its characteristics can strongly affect the results of automatic evaluation. As an illustration, consider Table 1, which shows an example of English-Spanish MT evaluated manually and automatically (back translation to English is given in square brackets and relevant constructions are marked in bold).[2] Manual evaluation is scaled from 1 to 4 and automatic evaluation score is produced by state-of-the-art evaluation system BLEU [2] (BLEU scores range from 0 to 1).

Table 1. Example of passive/active alternation in reference translation

		Human	BLEU
Source	All these activities should be monitored and supported by parliament.		
Reference	El parlamento debería controlar y apoyar todas estas actividades. [The parliament **should control and support** all these activities]		
Candidate	Todas estas actividades debería ser controladas y apoyado por el parlamento. [All these activities **should be controlled and supported** by the parliament]	4	0.1783

Here the English analytic passive construction is transformed into an active clause in the reference, whereas in MT no such changes are introduced and the source structure is preserved. However, MT obtains maximum score in manual evaluation. Clearly, human evaluators do not penalize the absence of optional changes if the sentence is well-formed and delivers the contents of the original. By contrast, the score produced by BLEU, which is based on n-gram matching, is extremely low because of the small number of shared word sequences between candidate and reference translations. If along with the available reference, other translation options preserving the

[2] Here and in what follows examples are extracted from English-Spanish MT evaluation data set used in the present work (see Section 5 for the description).

source analytic passive construction were provided, BLEU could do a much better job in approaching human assessment.

To analyze the actual impact of optional translation shifts on automatic MT evaluation, we developed a paraphrase generation system, which is based on a set of handcrafted transformation rules that "undo" optional shifts in HT. The system is designed for English-Spanish translation. We focus on the syntactic aspect of linguistic variation as lexical issues have been already addressed in the literature (see, for example, [3]). For evaluation, we performed a detailed manual annotation of structural changes in an English-Spanish parallel corpus and measured the proportion of cases where using additional references produced by our system improves automatic evaluation score.

The rest of this paper is organized as follows. In Section 2 we briefly describe the background of our work. Section 3 introduces the related work. Section 4 describes the paraphrase generation system. In Section 5 experiments and results are presented. Finally, in Section 6 we give the conclusions and discuss future work.

2 Background

Translation process is conditioned by the tension between two prototypical expectations: that of maximal similarity between source and translated texts and that of naturalness of the translated text in the target language. In terms of the distance between source and target texts, researches distinguish between analogous, equivalent and contextually appropriate translation [4]. Analogous translation involves similarity in form, as the translation retains as many forms of the original as possible. Equivalent translation gives priority to the semantic content, retaining the propositional meaning of the original as closely as possible. Finally, contextually appropriate translation optimizes discourse relevance and text processing conditions taking into account broad linguistic and extra-linguistic context without caring much for adherence to structure or lexis of the source.

The latter type of translation is the most common in practice. Translators normally shift away from the original and paraphrase what could be viewed as its close version. Optional shifts occur when formally similar structures have different semantic and/or pragmatic values in the languages involved [1]. Even in typologically related languages, where formally similar structures are available in many cases, not only we find different lexical and grammatical devices but also different uses of analogous lexical and grammatical means guided by language-specific principles of language use including stylistic issues and discourse processing conditions.

Here it should be noted that linguistic variation between possible translations of the same sentence is not only given by the presence or absence of optional translation shifts. It may occur when no formally equivalent construction is actually available in the target language and obligatory changes may be performed in various ways. Furthermore, alternative translation options can contain marked uses of language. For example, phrases occupying unmarked position in the source sentence may be topicalized in translation involving a change with respect to the original word order, due to

the differences in discourse processing and information structure preferences in the source and target languages. Note that in MT it is improbable to find such changes, as the unmarked options are normally the most frequent ones.

As we aim to generate translation alternatives in target language, we studied available formal descriptions of linguistic paraphrase [5,6] as well as translation shifts classifications [7,8,9]. Based on these works, we developed the following typology of translation phenomena[3], which was used for the design of transformation rules and for the evaluation of our paraphrase generation system.

1. Changes in grammatical features (finiteness, mood, modality, tense, aspect, etc.)
2. Changes in grammatical category (pronominalization, nominalization, manner adverbial → predicative adjective, etc.)
3. Diathesis changes (passive construction → active construction, personal clause → impersonal clause)
4. Level and function changes (phrase → clause, main clause → subordinate clause, temporal clause → conditional clause, locative-possessive alternation, etc.)[4]
5. Word order change (subject-predicate inversion, clitic climbing, changes in the position of adverbial modifiers, etc.)[5]
6. Changes in the number of constituents (ellipsis, additions of content words, deletions of content words)

As mentioned earlier, some changes are mandatory, as they are related to systemic differences between the languages involved, while others are not strictly required to produce a faithful and grammatically well-formed translation. In the present work we are interested in the differences between MT and HT induced by the presence of optional shifts in the latter, therefore only those were considered for the classification.

3 Related Work

Research in MT evaluation has demonstrated that the performance of the metrics improves significantly if a heterogeneous set of references is provided [10]. In practice, however, only one human reference is available for evaluation, and attempts have been made to generate additional references automatically [11]. For this purpose, data-driven methods are normally used [12]. Data-driven approaches have the advantage that information is automatically extracted from the data. However, they are not suitable for dealing with long-distance structural changes. More importantly, they do not allow inspecting the type of MT-HT differences that have been neutralized by using additional references.

---

[3] See Table 4 and Table 5 for illustration.

[4] Changes pertaining to this category have not been implemented, so we do not consider them in the following discussion.

[5] Individual changes that are entailed by other operations are not annotated separately. For instance, inversion of arguments induced by diathesis alternation is not considered in the category of word order changes.

A more similar approach to ours is developed by [9] who proposes a linguistic framework for formally codifying close translation properties. [9] states that close translation is the limit of MT performance. This is arguable because modern statistical MT actually tries to model translation decisions in context and can do better than close translation. We put into practice the idea presented in [9] by designing a system that generates close translation options automatically. We consider, however, that both shifted and close translation alternatives should be used for evaluation to be able to make a more fine-grained comparison of the quality of translations produced by different systems.

4 System Description

Our paraphrase generation system is intended to enhance MT evaluation with additional automatically generated references. The rationale behind the selection of particular paraphrase rules was their relevance in the context of English-Spanish translation. We defined sets of target language constructions that are approximately semantically equivalent in Spanish but are given different uses in source and target languages or no formal equivalent is available on either side. Thus, we expect that these constructions will be involved in prototypical structural changes in HT and are to be transformed in order to generate close translation options.

Table 2 presents the transformation rules we implemented. The rules are put in relation to the structural shifts typology presented in the previous section.

For cases where the direction of optional change in HT cannot be predicted, the rules are applied in both directions (marked with bi-directional arrows in Table 2). For example, the English verb forms in *-ing* with nominal function may be translated by nouns or infinitives in Spanish and in this case we cannot say that one option is closer to the source sentence than the other.

It should be noted that some of the constructions in Table 2 may be considered equivalent only given a specific linguistic context (for instance, tense alternations). We do not use any source-side information and thus applying such rules may result in paraphrases that change the contents of the original. However, these are not supposed to affect the results because the paraphrases are to be used together with the true human reference. Thus, in case human translator has changed a source structure that is preserved in MT, using the relevant paraphrase increases automatic evaluation score. If it is not the case, additional reference is not supposed to have any effect on the evaluation.

The system operates on dependency trees in CONLL format and returns full transformed sentences. At the analysis phrase, the structures to be transformed are identified by means of regular expression matching. In addition to syntactic information, grammatical dictionary of Spanish Resource Grammar [13] with verb frame information is used to introduce lexical restrictions for rule application.

If the conditions are matched and no restrictions are found, at the generation phase the system reconstructs the sentence with relevant changes using information extracted from the parses of the input sentences and morphological dictionary look-up in order to generate the appropriate word forms. From one input sentence, the system generates a set of paraphrases (as many as there are rules applied).

Table 2. Transformation rules for paraphrase generation based on translation shift typology

Translation shifts	Transformation rules
Grammatical Features	Past Simple ↔ Present Perfect Present Simple ↔ Present Perfect Present Simple ↔ Past Imperfective Simple Verb Form → Progressive construction Simple Future ↔ Periphrastic Future Recent Past Periphrasis → Present Perfect + 'recently' Habitual Aspect Periphrasis → Simple Verb Form + 'normally' Repetitive Aspect Periphrasis → Simple Verb Form + 'again'
Grammatical Category	Nominalization ↔ Denominalization Prepositional Phrase → Adverbial Modifier Copulative Clause → Adverbial Modifier
Diathesis	Active → Analytic Passive Synthetic Passive → Analytic Passive Personal → Impersonal
Word Order	Post-verbal Subject → Pre-verbal Subject Pre-verbal Adverbial ↔ Post-verbal Adverbial VP-External Adverbial → VP-Internal Adverbial Sentence-initial Adverbial ↔ Post-verbal Adverbial Sentence-initial Detached PP ↔ Post-verbal Detached PP Pre-nominal Adjectival Modifier → Post-nominal Adjectival Modifier Clitics before VP ↔ Clitics after VP
Addition / Deletion	Personal Pronouns [+Subject function][6] Repeated Prepositional Heads in Coordinated PPs

Note that not all of the relevant operations that have been described in Section 2 can be efficiently modelled in this way. One problem is that in cases of deletion operations in HT where content is left implicit, we lack information to reconstruct it. Furthermore, no quality language processing tools are yet available for analyzing certain complex phenomena. For example, in case of pronominalization shift, when

[6] Spanish is a pro-drop language and can elide subject pronouns. By contrast, their use is mandatory in English.

full noun phrase is substituted by a pronoun, which frequently happens in translation in order to avoid repetition, we lack high-quality co-reference resolution tools to reconstruct the original full noun phrase.

5 Experiments and Evaluation

The aim of the evaluation was twofold. In the first place, we wanted to assess the performance of the paraphrase generation system intrinsically. In the second place, we aimed to test the impact that translation phenomena discussed above have on MT evaluation. That is to say, we wanted to see how, in case optional transformations occur in HT, using additional references generated by the system affects automatic evaluation score.

For that purpose, we needed a parallel corpus annotated with translation shifts and a data set with manual evaluation scores for MT. There are parallel corpora in which translation shifts are annotated for some languages [8,9], but no such resource is available for English-Spanish translation. We decided to carry out manual annotation and classification of translation shifts on sentences extracted from [14] MT evaluation data set. The data set consists of 4,000 source sentences in English, their corresponding reference translations to Spanish randomly extracted from Europarl [15], the translations of four statistical MT systems and manual evaluation scores. MT systems were trained with data from the same domain. Manual evaluation scores were provided by professional translators using post-editing criterion.[7] It should be noted that Europarl is especially relevant for our work because it is widely used for MT development and evaluation and thus it is important to discuss the characteristics of the reference translations that are part of the corpus.

We randomly selected 290 sentences from the data set. Optional structural shifts in reference translations were annotated and classified manually using the typology presented in Section 2. The sentences were processed using MaltParser dependency parser [16] with Spanish models[8] and paraphrase generation system was applied to HTs to produce a separate reference set for each group of rules. MTs were automatically evaluated with BLEU in a single reference baseline scenario and in a multi-reference scenario, with automatically generated paraphrases. Note that when multiple references are provided BLEU takes into consideration n-gram matches between candidate translation and each of the references, which allows assessing the impact of different types of translation phenomena on evaluation.

We compared the resulting BLEU scores and calculated precision and recall based on the following principles. The purpose of using additional references was to increase BLEU scores for cases when a translation shift occurs in HT while MT contains the corresponding structure that is formally equivalent to the source. Thus, for each group of rules we considered that the application was successful if using the

[7] They were asked to indicate the amount of editing needed to make the MT ready for publishing, on a four-point scale: 1 – requires complete retranslation; 2 – a lot of post-editing is needed; 3 – little post-editing is needed; 4 – fit for purpose.

[8] Available at `http://www.iula.upf.edu/recurs01_mpars_uk.htm`

respective set of paraphrases increases BLEU score and the corresponding translation shift occurs in HT (true positives).

By contrast, rule application was considered unsuccessful when no translation shift of certain type occurs in HT and applying the corresponding set of rules increases the evaluation score (false positives), or when there is an optional change in the reference and applying the corresponding set of rules does not increase BLEU score (false negatives).

As mentioned earlier, our system currently covers a limited set of translation phenomena. Therefore, in order to assess the performance of the system per se, we counted recall separately for all the annotated translation shifts vs. translation shifts modelled by the rules. Table 3 presents precision and recall for each group of rules as well as the frequency of the translation phenomena involved.

Table 3. Precision and Recall for rule application and Frequency of translation shifts

Rule sets	P	R (all)	R (modelled)	Freq
Grammatical Features	0.76	0.43	0.60	104
Grammatical Category	0.70	0.30	0.61	77
Diathesis	0.59	0.20	0.43	66
Word order	0.79	0.40	0.72	151
Addition / Deletion	0.61	0.23	0.56	82
Total	**0.69**	**0.32**	**0.58**	**480**

The results must be interpreted as follows. The overall precision indicates that in 70% of cases of rule application the system successfully reconstructs the close translation option and using it as additional reference increases BLEU score. As expected, the recall is low in case all translation phenomena are considered, and much higher if calculated only for the phenomena covered by the rules. Thus, the system shows good performance in the cases it is designed to deal with. The overall number of translation shifts is high as there is an average of 1.7 optional changes per sentence in the reference, confirming the idea that such changes are indeed common practice in HT. An example of successful rule application is given in Table 4[9].

In this example clause-level adverbial modifier is changed into predicative adjective in HT (with corresponding changes in sentence structure). This transformation is common in English-Spanish translation, as translators are advised to avoid excessive use of manner adverbials in *-mente* (-ly) considered a calque from English where they are more frequent. MT preserves the structural organization of the original, which results in a sentence that is stylistically flawed, but is perfectly acceptable according to human evaluation score. The paraphrase generated by our system successfully neutralizes this shift in HT, and using it increases BLEU score. Note, however, that the increase is small as the system is not able to predict the exact position of the adverbial.

[9] HRT stands for Human Reference Translation and ART stands for Automatically generated Reference Translation.

Table 4. Example of category change in reference translation

Source	this event , on the eve of the lahti meeting , is clearly of particularly crucial significance to us .
MT	este acontecimiento , en vísperas de la reunión lahti , es claramente de especialmente crucial importancia para nosotros . [this event , on the eve of the lahti meeting , is clearly of particularly crucial significance for us .] Human evaluation = 4 BLEU with HRT = 0.2477 BLEU with ART = 0.2610
HRT	está claro que este acontecimiento , en vísperas del encuentro de lahti , reviste para nosotros una especial trascendencia . [**it is clear that** this event , on the eve of the lahti meeting , represents for us a special importance .]
ART	claramente , este acontecimiento , en vísperas del encuentro de lahti , reviste para nosotros una especial trascendencia . [**clearly** , this event , on the eve of the lahti meeting , represents for us a special importance .]

As far as specific groups of rules are concerned, the lowest results are for diathesis changes. In this group the most frequent transformation is reflexive passive → analytic passive. The resulting paraphrases are irrelevant, as they do not increase BLEU score because the corresponding shift frequently occurs in MTs. This is understandable given the nature of statistical MT. Since the change only involves local context and is consistently present in English-Spanish translations, it is expected to be found in high quality MT.

By contrast, word order changes are more challenging for statistical systems. For this reason, the group of rules that neutralize the optional changes affecting word order obtained the highest precision and recall. As an illustration, consider the example shown in Table 5.

Here the reference contains two optional changes: the transformation from analytic passive to reflexive passive and subject-predicate inversion. The first paraphrase delivers the close version with analytic passive construction. The second paraphrase reconstructs the word order of the source sentence neutralizing the subject-predicate inversion present in HT. In the case of word order, the rule is applied successfully as it increases BLEU score. In the case of diathesis transformation, the shift occurs in both HT and MT and thus the transformation performed by our system is not relevant.

Another source of errors is that, contrary to our assumption, not using source-side information does introduce noise. This is the case, for example, when the transformation involves adding a function word that happens to be present in MT but does not form part of the same syntactic construction.

Finally, both precision and recall are affected by parser errors[10]. For instance, order changes cannot be addressed in cases where the parser fails to identify the head of the

[10] The performance of MaltParser with Spanish models, considering exact syntactic match, is around 50% [17].

element to be moved. Parser errors are especially harmful for rule-based approach as the patterns have to be defined in detail and the conditions need to be exactly satisfied for the rules to apply.

Table 5. Example of diathesis change and subject-predicate inversion in human reference

Source	appropriate arrangements have been made for consultation with the member states .
MT	los preparativos apropiados se han hecho para su consulta con los estados miembros. [**appropriate arrangements MPASS[11] have made** for the consultation to the member states] Human evaluation = 4 BLEU with HRT = 0.3013 BLEU with ART1= 0.3013 BLEU with ART2 = 0.4683
HRT	se han realizado los preparativos apropiados para la consulta a los estados miembros. [**MPASS have made appropriate arrangements** for the consultation to the member states]
ART1	han sido realizados los preparativos apropiados para la consulta a los estados miembros. [**have been made appropriate arrangements** for the consultation to the member states]
ART2	los preparativos apropiados se han realizado para la consulta a los estados miembros. [**appropriate arrangements MPASS have made** for the consultation to the member states]

6 Conclusions and Future Work

Translation theory aims at explaining and predicting translators' behaviour. It is thus natural to use it in the field of MT. In present work, we bring together the research accomplished in the field of MT evaluation and theoretical notions from translation studies.

HT deviates from the source text in many ways making HT-MT comparison less informative for reference-based automatic MT evaluation. To show how this problem can be solved, we developed a rule-based paraphrase generation system for Spanish that produces additional translation options for English-Spanish automatic MT evaluation. We demonstrated that optional structural shifts have a negative effect on the performance of evaluation systems, which can be neutralized by using additional references that contain close translation options.

The results show that different translation phenomena have different impact on evaluation scores. The relevance of the paraphrases produced by our system depends on the corpus (underlying HT strategy) and the type of MT. The test set used in the present work contains only statistical systems trained with data from the same domain. Therefore, a considerable number of optional shifts that are regularly present in the reference are also found in MT. An idea worth investigating is that using the information on the type of reference (in our case, human or automatically generated)

[11] MPASS stands for the Spanish passive marker 'se'.

that MT is more similar to, quality levels can be defined and used to rate and describe the characteristics of a given system, making more fine-grained distinctions of MT quality.

The results are encouraging but certainly leave large room for improvement. We plan to augment the set of rules to perform a large scale evaluation of MT systems based on different strategies and assess the effect that using the paraphrases have on the correlation with human judgments. Also, to alleviate the shortcomings of rule-based paraphrase generation, a hybrid approach in which some of the relevant operations are learnt automatically may be used.

References

1. Szymańska, I.: Mosaics. A Construction-Grammar-based approach to translation. Semper, Warszawa (2011)
2. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. RC22176 (Technical Report), IBM T.J. Watson Research Center (2001)
3. Denkowski, M., Lavie, A.: Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In: Proceedings of the EACL 2014 Workshop on Statistical achine Translation (2014)
4. Doherty, M.: Language processing in discourse: a key to felicitous translation. Routledge, London (2002)
5. Barrón-Cedeño, A., Vila, M., Martí, M., Rosso, P.: Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. Computational Linguistics 39(4), 917–947 (2013)
6. Bhagat, R., Hovy, E.: What is a paraphrase? Computational Linguistics 39(3), 463–472 (2013)
7. van Leuven-Zwart, K.M.: Translation and original: Similarities and dissimilarities. Target 1(2), 151–181 (1989)
8. Cyrus, L.: Building a Resource for Studying Translation Shifts. In: Proceedings of the 5th International Conference on Linguistic Resources and Evaluation, pp. 1240–1245 (2006)
9. Ahrenberg, L.: Codified Close Translation as a Standard for MT. In: Proceedings of the 10th Annual Conference of the European Association for Machine Translation, pp. 13–22 (2005)
10. Albrecht, J., Hwa, R.: Regression for machine translation evaluation at the sentence level. Machine Translation 22(1-2), 1–27 (2008)
11. Owczarzak, K., Groves, D., Genabith, J.V., Way, A.: Contextual Bitext-Derived Paraphrases in Automatic MT Evaluation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pp. 148–155 (2006)
12. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proceedings of ACL (2005)
13. Marimon, M.: The Spanish DELPH-IN grammar. Language Resources and Evaluation 47(2), 371–397 (2013)
14. Specia, L., Turchi, M., Cancedda, N., Dymetman, M., Cristianini, N.: Estimating the Sentence-Level Quality of Machine Translation Systems. In: 13th Conference of the European Association for Machine Translation, pp. 28–37 (2009)

15. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit (2005)
16. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95–135 (2007)
17. Marimon, M., Bel, N., Padró, L.: Automatic selection of HPSG-parsed sentences for Treebank construction. Computational Linguistics 40(3), 523–531 (2014)

Arabic Transliteration of Romanized Tunisian Dialect Text: A Preliminary Investigation

Abir Masmoudi[1,2], Nizar Habash[3], Mariem Ellouze[1],Yannick Estève[2], and Lamia Hadrich Belguith[1]

[1] ANLP Research group, MIRACL Lab., University of Sfax, Tunisia
[2] LIUM, University of Maine, France
[3] New York University Abu Dhabi, United Arab Emirates
masmoudiabir@gmail.com, nizar.habash@nyu.edu, mariem.ellouze@planet.tn, yannick.esteve@lium.univ-lemans.fr, l.belguith@fsegs.rnu.tn

Abstract. In this paper, we describe the process of converting Tunisian Dialect text that is written in Latin script (also called Arabizi) into Arabic script following the CODA orthography convention for Dialectal Arabic. Our input consists of messages and comments taken from SMS, social networks and broadcast videos. The language used in social media and SMS messaging is characterized by the use of informal and non-standard vocabulary such as repeated letters for emphasis, typos, non-standard abbreviations, and nonlinguistic content, such as emoticons. There is a high degree of variation is spelling in Arabic dialects due to the lack of orthographic widely supported standards in both Arabic and Latin scripts. In the context of natural language processing, transliterating from Arabizi to Arabic script is a necessary step since most recently available tools for processing Arabic Dialects expect Arabic script input.

Keywords: Tunisian Dialect, corpus, transliteration, normalization, CODA.

1 Introduction

Recently, the evolution and development of information and communication technology have markedly influenced communication between correspondents. This evolution has made the transmission of information easier and has engendered new forms of written communication (email, chat, SMS, comments...).That is why we resort to the use of these written sources as a starting point to building large corpora automatically. However, most of these messages and comments are written with Latin script. The Tunisian Arabic Dialect (henceforth, *Tunisian Dialect*) written in Latin script is often referred to as 'Arabizi'. This fact is due firstly to the absence of Arabic keyboards in the new technologies (pc, smart-phone and tablet), which drove Tunisians to transcribe with Latin script. Secondly, it's due to the habit and the ease of Arabizi, especially that the Tunisians often insert words in French in their writings and their spoken conversations. Arabizi poses a problem for natural language processing (NLP) because some tools have recently become available for processing the Tunisian

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 608–619, 2015.
DOI: 10.1007/978-3-319-18111-0_46

Dialect input, e.g., ([11]; [16]), but they expect Arabic script input. We therefore need a tool that converts from Arabizi Tunisian to Arabic script Tunisian. However, the lackof standard orthography in the Tunisian Dialect compounds the problem: How should we convert Arabizi into Arabic script? Our answer is to use our orthographic convention CODA (Conventional Orthography for Dialect Arabic) [15].

In this study, we focus firstly on building a corpus consisting essentially of the Tunisian Dialect Arabizi messages taken from SMSs, social networks (Facebook, Twitter, etc.) and broadcast videos (Youtube, Dailymotion, etc.). Secondly, we address the problem of converting from Arabizi into Arabic script following the CODA convention. Thirdly, we present the result of our evaluation of this conversion.

The remainder of this paper is organized as follows: Section 2 reviews previous efforts in building DA resources. Section 3 explains our Tunisian Dialect corpus collection. We then present relevant linguistic facts (Section 4). Our method to transliterate Arabizi forms into Arabic script is explained in Section 5. In section 6, we report our experiments and evaluation.

2 Related Work

The transliteration problem has interested many linguists in different languages. Many researchers have worked on automatic transliteration in order to enrich lexicons and to create corpora which play a vital role in NLP applications. Concerning Arabic dialects, which suffer from a lack of resources, we notice recently the emergence of serious efforts to collect corpora using automatic transliterations as well as automatic translations and manual transcription.

In this context, we can cite the work of [4] who propose a system to transliterate Latin script into Arabic script. Their worker lies on the use of character transformation rules that are either handcrafted by a linguist or automatically generated from training data. They also employ word-based and character-based language models for the final transliteration choice. In another case, [6] presented a system that performs a transliteration of an Arabic text that is written using Latin script called Arabizi into Arabic script. His work is divided into two sections: language identification and transliteration. First, he used word and sequence-level features to identify Arabizi that is mixed with English. Second, for Arabic words, he modeled transliteration from Arabizi to Arabic script, and then applied language modeling on the transliterated text. Finally,[1] presented a system that converts a DA (Egyptian Arabic) text that is written with Latin script (called Arabizi) into Arabic script following the CODA convention for DA orthography. This system uses a finite state transducer trained at the character level to generate all possible transliterations for the input of Arabizi words. Then it filters the generated list using a DA morphological analyzer. After that, the best choice is selected for each input word using a language model.

There are several commercial products that convert Arabizi to Arabic script, namely: Microsoft Maren[1], Google Ta3reeb[2], Basis Arabic chat translator[3] and Yam-

[1] http://www.getmaren.com

[2] http://www.google.com/ta3reeb

[3] http://www.basistech.com/arabic-chat-translator-transforms-social-media-analysis/

li[4]. Since these products are for commercial purposes, there is little information available about their approaches, and whatever resources they use are not publicly available for research purposes.

Additionally, there is some work that uses automatic translation in order to convert text from DA to MSA. For example, [14] introduced a rule-based approach to translate EGY to MSA. Also, [2] used a rule-based method to transform from Sanaani dialect to MSA.

Moreover, there are other efforts that perform manual transcription to collect a corpus of DA. For example, [12] created a Tunisian Dialect corpus that they named TARIC: the Tunisian Arabic Railway Interaction Corpus. The creation of this corpus was done in three steps. The first step is the production of audio recordings; the second is the manual transcription of these recordings; and the third is the normalization of these transcriptions using CODA [15].

3 Corpus Collection

With the growth of the Web and the development of information and communication technology, people increasingly express and share their opinions through social websites and networks. Facebook, for example, is one of the most known and widely used participatory sites. These online resources and in particular the user comments have the following advantages for the constitution of a corpus: (1) a large amount of data, with more data generated and available daily; (2) the data is publicly available, with a coherent and structured format, and are easy to extract; (3) the data covers subjects with high levels of relevance; and (4) the dominant presence of DA.

So, we take advantage of this situation to collect a corpus of Tunisian Dialect Arabizi texts. We present next the different methods we used to build our corpus:

SMSs: We asked family and friends to send us their mobile phone text messages. The longest message consists of ten words, and the shortest consists of only one word.

Facebook: Today, social networks are one of the means of communication largely requested by users. Facebook is considered as the most popular social website in 2013 according to the website "The countries.com". Since social networks play an important status in the life of Tunisians, we chose to use their postings, messages and comments in Facebook to collect the corpus. The Facebook data extraction was done in two ways: (1) manually by copying personal messages and (2) automatically. To do this, a PHP script was developed in order to collect comments from only Tunisian pages. This script uses FQL[5] for comment extraction. We chose to use different types of Facebook pages to maximize vocabulary coverage and to ensure corpus diversity (media, politics, sports...).

[4] http://www.yamli.com/

[5] Facebook Developers: Facebook Query Language is a query language that allows querying users' Facebook data using the same interface style as SQL.

Youtube: Recent studies have shown that Youtube alone comprises approximately 20% of all HTTP traffic, or nearly 10% of the whole traffic on the Internet [5]. In the Arab World, people are increasingly using DA (e.g., Egyptian, Gulf, Tunisian, etc.) on sites like Youtube to comment and interact with their communities. In our work, only the Tunisian Dialect user Arabizi comments are kept. Table 1 provides the various statistics related to the collected corpus.

Table 1. The Tunisian Dialect corpus collection

Collection Source	Number of Messages	Number of Words
SMS	108	1,645
Facebook	70,237	864,935
Youtube	516	4,324
Total	70,861	870,904

Native speakers checked the collected corpus to verify that all the messages and words are in the Tunisian Dialect.

4 Linguistic Facts

4.1 Mixture of Languages

The Tunisian Dialect is distinguished by the presence of vocabulary from several languages other than Arabic such as Berber, French, Italian, Turkish and Spanish. This is due mainly to historical facts: the domination of the Ottoman Empire, European colonialism and peaceful trade-based interactions between civilizations. Indeed, [10] describe the linguistic situation in Tunisia as "poly-glossic" where multiple languages and language varieties coexist. This multilingualism is shown through the example in Figure 1, which is extracted from our corpus.

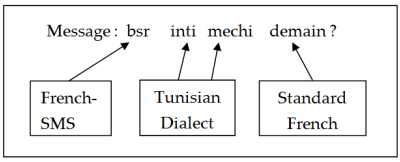


Fig. 1. Example of a text message in the Tunisian Dialect

The message in Figure1 consists of four words: the first word is in French SMS language ("bsr" which means "bonsoir" /good evening/ followed by two words in the Tunisian Dialect of Arabic origin ("inti" which means /you/ and "mechi" which means /are going/) followed by a standard French word "demain" /tomorrow/. In this message, we notice that three different languages varieties can be found in a single sentence: French SMS, Standard French and Tunisian Dialect in Arabizi. Given that

all of the words in our corpus are written with Latin script and due to use of foreign words in social media, it can be difficult to distinguish between Arabic words written with Latin script (Arabizi) and foreign words.

4.2 The Tunisian Dialect spontaneous Orthography

We noticed that a Dialect word can be written in several ways because in cases where there is no standard orthography, people use a spontaneous orthography that is based on various criteria [1]. The main criterion is phonology. Indeed, this technique involves writing words as they are pronounced. It replaces a sound with a Latin letter or a group of Latin letters. This mainly depends on language-specific assumptions about grapheme-to-phoneme mapping [1]. The same is true for the Tunisian Dialect, which has no standard Arabic-script orthography. Instead, it has a spontaneous Arabic-script orthography that is related to the standard orthography of MSA. Table 2 shows an example of writing a sentence in the Tunisian Dialect spontaneous orthography with different variants. It is an example from our corpus.

Table 2. The different spelling variants in the Tunisian Dialect and Latin script for writing the sentence "I have not bought bread today" versus its corresponding CODA form

Orthography	Example
CODA (Arabic script)	ما شريتش خبز اليوم *mAšriytišxebzAlyuwm*[6]
Non-CODA (Arabic script)	ماشريتش خبز اليومة *mAšriytišxebzAlywmaħ* مشريتش خبز اليوم *mašriytišxebzAlyuwma* مشريتش خبز اليومه *mašriytišxebzAlyuwmah*
Latin script	Machritechkhobzlyouma Ma chritichkhobzlyouma

It should be noted that Tunisian Dialect is characterized by a number of phonetic variations.

A few of these phonetic features of the Tunisian Dialect are presented [13]: The consonant ق'*q*' has a double pronunciation. In rural dialects, it is pronounced ق/*g*/. In the urban dialects, the consonant ق is pronounced /*q*/. Moreover, we noticed the elimination of a consonant in some words. For example, قلتلك '*qitlik*'/I told you/ can be pronounced قتلك '*qiltlik*'(the consonant ل '*l*' /l/ is eliminated) [11].Another Tunisian Dialect phenomenon is phonetic Assimilation [12]. This phenomenon can be defined as follows: action where a phoneme (assimilator element) communicates one or more of its features to a neighbor phoneme (the assimilated element). In Tunisian Dialect, the phonemeج'*j*' transforms to the phoneme ز'*z*'. For example, the Standard Arabic word عجوز '*ʕajuwz*' /old man/ becomes عجوز'*ʒjuwz*' or عزوز '*ʕzuwz*'. Additionally, a spontaneous orthography may reflect speech effects such a word stretching (repeated sequences of letters) to express intense emotions, e.g., 'Bnnnnina', 'Mabrouuuk'

[6]Arabic transliteration is in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

and 'Barrrrrrrcha' which mean بنينة'*bniynaħ*' /delicious/, مبروك'*mabruwk*' /congratulations/ and برشة 'bar*šaħ'* /so much/ respectively.

4.3 Arabizi

Arabizi is a spontaneous orthography used to write DA using Latin script. Arabizi is often used in communication over the internet (chat, comments, etc.) or for sending messages (instant messaging and mobile phone text messaging). As mentioned above, the use of Arabizi is due to different reasons: firstly the lack of Arabic keyboards in some new technologies and secondly the habit and the ease of Arabizi especially that Tunisians often insert words in French in their writings and their spoken conversations. The orthography used to write in Arabizi depends principally on a phoneme-to-grapheme mapping between the Arabic pronunciation and Latin script. Crucially, Arabizi is not a simple transliteration of Arabic, under which each Arabic letter in some orthography is replaced by a Latin letter (as is the case in the Buckwalter transliteration used widely in natural language processing) [1]. Next we present some specific aspects of Arabizi.

– **Consonants**: We present some grapheme-phoneme equivalences between Latin script and Arabic script extracted from our corpus. For example, the Latin script 'b', 's' and 'l' are used to represent the sound of Arabic letters بْ '*b*', سْ '*s*' and لْ '*l*' respectively. However, we encountered some ambiguities due to the absence of sufficient Latin script to present all the pronunciations of the Arabic script, which can be an obstacle in transliteration. For example, the Latin script "t" is used to represent the sounds of the Arabic letters ت '*t*' and ط '*T*'. Another example, the Latin script "s" is used to represent the sounds of the Arabic letters س'*s*' and ص '*S*'. Also, the Latin letter "h" is used to represent the sounds of the Arabic letters ح'*H*' and ه '*h*'. Additionally, some pairs of Latin script can ambiguously map to a single Arabic letter or pair of letters: e.g., "dh" can be used to representض'*D*' and ده '*dh*', and "kh" can be used to representخ'*x*' and كه '*kh*'. In Arabizi, digits may replace letters and sounds that do not have equivalents in the Latin alphabet. For example, the digits 3, 5, 7 and 9 are used to represent the sounds of the lettersع'ئ,خ '*x*', ح '*H*' and ق '*q*', respectively. Furthermore, when a digit is followed by "'", the numbers 3, 6, 7 and 9 change their interpretations and become غ '*g'*, ظ '*Ď*', خ '*x*' andض '*D*'. We note in this context that the use of digits is a also characteristic of French SMS language where digits replace sound sequences reflecting the pronunciation of the digits, e.g., “demain - 2m1” /tomorrow/. This causes difficulty in deciphering messages given the use of digits in Tunisian Arabizi.

– **Vowels:** Tunisians use the Latin script symbols (a, e, i, o, u, y) to represent the Tunisian Dialect's short and long vowels.

– **Foreign Words:** Many foreign words are used and even integrated in the Tunisian Dialect messages and comment such as *demain* "tomorrow".

– **Abbreviations:** Arabizi may include some abbreviations such as 'hmd', 'wlh' and 'slm' which mean الحمد الله *'HamdAllah'* /Thanks God/, وَ الله *'wa Allah' /By God/*, سلام 'salAm' /Peace/, respectively.

– **Sound Effects:** We also observed the frequent use of written out representations of speech effects, including representations of laughter (e.g., *hhhhh*), filled pauses (e.g.,*umm*), and other sounds (e.g., *hmmm*).

– **Acronyms:** These correspond to the initials of a group of words forming an expression or a name of an institution. For example, the acronyms 'T7' and 'o/n' which mean تونس سبعة *'tuwnis sabçaħ'* /Name of a Tunisian Channel/ and نعم أو لا *'na Eam AawlA'* /yes or no/, respectively.

– **Emoticons and Emoji:** Tunisian messages express a person's state of emotion by emoticons. Emoticons are a set of numbers or letters or punctuation marks used to express feelings. Emoji are a special set of images used in messages.

4.4 CODA

CODA is a conventionalized orthography for Dialectal Arabic [7]. In CODA, every word has a single orthographic representation. The design of CODA respects several principles. Firstly, CODA is an ad hoc convention using only the Arabic characters, including the diacritics for writing Arabic dialects. Secondly, CODA is consistent. A unique orthographic form that represents the phonology and morphology for each word is used. CODA uses MSA-consistent and MSA-inspired orthographic decisions (rules, exceptions and ad hoc choices). CODA preserves, also, dialectal morphology and dialectal syntax. CODA is easily learnable and readable. All Arabic dialects generally share the same CODA principles; each dialect will have its unique CODA map that respects its phonology and morphology. However, CODA is not a purely phonological representation. Text in CODA can be read perfectly in DA given the specific dialect and its CODA map. CODA has been designed for the Egyptian Dialect [7] as well as the Tunisian Dialect [15] and the Palestinian Levantine Dialect [9]. For a full presentation of CODA and an explanation of its choices, see ([7], [15]).

5 Arabizi to Arabic Script

Our objective is the following: for each Arabizi word in the input, we want to select its Arabic script form following CODA. In this paper, we do this by first automatically generating a set of possible transliterations into Arabic script (all following CODA as much as possible). We then manually select the best choice in context with the help of one Tunisian native speaker annotator. The annotator is instructed to select from among the choices given and not add any additional answers. If none of the answers are correct, the annotator selects the form that is the least problematic.

To accomplish the first step (generation of forms), we use a rule-based approach that consists in using a set of rules and a lexicon of exceptions. This lexicon of exceptions contains principally the abbreviations and the acronyms. The Arabizi form of each exceptional word is entered with its Arabic script form. The lexicon of exceptions is scanned first. Otherwise, we must apply the rules to the word to generate its Arabic form.

The process of transliteration consists of a certain number of well-defined steps:

– We directly transliterated abbreviations and acronyms using an exception lexicon.

– Emoticons and emoji were replaced in the transliteration with #.
– Since people often repeat sequences of letters to express intense emotions, we removed any repetition of a letter beyond one repetition. For example, we transformed the word “bninnnna” /delicious/ to “bnina”.
– The final step consists in the application of our rules for each word. Since we perform the transliteration of Arabizi into Arabic script following CODA, a pretreatment phase is necessary: For example, in the case where CODA requires two consecutive Arabizi words to be merged, we indicate this by adding a plus to the end of the first word. For example, if the two Arabizi words '3al' (Arabic prepositions, English equivalent: 'on/upon/about/to’) and 'tawla' /Table/ are merged and become '3al+tawla', the output is عالطاولة 'ʕaAlTAwlaħ'/on the table/. According to CODA, this Arabic proposition /on/ must be attached to the beginning of the second word which begins with the definite article *Al*. So, this pretreatment is necessary in order to have a correct CODA form in Arabic script.

In the case where CODA requires an input Arabizi word to be broken into two or more Arabic script words, we indicate this by adding a dash between the words. For example, the Arabizi word 'Ma5rajch' /he didn’t come out/ must be broken into two Arabic words as /'Ma' - '5rajch'/ما -خرجش *'mA xrajš'* /he didn’t come out/ where 'Ma' (equivalent of [not] in English) represents the Tunisian Dialect negation clitic which cannot be attached to the next word according to CODA.

As mentioned above, we encountered some ambiguities in Arabizi consonants due to the absence of sufficient Latin script to present all the pronunciations of the Arabic script, which can be an obstacle in transliteration. We noticed that only experts of the Tunisian Dialect can distinguish these cases. To overcome these obstacles, we proposed a solution that consists in enumerating all the possible versions of the word in the input. After that, the user picks the best choice of all the possibilities. For example, the Arabizi word 'hlel' contains the Latin grapheme 'h' which is used to represent the sounds of the Arabic letters ه'h' and ح'H'. So, the output should be all the possibilities of this Latin grapheme as هْلالْ'hlAl' or حْلالْ 'HlAl'.

Arabizi to Arabic script example:

<**Arabizi**>3andik flous bech tichry karhba!!!</**Arabizi**>
<**Arabic**>!!! عنـــدك فلوس/فلـــوص باش/باســـه تشـــري كرهيـــة/كرهياء/كرهيا/كرهبى </**Arabic**>
<**Arabic- Annotator**>!!! عنـدك فلـوس بـاش تشـري كرهيـة</**Arabic- Annotator**>
<**Means**>have you money to buy a car!!!</**Means**>

- Arabizi word ‘flous’ contains Latin letter ‘s’ which is used to represent the sounds of the Arabic letters س's' and ص 'S'. So, the output should be all the possibilities of this Latin grapheme, فلـــوس ‘flws’ /money/ or فلـــوص ‘flwS’.
- Arabizi word ‘besh’ contains pairs of Latin script ‘sh’ can ambiguously map to a single Arabic letter ‘ش’ or pair of letters ‘سه’. So, the output should be all the possibilities of this Latin grapheme, بـاش bAš’ /will/ or باســه ‘bAsh’.
- Arabizi word ‘karhba’ ends with Latin letter ‘a’: in our work when the letter ‘a’ appears at the end of the word, it can be transliterate by several Arabic letters that are {ة ،ا ،اء ،ى}. So, the transliteration of Arabizi word ‘karhba’ is كرهبـة ‘karhbaħ’ /a car/, كرهبـاء ‘karhbA’’and كرهـبى ‘carboy’.

Overall, using the above-mentioned method, we annotated 530 sentences.

6 Experimental Setup and Evaluation

In this section, we evaluate the quality of the Arabizi-to-Arabic script generation step described above. Since we do not automatically select a choice in context, the evaluation is intended to judge the degree our transliteration mapping script can help the overall process of transliteration. We carried out two types of evaluation: out-of-context evaluation and in-context evaluation. In the following section, we will give more details on the processes of evaluation.

Out of context evaluation: We asked judges who are native speakers of the Tunisian Dialect to transliterate manually a set of 3,500 Arabizi words (the words are not redundant) into Arabic script. This set of words includes especially words of Arabic origin and foreign words such as French words. The distribution of these words is as follows: 2,754 Arabic words and 746 foreign words. The evaluation consisted in comparing what we had proposed in our system as a transliteration with the decisions of the judges. We compute the recall of our system as the percentage of agreement between the judges' transliterations and the transliterations proposed by our system. Table 3 shows the results.

Table 3. The recall of the judges' transliterations by our system in the case of out of context evaluation

Type	Recall
Words of Arabic origin	93%
Foreign words	90%

The analysis showed that errors of words of Arabic origins are mainly due to the following reasons:

– Errors due to the ambiguity of the Arabizi word: the input contains a typo making it impossible to produce the gold reference. For example, the input '5obs' contains a typo where the final "s" should turn into z so that it means خبز '*xubz*' /Bread/.
– Errors occur where the system generates translation of some words that are not compatible with the CODA form. For example, the system generates the non-CODA form ليّام '*layyAm*' /the days/ instead of the correct CODA form الأيّام '*Alyyam*' /the days/.

– Other types of errors:
- Morphological errors: we noticed an incorrect transliteration of the third person plural verbal suffix وا '*wA*' in some verbs. For example, the system generates the verb form خرجُ '*xarju*' instead of the correct verb form خرجوا '*xarjwA*' /they came out/.
- Segmentation errors: we noticed that some particles such as لا 'lA' /no/ are attached to words. For example, the system generates the form لامشى 'lAm*šA′* /Not-walk/ instead of the correct form لا مشى 'lAmšA' /No he left/.

– Errors due to the incorrect transliteration of some foreign words. For example, the system generates the transliteration of the foreign word 'courage' as كورَجْ'*kwraj*' /courage/ but according to the judges, this word must be translated as كورَاجْ '*kwrAj*' /courage/.

In context evaluation: In this evaluation, we computed the accuracy of producing the correct transliterated equivalent in context. So, we asked 4 judges to transliterate 200 sentences containing 832 words. In this sample, we repeated some words in the same sentence but in a different context.

At the beginning, we tested the percentages of agreement between the transliterations of the judges. Table 4 gives the results of inter-judge agreement. The variation in percentage is due to the fact that for some words, the judges did not agree with each other.

Table 4. Results of inter-judge agreement

	2 judges	**3 judges**	**4 judges**
Agreement	94%	93%	90%

In an analysis of inter-annotator agreement, the overall agreement between the four judges was 90%. We analyzed all the disagreements and classified them in fourhigh level categories:

– **CODA**: Some cases of disagreement were related to CODA decisions that did not carefully follow the guidelines. In some cases, the disagreements are related to the spelling of the Hamza and in other cases, the disagreements involved the spelling of the Tunisian Dialect words.
– **Foreign words**: Some cases of disagreement were related to foreign words. In fact, in some cases the judges did not agree on the transliteration of foreign words. For example, the French word '*demain*' /tomorrow/ was transliterated into Arabic script by two judges as دومان *'dwmAn'*/tomorrow/ and it was transliterated into Arabic script by two other judges' as دمان *'dumAn'*/tomorrow/.
– **Ambiguity:** The judges' disagreement reflected a different reading of Arabizi word which resulted in an inflectional feature.

After that, we performed a second evaluation that consisted in comparing what we have proposed in our system as a transliteration with the proposals of the judges. The percentage of agreement between the judges' transliterations and the transliterations proposed by our system was calculated. The calculation of the percentage of agreement and disagreement was done as follows: If there is an agreement between the proposal of our system and only one of the proposals of the judges, we attributed a value of 1, and if not, the value should be 0. Table 5 shows the percentage of agreement between the judges' transliterations and the transliterations proposed by our system in the case of in context evaluation.

Table 5. The percentage of agreement between the judges' transliterations and the transliterations proposed by our system in the case of in context evaluation

Type	Agreement
Words of Arabic origin	92%
Foreign words	89%

The errors are mainly due to the following reasons:

– Errors due to the ambiguity of the Arabizi word; for example, the Arabizi word is 'jbal'/mountain/in the context 'barcha jbal' /many mountains/ where the output from the system is جبل '*jbal*' /mountain/, while the correct answer is جبال '*jbAl*'/mountains/ instead.
– Errors occur where the system generates some word translations that are not compatible with the CODA form. For example, in the case where Arabizi word is 'ma9alech' the system generates the non-CODA form مقالش '*maqAališ*/he didn't say/ instead of the correct CODA form ما قالش *mA qAališ*(two separate words).
– Errors due to the incorrect transliteration of some foreign words.

7 Conclusion

This paper presented an effort to create a transliteration tool for the spontaneous romanizations of Tunisian Dialect (Tunisian Arabizi). This tool allows a conversion from Arabizi into Arabic script following the CODA convention for DA orthography. To do this, we collected a corpus from social media and SMS messaging. The language used in social media and in SMS messaging is characterized by the use of informal and non-standard vocabulary such as repeated letters for emphasis; typos and nonstandard abbreviations are common; and nonlinguistic content, such as emoticons, is written out. This is due firstly to the absence of standard orthographies of all the Arabic Dialects; secondly, this is due to the lack of standard Romanization. In the context of NLP, tools have recently become available for processing the Tunisian Dialect input, and they expect Arabic script input. So, transliterating from Arabizi to Arabic script is necessary. To perform the transliteration, we used a rule-based approach for the implementation of our system. This system generates all possible transliterations for the Latin script input. After that, the annotator is instructed to select from among the choices given and not add any additional answers. If none of the answers are correct, the annotator selects the form that is the least problematic.. Since we do not automatically select a choice in context, the evaluation is intended to judge the degree our transliteration mapping script can help the overall process of transliteration. We carried out two types of evaluation: out-of-context evaluation and in-context evaluation. The error rate of words of Arabic origin is ~10%.

In the future, we plan to improve several aspects of our models, particularly the use of an automatic tool to pick the best choice among all the possibilities generated by our transliteration system for each Arabizi word. We also plan to work on the `problem` of automatic identification of Arabic and non-Arabic words [8].

References

1. Al-Badrashiny, M., Eskander, R., Habash, N., Rambow, O.: Automatic Transliteration of Romanized Dialectal Arabic. In: Proceedings of the Eighteenth Conference on Computational Language Learning, Maryland, USA (2014)
2. Al-Gaphari, G., Al-Yadoumi, M.: A method to convert Sana'ani accent to Modern Standard Arabic. International Journal of Information Science and Management (2010)
3. Bies, A., Song, Z., Maamouri, M., Grimes, S., Lee, H., Wright, J., Strassel, S., Habash, N., Eskander, R., Rambow, O.: Transliteration of Arabizi into Arabic Orthography: Developing a Parallel Annotated Arabizi-Arabic Script SMS/Chat Corpus. In: Arabic Natural Language Processing Workshop, Qatar (2014)
4. Chalabi, A., Gerges, H.: Romanized Arabic Transliteration. In: Proceedings of the Second Workshop on Advances in Text Input Methods (2012)
5. Cheng, X., Dale, C., Liu, J.: Understanding The Characteristics Of Internet Short Video Sharing: YouTube As A Case Study (2007)
6. Darwish, K.: Arabizi Detection and Conversion to Arabic. CoRR (2013)
7. Diab, M., Habash, N., Owen, R.: Conventional Orthography for Dialectal Arabic. In: Proceedings of the Language Resources and Evaluation Conference, Istanbul (2012)
8. Eskander, R., Al-Badrashiny, M., Habash, N., Rambow, O.: Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script. In: Arabic Natural Language Processing Workshop, Qatar (2014)
9. Jarrar, M., Habash, N., Akra, D., Zalmout, N.: Building a Corpus for Palestinian Arabic: a Preliminary Study. In: Proceedings of the Arabic Natural Language Processing Workshop, EMNLP, Doha (2014)
10. Lawson, S., Sachdev, I.: Code Switching in Tunisia: attitudinal and behavioral dimensions. Journal of Pragmatics 32 (2000)
11. Masmoudi, A., Ellouze Khmekhem, M., Estève, Y., Bougares, F., Dabbar, S., Hadrich Belguith, L.: Phonétisation automatique du Dialecte Tunisien. 30 éme Journée d'étudessur la parole, Le Mans-France (2014)
12. Masmoudi, A., Ellouze Khmekhem, M., Estève, Y., Hadrich Belguith, L., Habash, N.: A corpus and a phonetic dictionary for Tunisian Arabic speech recognition. In: 19th edition of the Language Resources and Evaluation Conference, Iceland (2014)
13. Masmoudi, A., Estève, Y., Ellouze Khmekhem, M., Bougares, F., Hadrich Belguith, L.: Phonetic tool for the Tunisian Arabic. In: The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages, Russia (2014)
14. Shaalan, K., Abo Bakr, H., Ziedan, I.: Transferring Egyptian Colloquial into Modern Standard Arabic. In: International Conference on Recent Advances in Natural Language Processing, Bulgaria (2007)
15. Zribi, I., Boujelbane, R., Masmoudi, A., Ellouze Khmekhem, M., Hadrich Belguith, L., Habash, N.: A Conventional Orthography for Tunisian Arabic. In: Proceedings of the Language Resources and Evaluation Conference, Iceland (2014)
16. Zribi, I., Ellouze Khmekhem, M., Hadrich Belguith, L.: Morphological Analysis of Tunisian Dialect. In: International Joint Conference on Natural Language Processing, Nagoya, Japan (2013)

Cross-Dialectal Arabic Processing

Salima Harrat[1], Karima Meftouh[2], Mourad Abbas[3], Salma Jamoussi[4], Motaz Saad[5], and Kamel Smaili[5]

[1] ENSB*, Ecole Supérieure d'Informatique (ESI), Algiers, Algeria
[2] Badji Mokhtar University, Annaba, Algeria
[3] CRSTDLA**, Algiers, Algeria
[4] MIRACL***, Pole Technologique de Sfax, Tunisia
[5] Campus Scientifique LORIA, Nancy, France

Abstract. We present, in this paper an Arabic multi-dialect study including dialects from both the Maghreb and the Middle-east that we compare to the Modern Standard Arabic (MSA). Three dialects from Maghreb are concerned by this study: two from Algeria and one from Tunisia and two dialects from Middle-east (Syria and Palestine). The resources which have been built from scratch have lead to a collection of a multi-dialect parallel resource. Furthermore, this collection has been aligned by hand with a MSA corpus. We conducted several analytical studies in order to understand the relationship between these vernacular languages. For this, we studied the closeness between all the pairs of dialects and MSA in terms of Hellinger distance. We also performed an experiment of dialect identification. This experiment showed that neighbouring dialects as expected tend to be confused, making difficult their identification. Because the Arabic dialects are different from one region to another which make the communication between people difficult, we conducted cross-lingual machine translation between all the pairs of dialects and also with MSA. Several interesting conclusions have been carried out from this experiment.

1 Introduction

In Arab countries, the majority of people speaks dialects. Modern Standard Arabic (MSA) is the official language used only in formal speeches, media and education. What may be surprising is that even educated people, in their daily life prefer speaking the dialect which is their mother-tongue. Consequently, studying the dialects becomes a priority which could take benefit from natural language processing tools.

During the last decade, researchers have been interested to Arabic dialects processing, like building lexicon, morphological analysis, POS tagging, etc, [1–4].

* Ecole Normale Supérieure Bouzareah.
** Centre de Recherche Scientifique et Technique pour le Développement de la langue Arabe.
*** Multimedia, InfoRmation systems and Advanced Computing Laboratory.

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 620–632, 2015.
DOI: 10.1007/978-3-319-18111-0_47

Recent works have been dedicated to other tasks, such as Machine Translation [5, 6] and dialect identification [7, 8]. In [9], a work of building a small multilingual dialectal corpus is presented further including the MSA.

In this paper, we will focus on a set of Arabic dialects and more particulary on three from Maghreb (two from Algeria and one from Tunisia). On the other side we will conduct a study and experiment on Palestinian and Syrian dialects. To do that, we build a parallel corpus, study the relationship between dialects and MSA, distinguish one dialect from another and present few experiments of machine translation between MSA and the different dialects. This paper is structured as follows: in section 2 we give an overview of the considered dialects. We discuss in section 3 how resources are built with some related works, then we detailed how we created our parallel corpus. Section 4 is dedicated to an analytical comparison of all dialects and MSA. Section 5 presents dialect identification experiments whereas the last one gives results of machine translation between all dialects and MSA. Finally, we conclude in section 7 by summarizing all the work.

2 Overview of the Used Dialects

Arabs use in their daily conversations dialects which could be considered such as variants of MSA. Tunisian and Algerian dialects share many features with the other Maghrebi dialects because of their similar history. It is worth mentioning that they contain many words borrowed from other languages, mainly Berber, French, Turkish, Italian and Spanish. Syrian and Palestinian dialects share an important number of features since they are included in the Levantine Arabic dialect continuum. In the following, we give a short overview about each dialect we study in this article.

2.1 Algerian Dialect

Algerian dialect is an informal spoken language, not used in official speech. Its vocabulary is roughly similar through all Algeria. However, in the east of the country, the dialect is closer to Tunisian whereas in the west it is closer to Moroccan. Most of the words of Arabic dialect come from MSA [10], but there is significant variation in the vocalization in most cases, and omission of some letters in other cases. Contrary to MSA, few letters are not used in Algerian as ظ and ذ, where most of the time they are respectively pronounced as ض and د.

Moreover Algerian dialect uses some non-Arabic letters like ڤ and پ.

2.2 Tunisian Dialect

Like other Maghrebi dialects, the vocabulary of the Tunisian dialect is mostly Arabic, with significant Berber substrates. However, its morphology, syntax, phonology and vocabulary differ from standard Arabic. The Tunisian dialect is

very agglutinative: people tend to use very few words for conversation where one word may express a whole sentence. It differs from MSA especially in its negation form where the markers are always agglutinated to other words as affixes or suffixes. Moreover, in Tunisian dialect, several Arabic words are used with substantial changes in their stem formation.

2.3 Syrian and Palestinian Dialects

Syrian and Palestinian dialects are part of Levantine Spoken Arabic which covers also dialects spoken in Lebanon and Jordan. Levantine Arabic shares most phonological, structural, and lexical features with other varieties of Arabic. At the same time, there are differences among Levantine dialects based on geography and urban/rural division. Arabic Syrian dialect is influenced by the Syriac language, a Semitic language of the Middle East, belonging to the Aramaean language group. It contains a large proportion of Arabic words and also words borrowed from Turkish and French. Palestinian dialect has slight phonetic differences from north Levantine dialects. It can be classified into two main categories: urban and countryside. It can be classified also according to geographical area (north and south). Palestinian dialect built in this work is mainly the dialect of people who live in Gaza strip.

3 Building a Parallel Corpus

It is well known that parallel corpora are the foundation stone of several natural language processing tasks, particularly cross-language applications such as machine translation, bilingual lexicon extraction and multilingual information retrieval. Building this kind of resources is a challenging task especially when it deals with under-resourced languages [11]. Arabic is one of these languages for which parallel corpora are scarce. The problem is much deeper with the Arabic dialects which are used by a huge number of people but, unfortunately they are often not written. To overcome the need of text corpora covering these languages, researchers can choose one of two main possible solutions: building the corpus from scratch or crawl the web to build a parallel set of sentences.

The solution adopted for our work is the first one: from scratch, since the overall goal of this work is Speech-to-Speech translation we need real everyday conversations. In the following, we focus on Annaba's dialect (ANB), the language spoken in the east of Algeria, on Algiers's dialect (ALG), the language used in the capital of Algeria, on Sfax's dialect (TUN) spoken in the south of Tunisia, Syrian (SYR) and Palestinian (PAL) dialects.

ANB corpus was created by recording different conversations from every day life whereas, for ALG, we used the recordings corresponding to movies and shows which are often expressed in the dialect of Algiers. Then we transcribed both of them by hand. To increase the size of the two corpora, we translated each of them into the other. Afterwards, these two corpora have been translated into MSA.

In order to introduce both Tunisian, Syrian and Palestinian dialects, we used MSA as a pivot language. We translated the MSA corpus to TUN, SYR and PAL. The Tunisian corpus was produced by 20 native speakers. Each one was responsible of translating almost 320 sentences from MSA to TUN. Speakers have very slight differences in their spoken languages. All of them are from the south of Tunisia where people tend to use Arabic words rather than French words as it is the case in the north of the country. In fact, the dialect used in the south is closer to the Standard Arabic than that used in the north of Tunisia. Syrian and Palestinian corpora were created in the same way as Tunisian one except that each of them has been obtained by two translators. Finally, we get a parallel corpus including ANB, ALG, TUN, SYR, PAL and MSA.

It should be noted that each dialect word is written by adopting the Arabic notation, that means if a dialectal word does exist in MSA, it is written in a standard form without any change, otherwise the word is written as it is uttered. We give in Table 1 an example of a same sentence from the built corpus. We can remark, even if we do not read Arabic that some words have the same form in several dialects, while others are completely different.

Table 1. An example of a sentence from the parallel corpus

Dialect/Language	Sentence
ALG	الوقت ڤاع لي درتو باش نجي ليك حبيتي تخسريه في دقيقة
ANB	الوقت أدا كل لي عڤبتو باه نجي عندك حايبة تفزديه في دقيقة
TUN	الوقت الكل الي قعدتو باش نجيك اتحب اتضيعو في دقيقة
SYR	كل الوقت اللّي قضيتو و انا جاي عندك بدك تروحيه بدقيقة
PAL	كل الوقت اللّي اخدتو عشان آجي عندك بدك تضيعيه في دقيقة
MSA	كل الوقت الذي استغرقته لكي آتي عندك تريدين أن تهدريه في دقيقة
Meaning	All the time that I put to visit you, you want to spoil it in one minute

4 Analytical Comparison

In the following, we will compare dialects between them and with MSA. The idea is to understand what is close to what? What is different? etc. We hope that this will help us in a future work to take advantage of MSA in order to develop linguistic tools for processing dialects.

4.1 Multi-dialect Corpus Statistics

The obtained parallel corpus is made up of 6400 parallel sentences. The MSA part contains 40906 words including 9131 different words. The five dialects, ALG,

ANB, TUN, SYR and PAL include an average of 37500 words with a vocabulary which does not exceed 10250 words (see Table 2). The average number of words in a dialect sentence is of 6 while it is of 7 for MSA. The shortest sentence in the corpus is composed of 4 words and the longest one contains 25 words.

Table 2. Parallel corpus description

Corpus	#Distinct words	#Words
ALG	8966	38707
ANB	9060	38428
TUN	10215	36648
SYR	9825	37259
PAL	9196	39286
MSA	9131	40906

4.2 Common Lexical Units between MSA and Dialects

MSA language is the same throughout the Arab world, while the dialects vary according to the geographical location. In this Section, we are interested in measuring how much the dialectal vocabulary is close to MSA by using the aforementioned parallel corpus. The experiments we achieved, show that the dialects employ many MSA words, even if the utterance of these words depends strongly on each dialect. Particularly, PAL is closest to MSA than other dialects are (Table 3).

Table 3. Percentage of common words between Arabic dialects and MSA

Dialect	ALG	ANB	TUN	SYR	PAL
%	21.18	21.07	37.60	37.36	51.68

These results are not surprising. Indeed, Middle-East Arabic dialects tend to be closer to MSA than those of the Maghreb. Also, it would be noticed that Arabic dialects spoken in south of Maghreb countries include more Arabic words than those spoken in the north. This explains the different rates in terms of common words between the two Algerian dialects on one side and the Tunisian dialect on another side. Indeed TUN is spoken in the south of Tunisia while ALG and ANB are dialects of northern Algeria. In Table 4, we give few examples of the most frequent words between the studied Arabic dialects and MSA.

In the same way, we computed the percentage of common words between all pairs of dialects. The Table 5 represents the percentage of common words between different dialects. These values show that ALG and ANB share the largest number of words, followed by PAL and SYR. These results were excepted because ALG dialects and ANB are close since they are used in two cities separated

Table 4. The most frequent words of each dialect relatively to our corpus

Dialect	Most Frequent words			
ALG	واحد *one*	صح *right*	راح *he went*	كامل *full*
ANB	واحد *one*	صح *right*	راح *he went*	عندك *you have*
TUN	كان *it was*	وقت *time*	واحد *one*	الكل *all*
SYR	اليوم *today*	مرة *one time*	واحد *one*	عندي *i have*
PAL	اليوم *today*	واحد *one*	طيب *good*	راح *he went*

by 372 miles, as PAL and SYR which are used in the same geographic location separated by only 175 miles. Also, TUN shares more words with PAL than the two other Maghrebi dialects do. Only 23% in average of words are common to Syrian and Maghrebi dialects. This result reinforces the fact that we made at the beginning of the article about the difficulty of conversing between Arabic people, from Maghreb and middle-east.

Table 5. Cross dialect percentage of common words

	Percentage of common words				
Ref.	ALG	ANB	TUN	SYR	PAL
ALG	-	73.62	35.43	24.16	25.43
ANB	72.86	-	34.25	23.59	25.00
TUN	31.10	30.38	-	29.79	33.49
SYR	21.01	20.73	29.52	-	44.00
PAL	24.79	24.63	37.20	49.33	-

We estimated also the percentage of common words at sentence level between each pair of languages. For each pair of the k^{th} aligned sentences $S^k_{L_i}$ and $S^k_{L_j}$ from the bitext (L_i, L_j). The common words is calculated as in formula 1 , it corresponds to the percentage of common words in the two sentences to the total number of words in both sentences. Then we estimate the average common number of words over all the sentences.

$$Ovp(S^k_{L_i}, S^k_{L_j}) = \frac{|S^k_{L_i} \cap S^k_{L_j}|}{|S^k_{L_i} \cup S^k_{L_j}|} \quad (1)$$

Table 6 presents the overlap between the Arabic dialects and MSA at sentence level. The achieved results confirm those of the two last experiments. PAL is the closest dialect to MSA followed by TUN then SYR, while ALG and ANB are the farthest. This experiment also highlights the closeness between Algerian dialects (ALG and ANB) and Levantine dialects (PAL and SYR). It shows also That TUN is closer to PAL and to SYR than ALG and ANB.

Table 6. Overlapping of vocabularies between Dialects and MSA

	ALG	ANB	TUN	SYR	PAL
MSA	0.12	0.10	0.16	0.14	0.21
PAL	0.13	0.11	0.17	0.21	
SYR	0.09	0.09	0.13		
TUN	0.16	0.13			
ANB	0.32				

4.3 Measuring the Cross-Language Divergence

In this section, we are interested by measuring the divergence between dialects and MSA throw unigram language models. For this purpose we choose to use the Hellinger Distance (HD) [12][13], a measure of distributional divergence. It quantifies the similarity between two probability distributions. It has been used to detect failures in classification performance [14] and in machine learning it is used to estimate the class distribution [15]. In [16], this distance was used to measure information loss in data protection. Hellinger distance is symmetric and non-negative, and obeys to triangle rule.

In order to measure the divergence between two languages with HD, let consider a bi-text(L_i, L_j) with the vocabularies V_i and V_j respectively. We constitute V, a vocabulary including 10K words including the common words between V_i and V_j and from the remaining words of the two vocabularies we include the most frequent ones to complete V. For each side of the bi-text, a unigram probability distribution $P(w|L_i)$ is computed over V. To avoid zero probabilities due to the words not belonging to the considered language, we decided to smooth the probabilities. The comparison of the two distributions is then calculated as follows:

$$HD(L_i, L_j) = \sqrt{\frac{1}{2} \sum_{w \in V} (\sqrt{P(w|L_i)} - \sqrt{P(w|L_j)})^2} \quad (2)$$

Table 7 draws HD values computed between all dialects and MSA. These values show that PAL is the closest dialect to MSA followed by TUN then SYR, whereas ALG and ANB are the most divergent. The closest dialects according to HD are ALG and ANB and also PAL and SYR. The closest dialect to MSA is PAL and the farthest are ALG and ANB. Another interesting and expected result is the one related to the distance between TUN and the other dialects, TUN is closer to ALG and ANB than to PAL and SYR.

Table 7. Hellinger distance values for the different pairs of languages

	ALG	ANB	TUN	SYR	PAL
MSA	0.72	0.72	0.60	0.62	0.55
PAL	0.85	0.86	0.81	0.76	
SYR	0.84	0.86	0.81		
TUN	0.79	0.80			
ANB	0.73				

5 Dialect Identification

In this section, we deal with the issue of using several languages in the same sentence. This is very common in Arabic world and especially in Maghreb. This phenomenon is commonly named code switching. Arabic people often switch between several languages. For instance, in Algeria, people could switch from dialect, to MSA to French. In the following, French will not be taken into account. To identify the different languages in order to apply the appropriate tools, we consider the identification of language such as a classification issue which will be treated in the following by a Naive Bayes classifier (NBC). NBC is probabilistic learning algorithm, it is used for many issues in NLP [17–19]. A naive Bayes classifier assumes that all features representing a given problem are conditionally independent given the value of classification variables.

For our purpose, NBC is based on 3-grams features. Given n classes corresponding to n languages, the purpose is to assign the most suitable class C_i in accordance to a set of features $F = \{f_1, ..., f_n\}$ which maximizes the conditional probability:

$$p(C_i \mid f_1, ..., f_n) = p(C_i) \prod_{j=1}^{n} p(f_j|C_i) \tag{3}$$

where $p(C_i)$ is the probability of the class C_i and $p(f_j|C_i)$ is the conditional probability of the feature f_j observed with the class C_i.

For the experiment, we created a corpus by merging MSA, ALG, ANB, TUN, SYR and PAL for which each sentence is annotated by its corresponding language class. By selecting randomly 80% of the data, we create the training corpus and the remaining has been dedicated for test. Classification results in Table 8 show that the recall for MSA is the highest one (75%); this could be explained by the fact that MSA writing obeys to strict rules contrary to dialects for which no formal writing rules exist: a dialect word could be written in different forms which are all acceptable. Consequently, this phenomenon generates a larger distribution probability for dialects than MSA ones.

Table 9 draws the confusion matrix of the classifier. For dialect side, it is clearly shown that the highest confusion rates are those between ALG and ANB and between PAL and SYR, this confusion is justified by the closeness between these pairs of dialects; ALG and ANB for example share an important vocabulary in spite of their difference. For MSA side, it is shown that the highest confusion

Table 8. Dialect identification results using the parallel corpus

Language	Precision	Recall	F
ALG	0.48	0.50	0.49
ANB	0.49	0.49	0.49
TUN	0.68	0.52	0.59
SYR	0.62	0.55	0.58
PAL	0.53	0.57	0.55
MSA	0.64	0.75	0.69

Table 9. Confusion matrix rates for dialect identification using the parallel corpus

	Estimated language classes					
True language classes	ALG	ANB	TUN	SYR	PAL	MSA
ALG	**50**	35	4	6	2	4
ANB	38	**49**	5	2	3	3
TUN	12	8	**52**	6	9	14
SYR	3	3	4	**55**	24	11
PAL	4	3	4	16	**57**	17
MSA	2	2	4	5	12	**75**

rates related to MSA are those with PAL, whereas for ALG and ANB dialects, confusion rates related to MSA do not exceed 4% for both dialects.

6 Machine Translation

Arabic language translation has been widely studied. The rich morphology of Arabic is seen as a rocky barrier in building efficient translation systems. Indeed, Arabic is characterized by complex a morphology and a rich vocabulary. It is a derivational, flexional and agglutinative language. We recall that, in order to compare it to English, an Arabic word (or more rigorously a lexical entry) can sometimes correspond to a whole English sentence [20].

Moreover, Arabic words are often ambiguous because a single word could have multiple morphological analyses. This is due to the richness of the Arabic affixation and the omission of short vowels. In addition, articles, prepositions, pronouns, etc. can be affixed to adjectives, nouns, verbs and particles to which they are related. All these phenomena increase the ambiguity and make the traditional issues of NLP more challenging such as machine translation from and to Arabic.

As shown in the previous experiments, dialects even if they are inspired strongly from Arabic, the significant differences may prevent communication between people of Arabic world. That is why, it is very important to propose machine translation between different dialects and MSA. In the following, we present several experiments in order to develop machine translation between Arabic dialects and MSA. For each pair of languages we used a parallel corpus

Table 10. BLEU score of Machine Translation on different pairs of languages using two smoothing techniques

	Target											
	ALG		ANB		TUN		SYR		PAL		MSA	
Source	KN	WB	KN	WB	KN	WB	KN	WB	KN	WB	KN	WB
ALG	-	-	61.06	**60.81**	**9.67**	9.36	7.29	**7.95**	**10.61**	10.14	**15.1**	14.64
ANB	**67.31**	65.55	-	-	**9.08**	8.64	7.52	**7.95**	**10.12**	9.84	**14.44**	13.95
TUN	**9.89**	9.48	**9.34**	9.01	-	-	**13.05**	12.93	**22.55**	22.21	**25.99**	25.21
SYR	**7.57**	7.50	7.50	**7.64**	**13.67**	13.23	-	-	**26.60**	25.74	**24.14**	22.96
PAL	**11.28**	10.67	**9.53**	9.15	**17.93**	16.64	**23.29**	23.07	-	-	**40.48**	39.76
MSA	**13.55**	13.05	**12.54**	11.72	20.03	**20.44**	**21.38**	20.32	**42.46**	41.37	-	-

of 6400 sentences (5900 have been dedicated to training and the remaining for tests).

All the MT systems we used are phrase-based [21] with default settings: bidirectional phrase and lexical translation probabilities, distortion model, a word and a phrase penalty and a trigram language model. We used GIZA++ [22] for alignment and SRILM toolkit [23] to compute trigram language models. Since the parallel corpus is small, we experimented the Kneser-Ney and Witten-Bell smoothing techniques hoping to identify the one which best fits. The results conducted on the test set are presented in terms of BLEU in Table 10. This experiment leads to very interesting conclusions. First of all, for small parallel corpus, it seems that the smoothing technique has an impact on translation results. A difference of almost 2 points in terms of BLEU scores has been observed for translating from ANB to ALG. But, we can not generalize by affirming that one smoothing technique is definitely better than another. High score of translation has been achieved between ANB and ALG in both sides. This result is natural since these two dialects are used in the same country and share up to 60% of words. Almost the same observation is made for the pair SYR and PAL since these two dialects belong to the same language family (Levantine).

Another interesting and expected result is BLEU score between MSA and dialects. In fact, the highest one is related to PAL in both sides since this dialect is the closest one to MSA as shown in other experiments of sections 4.2 and 4.3. Most surprising results are those relative to SYR and TUN. It seems that it is easier to translate TUN to MSA than SYR to MSA. Also, translating from MSA to TUN gives better results than from MSA to the Algerian dialects. In the symmetric side of translation we get the same scale of results. This definitely shows the closeness of TUN dialect to MSA in comparison to the Algerian dialects.

7 Conclusion

In this paper we present an analytical study of Arabic dialects from Middle-east and Maghreb. Maghreb's dialects use several words from French, Turkish, ... and adapted them phonetically so they become full words of these dialects.

In the opposite, Middle-east dialects are more close to MSA because they share an important vocabulary with it.

To make this research and study possible, we started from scratch because for these vernacular languages, there is no available written resources. We build a parallel corpus including 5 dialects (two from Algeria, one from Tunisia, and the two others from Middle-east: Palestine and Syria) and MSA. We perform different experimentations in order to study the relationship between MSA and dialects on one hand and cross-dialects on the other hand. For this, we calculated the overlapping between each pair of vocabularies. We then calculated the distance between the distributions of each pair of languages in order to measure which language is closer to which one. The carried out results are consistent with the fact that Middle-East Arabic dialects are closer to MSA than those of the Maghreb. This has been confirmed by the other experiments of identification handled by machine learning techniques. We showed that it is easier to identify MSA than dialects because it is a natural language with the whole standard linguistic constraints. Concerning the experience on identification, the results could be separated into two classes. The first one concerns ALG and ANB and the other one the three other dialects. In fact for this last class, the F-measure results are close and the difference between them are not statistically significant. This means that it is easier to identify PAL, SYR and TUN than Algerian dialects.

We conducted also several experiments of machine translation between all the pairs of languages. We took advantage from this experiment to try to understand whether the smoothing techniques could have an impact on BLEU score when we are faced to small corpora. We remarked that in some cases, the used method could improve BLEU significantly. High score of translation has been achieved between ANB and ALG in both sides. This result is natural since these two dialects are used in the same country and share up to 70% of words. In the near future, we will extend this work to other dialects and will propose a speech to speech system which is the main objective of this work.

References

1. Kilany, H., Gadalla, H., Arram, H., Yacoub, A., El-Habashi, A., McLemore, C.: Egyptian Colloquial Arabic Lexicon. In: LDC Catalog Number LDC99L22 (2002)
2. Kirchhoff, K., Bilmes, J., Das, S., Duta, N., Egan, M., Ji, G., He, F., Hopkins, J., Liu, D., Noamany, M., Schone, P., Schwartz, R., Vergyri, D.: Novel Approaches to Arabic Speech Recognition: Report from the, Johns-Hopkins Summer Workshop. In: Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, pp. 344–347 (2002)
3. Habash, N., Rambow, O.: Magead: A Morphological Analyzer and Generator for the Arabic Dialects. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 681–688 (2006)
4. Chiang, D., Diab, M., Habash, N., Rambow, O., Shareef, S.: Parsing Arabic Dialects. In: Proceedings of the European Chapter of ACL (EACL). (2006)

5. Zbib, R., Malchiodi, E., Jacob, D., Stallard, D., Matsoukas, S., Schwartz, R., Makhoul, J., Zaidan, O., Callison-Burch, C.: 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, NAACL HLT 2012, pp. 49–59 (2012)
6. Salloum, W., Habash, N.: Dialectal Arabic to English Machine Translation: Pivoting through Modern Standard Arabic. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL HLT 2013, pp. 348–358 (2013)
7. Zaidan, O., Callison-Burch, C.: Arabic Dialect Identification. Computational Linguistics 40, 171–202 (2014)
8. Elfardy, H., Diab, M.: Sentence Level Dialect Identification in Arabic. In: ACL (2), pp. 456–461 (2013)
9. Bouamor, H., Habash, N., Oflazer, K.: A Multidialectal Parallel Corpus of Arabic. In: Proceedings of the Language Resources and Evaluation Conference, LREC 2014, pp. 1240–1245 (2014)
10. Meftouh, K., Bouchemal, N., Smaili, K.: A Study of a Non-resourced Language: an Algerian Dialect. In: Third International Workshop on Spoken Languages Technologies for Under-resourced Languages, pp. 125–132 (2012)
11. Skadiņa, I., Aker, A., Giouli, V., Tufis, D., Gaizauskas, R., Mieriņa, M., Mastropavlos, N.: A Collection of Comparable Corpora for Under-resourced Languages. In: Proceedings of the 2010 Conference on Human Language Technologies – The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT 2010, pp. 161–168 (2010)
12. Kailath, T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Transactions Communication Technology 15, 52–60 (1967)
13. Rao, C.R.: A Review of Canonical Coordinates and an Alternative to Correspondence Analysis Using Hellinger Distance. Quaderns Estadistica i Investig Ope, Questiio 19, 23–63 (1995)
14. Cieslak, D.A., Chawla, N.V.: A Framework for Monitoring Classifiers Performance: When and Why Failure Occurs? Knowledge and Information Systems, 83–109 (2009)
15. González-Castro, V., Alaiz-Rodríguez, R., Alegre, E.: Class Distribution Estimation Based on the Hellinger Distance. Information Sciences, 146–164 (2013)
16. Torra, V., Carlson, M.: On the Hellinger Distance for Measuring Information Loss in Microdata. In: Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (2013)
17. Pop, I.: An Approach of the Naive Bayes Classifier for the Dcument Classification. General Mathematics 14(4), 135–138 (2006)
18. Pedersen, T.: A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. In: Proceedings of 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics, pp. 63–69 (2000)
19. Ahmed, F., Nurnberger, A.: Arabic/English Word Translation Disambiguation Using Parallel Corpora and Matching Schemes. In: 12th EAMT Conference, pp. 6–11 (2008)
20. Badr, I., Zbib, R., Glass, J.: Segmentation for English-to-Arabic Statistical Machine Translation. In: Proceedings of the ACL 2008 Conference Short Papers, pp. 153–156 (2008)

21. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Demonstation Session, pp. 177–180 (2007)
22. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics 29(1), 19–51 (2003)
23. Stolcke, A.: Srilm – an Extensible Language Modeling Toolkit. In: ICSLP, Denver, USA, pp. 901–904 (2002)

Language Set Identification in Noisy Synthetic Multilingual Documents

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen

The University of Helsinki
Department of Modern Languages, Helsinki, Finland
{firstname,lastname}@helsinki.fi

Abstract. In this paper, we reconsider the problem of language identification of multilingual documents. Automated language identification algorithms have been improving steadily from the seventies until recent years. The current state-of-the-art language identifiers are quite efficient even with only a few characters and this gives us enough reason to again evaluate the possibility to use existing language identifiers for monolingual text to detect the language set of a multilingual document. We are using a previously developed language identifier for monolingual documents with the multilingual documents from the WikipediaMulti dataset published in a recent study. Our method outperforms previous methods tested with the same data, achieving an F_1-score of 97.6 when classifying between 44 languages.

1 Introduction

The method presented in this article has been developed as a part of the Kone Foundation funded project The Finno-Ugric Languages and the Internet[1]. The project has a need for word-level language identification between 300+ languages, including some closely related languages, as the project aims to gather texts written in small Uralic languages from the Internet. So far the project has downloaded and identified the language of several thousand million files, most of which are multilingual to some extent. The language identifier currently in use [1, 2] is capable of correctly handling only monolingual files, which means that text sections in small Uralic languages between text in other languages may not have been found. As an example of a multilingual text, we present a line from Finnish Wikipedia in Fig. 1. The example includes 7 words in Finnish and 6 words in Latin.

Multilingual language identification for corpora creation purposes has earlier been studied by Ludovik and Zacharski [3]. Multilingual language identification is also needed for automatic processing of multilingual documents in general, for example machine translation or information retrieval [3–10]. Stensby et al. [11] considered the problem of detecting the language while it is being written.

[1] http://suki.ling.helsinki.fi

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 633–643, 2015.
DOI: 10.1007/978-3-319-18111-0_48

Aasiankultakissa, (*Catopuma temminckii* eli *Profelis temminckii* eli *Felis temminckii*) on Kaakkois-Aasiassa elävä kissaeläin.
'Asian golden cat, (*Catopuma temminckii* or *Profelis temminckii* or *Felis temminckii*) is a cat living in South-East Asia.'

Fig. 1. Multilingual example from Finnish Wikipedia of a sentence in Finnish and Latin with the English and Latin gloss in quotes

Automated methods for language identification have been improving steadily from the seventies until recent years. The current state-of-the-art language identifiers are quite efficient even with only a few characters and this gives us enough reason to evaluate the possibility of using existing language identifiers for monolingual text to detect the language set of a multilingual document.

2 Earlier Work

Here we briefly review the work already done in multilingual document identification. In 1995, Giguet [12] categorized sentences within multilingual documents. He managed to achieve 99.4% correct classification of sentences between 4 languages. In 1999, a vector-spaced categorizer called Linguini was presented by Prager [4]. Linguini identifies the languages and their proportions for the whole document and his method was evaluated by Lui et al. [10], the results of which are found later in this article. Also in 1999, Ludovik and Zacharski [3] segmented multilingual documents between 34 languages. Their 6 documents were artificially created and they each contained all the languages, so their task was not to detect the language set of a document, but to segment it according to the languages. Teahan considered segmenting multilingual text in 2000 [13]. He was using PPMD models for six languages. In 2006, the problem of multilingual web-documents was researched by Mandl et al. [6]. They were trying to identify which of the 8 languages known by the language identifier the text was written in by using a sliding window of 8 words. Their method reached 97% accuracy. Multiple language web pages were considered by Rehurek and Kolkus [14]. They evaluated their method with single sentences in 9 languages. Romsdorfer considered language identification with multilingual text-to-speech synthesis [15] [16]. In 2010, Murthy and Kumar [7] classified small text samples between two Indian languages. Also in 2010, Stensby et al. [11] classified multilingual documents between 9 languages with 97% average accuracy. Word level language identification in online multilingual communications was considered by Nguyen and Dogruöz [17]. They classified words between two languages, Turkish and Dutch, with up to 98% accuracy. Yamaguchi and Tanaka-Ishii [18] addressed the problem of segmenting multilingual text into language segments. Their method was also evaluated by Lui et al. [10]. In 2013, King and Abney [19] considered the problem of directly labeling the language of words between 31 languages. More recently, the problem was tackled by King et al. [9]. Their method achieves the highest accuracy of 89.94% when using 5-grams for classifying between 2 languages: English and Latin. Lui et al. [10] concentrated on identifying the presence of different languages in multilingual documents from

a set of 44 languages, achieving the F_1-score of 95.9 on document-level. A new masters thesis on the subject was published in 2014 by Ullman [20], who experimented with multilingual documents in 5 languages.

Generally, the results of the previous studies can not be directly compared with each other, as the test setups differ considerably. The set of possible languages is usually different in size as well as in the selection of individual languages. The way the test corpora are generated or annotated is usually different, each containing language segments of different sizes. Lui et al. [10] created an openly available corpus, WikipediaMulti, for evaluating multilingual language identification. They used it to evaluate two previously introduced methods [4, 18] as well as their own. In order to provide comparable results we opted to utilize this same corpus[2] in the evaluation of the method proposed in this article.

3 Proposed Method

The proposed method is built on the idea of using already existing monolingual language identifiers in trying to identify the set of languages of a multilingual document. The basic idea is simply to slide an overlapping byte window of size x through the document in steps of one byte. The text in each window is sent to a separate language identifier algorithm, which gives the most likely language for the window. There is a variable called CurrentLanguage, which is first given the language of the first byte window as its value. CurrentLanguage changes after z consecutive window identifications have given a differing language from the CurrentLanguage. The document is given a label for each language that has been the CurrentLanguage at some point when going through the document.

The idea of using a window approach in multilingual language identification was also proposed by Mandl et al. in 2006 [6]. However, they used the number of words as the size of the window and the language was changed each time a different language (from a selection of 8 languages) was identified for the window. When we are handling noisy documents or the number of languages to be identified is large, or we handle languages without white space breaks, we need to have a sliding window frame and several frames agreeing on the language change before actually changing the CurrentLanguage.

4 Test Setup

We are using WikipediaMulti, which is a synthesized corpora of multilingual texts made available by Lui et al. [10]. It consists of three parts each with 44 languages: 5000 monolingual documents for training, 5000 multilingual documents for development and another 1000 multilingual documents for testing. All the multilingual documents have been generated by randomly concatenating parts of monolingual documents together. A separate metadata-file is used for marking the languages which should be found in each document, together with

[2] Corpus can be found at "`http://people.eng.unimelb.edu.au/tbaldwin/#resources` " under the title "Multilingual language identification dataset".

their respective sizes. Example of the metadata can be seen in the Figure 2, where document id is followed by part number (twice), language code and the size of the part in bytes.

```
doc001,1,1,de,1177
doc001,2,2,tr,394
doc001,3,3,el,1015
doc001,4,4,ru,315
doc001,5,5,es,728
```

Fig. 2. Example metadata for a multilingual document from WikipediaMulti dataset

5 Evaluation

We trained a previously developed language identifier [1, 2] for the 44 languages in the WikipediaMulti dataset using the 5000 monolingual documents provided. The language identifier used has a few tunable parameters: the units used by the language identifier and their cut off in terms of their relative frequencies in the training material. The units we used are tokens and character n-grams from one to five, with a relative frequency of 0.0000005 as cut-off. The algorithm used by the language identifier is called token-based backoff. In the token-based backoff each token of the mystery text is given equal value when deciding the language of the whole text. The probabilities of languages for each token are calculated independently of the surrounding tokens and the average over the probabilities of all the tokens is used to determine the most likely language. Primarily the relative frequencies of tokens in the training corpus are used as probabilities, but when a previously unseen token is encountered the identifier backs off to using the relative frequencies of character n-grams.

We report the document-level averages of recall, precision and the F_1-score. Document-level averages are referred to as *micro-averages* by Lui et al. [10]. The F_1-score is calculated from the recall r and the precision p, as in (1).

$$F_1 = 2\left(\frac{pr}{p+r}\right) \tag{1}$$

We started the experiment by taking the first 100 bytes ($x = 100$) from the beginning of the document and identifying its language with the language identifier. The document was given a label with the language identified and the language was set as the CurrentLanguage. Then we moved forward one byte and sent the following 100 bytes to the language identifier, thus including 99 of the same bytes as the first one. We continued moving forward by one byte intervals until the end of the document. If the language identified differed from the CurrentLanguage 25 times in a row ($z = 25$), then the CurrentLanguage was set to the language identified last and the document was given a label with the identified language. We repeated the process to the end of the document. Giving

the document labels this way resulted in a recall of 99.36%, precision of 88.50% and the F_1-score of 93.6.

Then we started to increase the length of the text to be identified. As can be seen in the Table 1, the F_1-score started to decrease after the window reached 400 bytes in length ($x = 400$) as the recall was decreasing quicker than precision was increasing.

Table 1. Recall, precision and F_1-score with differing length of byte-window

x in bytes	z in times	Recall	Precision	F_1-score
100	25	99.36%	88.50%	93.6
200	25	99.12%	93.92%	96.4
300	25	98.72%	95.47%	97.1
400	25	98.33%	96.11%	97.3
500	25	97.77%	96.61%	97.2
600	25	97.27%	96.98%	97.1

Next step was to try to optimize z, the number of times the identification had to differ, with the text length x of 400. The results of these experiments can be seen in the Table 2. The F_1-score was clearly decreasing both directions from z being 100. Our best results on the development set were achieved using x of 400 and z of 100.

Table 2. Recall, precision and F_1-score with byte-window of 400

x in bytes	z in times	Recall	Precision	F_1-score
400	200	97.13%	97.54%	97.3
400	100	97.83%	97.08%	97.5
300	100	98.31%	96.68%	97.5
400	50	98.11%	96.55%	97.3
400	25	98.33%	96.11%	97.3
400	10	98.43%	95.64%	97.0

The F_1-score of 97.5 was higher than the 95.9 reported by Lui et al. [10], and had reached a local optimum. We decided to try our method on the test set. From the test set we got micro-average recall of 97.87%, precision of 97.41% and the F_1-score of 97.6 and macro-average recall of 97.86%, precision 97.66% and the F_1-score of 97.7. We have included the results from the other methods tested by Lui et al. [10] in Table 3. SegLang refers to a system by Yamaguchi and Tanaka-Ishii [18] and Linguini to a system by Prager [4].

5.1 Errors with the Test Set

We decided to take a closer look at the errors made by our system on the test set. Our F_1-score was already 97.6, which meant that there were not that many errors and we analyzed them all. These errors can be categorized in 6 different categories.

Table 3. Recall, precision and F_1-score with different methods

System	Recall	Precision	F_1-score
SegLang	97.5%	77.1%	86.1
Linguini	77.4%	83.8%	80.5
LLB	95.5%	96.3%	95.9
Proposed method	97.9%	97.4%	97.6

Segments Written in an Unlabeled Language. There were 32 documents where our language identifier had detected English as a language without it being in the list of language labels for that document. In 13 documents, English had completely replaced one of the languages indicated by labels and in 9 cases the segment labeled with non-English contained more English than the labeled language. Six documents contained more than 200 character English incursions in a labeled language. In two documents, the labeled language contained many English words. In one document, Spanish had completely replaced Indonesian. One document contained 274 byte incursion in Russian at the end of a Hebrew part and one had 1500 bytes of French after a Georgian part. One document (wikipedia-multi/docsUE1/doc058), labeled only as Italian, was in fact multilingual, being a Wikipedia article about a common Slavic song in Macedonian, Croatian, Slovenian, Bulgarian, Russian and Polish. Another document had only names of books in English and Spanish in the part labeled Malaysian. One Estonian part consisted mostly of words in an unknown language. It was identified as Slovenian, Portuguese and Croatian by the language identifier with the 44 language selection, and as Breton when identified with the language identifier with 285 languages.

The errors in this category cannot be considered as errors with language identification, but are, in fact, errors in the labeling of the test set. There are several shorter incursions in English, and maybe in other languages as well, in many of the documents. We adjusted the length x of our detection window according to the existing labels, which is why x grew so large that our language identifier no longer noticed the shorter incursions.

Extremely Close Languages. In these results, the most problematic close language pair was Indonesian and Malaysian. In 26 documents, they had been erroneously identified, most documents being labeled with both of the languages. The languages are highly similar as can be seen from the top 10 words in our training set for each language in the Table 4.

The differences in frequencies of these words are not language specific. They are rather the result of the topics and domains of the randomly selected articles. The frequency list for Malaysian again brings to focus the previous errors with the corpora used. The ninth most common word in Malaysian is actually an English word and is the result of large English incursions in the Malaysian training texts.

Table 4. The 10 most common words in Malaysian (ms) and Indonesian (id) in the training set

word	number in ms	word	number in id
dan	2182	yang	2698
yang	1952	dan	2436
di	1368	di	1577
pada	870	dengan	1129
dengan	796	untuk	945
untuk	702	pada	929
dalam	681	dari	841
ini	614	dalam	754
the	579	ini	689
oleh	529	itu	631

In one document the beginning of Galego part was identified as Portuguese. Once the beginning part of Norwegian segment was identified as Danish. These languages are relatively close to each other, but much farther away than the Indonesian - Malaysian pair.

More than One Writing System for a Language. The Azeri language can be written using either an Arabic or Latin character set. The training partition for Azeri was mostly in Latin characters, which resulted in Azeri written with Arabic characters sometimes to be identified as Farsi. This could be corrected by creating two different language models for Azeri, one with Latin characters and another with Arabic characters.

Segment Consisting Mostly of Non-alphabetic Characters. One document contained a segment labeled as Macedonian, which consisted of hundreds of numbers and only less than 20 tokens in Cyrillic and another 20 tokens in Latin characters. Macedonian is written with Cyrillic characters, hence the segment was erroneously identified to contain Romanian, Bulgarian, and Russian in addition to Macedonian. One Malaysian labeled part consisted only of lots of numbers together with some U.S. place names. In one document, there was, after Hindi in a Hindi labeled part, many dates in numbers together with abbreviations of English months.

Place Names and Lists of Abbreviations. Two documents had excessive numbers of foreign place names, which were identified with their respective languages. One Slovenian part contained a large list of unknown character combinations, which could have been some sort of model numbers or abbreviations. Place names and lists of part numbers have also proven to be especially troublesome in the language identification done while crawling web pages.

Very Short Segments of Labeled Language. There were 27 language segments from 15 to 164 bytes in length which were not identified correctly. Also 10 longer segments were incorrectly identified. It is clear that these segments, which were shorter than our 400 byte window, were too short for the language identifier to notice. It is probable that our byte window grew so large, because there is a greater number of incorrectly than correctly labeled short language segments within the development set.

6 Discussion

We also tested identifying the languages with previously generated language models [2]. We took a subset of 43 languages from the 285 languages we used in our evaluation of the monolingual language identifier and the results are on the second line of the Table 5. We had only one language for the Indonesian/Malaysian pair, so the results cannot be directly compared. In these tests we also used z of 50. We also tested the new method with the language identifier having 285 languages to choose from. The results can be seen in the Table 5. It is notable how little difference there is between the scores, even though the task of categorizing between 285 languages is a lot more challenging than between 43 languages. This reflects the great accuracy we achieved when evaluating our language identifier algorithm, it reached 100.0% in both recall and precision already at the test length of 120 characters with 285 languages.

Table 5. Recall, precision and F_1-score with different language models using the proposed method

System	Recall	Precision	F_1-score
Language models from [2], 43 languages	98.30%	97.55%	97.9
Language models from [2], 285 languages	98.27%	97.33%	97.8

In order to provide a working prototype we tested the proposed method with our own implementation of the Cavnar & Trenkle algorithm [21] for language identification. We used language models generated from the WikipediaMulti training set and the number of n-grams in each of the language models was 20000. The language identifier using the Cavnar & Trenkle algorithm doesn't achieve as high F_1-scores as the one using our own algorithm [2], but it still outperforms the one proposed by Lui et al. [10]. It attains F_1-score of 96.2 when using 400 byte window and a threshold z of 100. We also tested with the same window, but jumping every other byte when moving the window with reduced threshold z of 50. Jumping every other byte halves the time used for identifications, with only a small drop in F_1-score. The working prototype using the Cavnar & Trenkle algorithm can be downloaded from our web page[3].

[3] `http://suki.ling.helsinki.fi/MultiLI`

Table 6. Recall, precision and F_1-score with language identifier using the Cavnar & Trenkle algorithm and language models from WikipediaMulti

System	Recall	Precision	F_1-score
C & T algorithm with 20000 n-grams, no jump	97.23%	95.11%	96.2
C & T algorithm with 20000 n-grams, jump 2 bytes	97.27%	94.68%	96.0

Two thirds of the total amount of errors made by our system were directly or indirectly caused by incorrect labeling of languages in the test set. With the quality of the development and the test material at hand, we did not think it would be sensible to continue to token-level identifications. We will need a more precise dataset for that task. It would be easy and quite quick to find at least the most problematic unlabeled segments from the WikipediaMulti dataset using the method presented in this paper, but it wouldn't be correct to use the new derived dataset for the evaluation of at least the method itself.

When setting up a multilingual identification system, it is important to decide the minimum length for the text to be identified. If we are interested in loanwords, we might want to investigate character sequences shorter than tokens, if we are interested in foreign words used inside the sentences we might want to use tokens as the length and, if we are interested in sentences, the length should be set to a sentence. If we want to create a language corpus to research character combinations in a certain language (for example when calculating distances between languages), we might not want the foreign words polluting the language we are interested in. One of our next tasks will be to find or to create a more precisely labeled multilingual corpus for experiments with token-level language identification.

7 Conclusions

We have presented a simple method to identify the language set of multilingual documents. The method uses existing language identifier designed for monolingual texts. We evaluated the method using a corpus designed for multilingual language identifier evaluation. The method presented in this article clearly outperforms the methods previously evaluated with the same corpus, reducing the average recall error by 53% and the average precision error by 30% when compared to the previously best method.

Acknowledgments. This work was supported by Kone Foundation from its language programme[4]. We also thank Timothy Baldwin and Marco Lui for their help with the WikipediaMulti dataset.

References

1. Jauhiainen, T.: Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki (2010)

[4] `http://www.koneensaatio.fi/en/grants/language-programme/`

2. Jauhiainen, T., Lindén, K.: Identifying the language of digital text. In review, submitted 08/14 (2015)
3. Ludovik, Y., Zacharski, R.: Multilingual document language recognition for creating corpora. Technical report, New Mexico State University (1999)
4. Prager, J.M.: Linguini: Language identification for multilingual documents. In: Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, Maui (1999)
5. Ozbek, G., Rosenn, I., Yeh, E.: Language classification in multilingual documents. Technical report, Stanford University (2006)
6. Mandl, T., Shramko, M., Tartakovski, O., Womser-Hacker, C.: Language identification in multi-lingual web-documents. In: Natural Language Processing and Information Systems. Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems, Klagenfurt (2006) 153–163
7. Murthy, K.N., Kumar, G.B.: Language identification from small text samples. Journal of Quantitative Linguistics **13** (2006) 57–80
8. Hughes, B., Baldwin, T., Bird, S., Nicholson, J., MacKinlay, A.: Reconsidering language identification for written language resources. In: Proceedings of the International Conference on Language Resources and Evaluation, Genoa (2006) 485–488
9. King, L., Kübler, S., Hooper, W.: Word-level language identification in The Chymistry of Isaac Newton. Literary and Linguistic Computing (2014)
10. Lui, M., Lau, J.H., Baldwin, T.: Automatic detection and language identification of multilingual documents. Transactions of the Association for Computational Linguistics **2** (2014) 27–40
11. Stensby, A., Oommen, B.J., Granmo, O.C.: Language detection and tracking in multilingual documents using weak estimators. In: Proceedings of the Joint IAPR International Workshop of SSPR&SPR. Volume 6218 of NLCS., Cesme, Springer, Heidelberg (2010) 600–609
12. Giguet, E.: Multilingual sentence categorization according to language. In: Proceedings of the European Chapter of the Association for Computational Linguistics SIGDAT Workshop "From text to tags : Issues in Multilingual Language Analysis", Dublin (1995) 73–76
13. Teahan, W.J.: Text classification and segmentation using minimum cross-entropy. In: Proceedings of the 6th International Conference "Recherche d'Information Assistee par Ordinateur", Paris (2000) 943–961
14. Řehůřek, R., Kolkus, M.: Language identification on the web: Extending the dictionary method. In: Proceedings of the 10th International CICLing Conference. Volume 5449 of NLCS., Springer, Heidelberg (2009) 357–368
15. Romsdorfer, H., Pfister, B.: Text analysis and language identification for polyglot text-to-speech synthesis. Speech communication **49** (2007) 697–724
16. Romsdorfer, H.: Polyglot text-to-speech synthesis. PhD thesis, Swiss Federal Institute of Technology, Zürich (2009)
17. Nguyen, D., Dogruöz, A.S.: Word level language identification in online multilingual communication. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Seattle (2013) 857–862
18. Yamaguchi, H., Tanaka-Ishii, K.: Text segmentation by language using minimum description length. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Jeju Island (2012) 969–978
19. King, B., Abney, S.: Labeling the languages of words in mixed-language documents using weakly supervised methods. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta (2013) 1110–1119

20. Ullman, E.: Shibboleth - a multilingual language identifier. Master's thesis, Uppsala University, Uppsala (2014)
21. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas (1994) 161–175

Feature Analysis for Native Language Identification

Sergiu Nisioi[1,2]

[1] Center for Computational Linguistics,
University of Bucharest
[2] Oracle RightNow,
Bucharest, Romania
sergiu.nisioi@gmail.com

Abstract. In this study we investigate the role of different features for the task of native language identification. For this purpose, we compile a learner corpus based on a subset of the EF Cambridge Open Language Database - EFCAMDAT [10] developed at the University of Cambridge in collaboration with EF Education. The features we are taking into consideration include character n-grams, positional token frequencies, part of speech n-grams, function words, shell nouns and a set of annotated errors. Last but not least, we examine whether the essays of English learners that share the same mother tongue can be distinguished based on their country of origin.

1 Introduction

The concept of *interlanguage* first proposed by Selinker [27] proved to be essential in understanding the means through which adults acquire a second language. The term currently describes the entire linguistic system that emerges when second language learners - both child and adult - express meaning in the target language [26]. Interlanguage is usually regarded as a separate linguistic system that is different from the target language (TL) and the learner's mother tongue.

The main objective of native language identification (NLI) resides in the analysis and classification of texts that belong to specific groups of learners. Although the classes are usually determined by learners' mother tongues, in our study we label the documents based on the country from which the learners originate. This being the case, we pretermit the different dialects or minority languages that are spoken within a country.

Our study is focused on English texts belonging to learners originating from different geographic regions. Relying upon the existing psycholinguistic studies on the phenomenon of interlanguage, we first investigate and analyze the features that can be used for automatic text classification. In addition to the standard features used in literature [29, 32], we further suggest the use of shell nouns [25] as interlanguage markers. In our first set of experiments we train a classifier to identify the country of origin corresponding to each text. The results obtained,

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 644–657, 2015.
DOI: 10.1007/978-3-319-18111-0_49

further confirm previous NLI studies [16, 29, 31] and methodologies to automatically detect the native language of an individual based on his writing. In addition, in the later sets of experiments we construct a classifier to distinguish between English texts of learners sharing a similar or identical mother tongue but whose countries of birth are different. For example, learners from Spain and learners from Argentina may share the same native language (Spanish) but their linguistic backgrounds are different - cultural and social norms and possibly distinct curricula of learning English can contribute to the way learners acquire English.

We define the linguistic background of a learner as the entire set of linguistic input he was subjected to so far. The linguistic input can consist of previous languages learned (others than English, including the mother tongue), previous methodologies and curricula followed in order to acquire those languages and all the possible interactions the learner might have had with native English speakers or in native English communities. What is more, political and cultural factors can also act upon the linguistic background.

We hypothesize the existence of a thinner line of delimitation between different speakers of the same native language (such as Spanish for Colombia and Spain or German for Austria and Germany). In comparison, learners having different native languages (Korean and Japanese or Telugu and Hindi), belonging to related linguistic backgrounds are likely to go through similar developmental processes when acquiring a *foreign* language.

2 Previous Work

One of the first multi-class native language identification studies [17] was conducted on the International Corpus of Learner English (ICLE) [11]. The different distribution and diversity of topics proved to be a disadvantage when evaluating cross-validated results [1]. TOEFL-11 is a learner corpus used for the 2013 NLI shared task [29], generally regarded as a better choice, the topics being similar and uniformly distributed across different learners. As Jarvis et al. [13] point out, the corpus lacks a uniform distribution of proficiency levels (low, medium, high) per native language, for example, only 1.4% of the texts coming from native German students are low proficiency texts.

A broad set of features and machine English teaching methods have been tested for the native language detection task [29–31]. Jarvis et al. [13] experiment with an L2-regularized L2-loss support vector machine [8] in combination with a mix of word n-grams, POS n-grams and lemma n-grams. In total they used around four hundred thousand features to achieve the best classification results for the NLI shared task [29]. Another approach that proved to output good results is based on a large spectrum of character n-grams. Ionescu et al. [12] combine a kernel machine with character n-grams to efficiently compute similarities when the space of features grows exponentially. Other high performance systems in the shared task [29] also used large numbers of character n-grams for classification. In cases like this, the features can cover topic related aspects

or even particular named entities (learners from Germany referring to German language, names or locations) and greatly outnumber the training/testing examples. Therefore, the results are usually difficult to interpret in terms of the psycholinguistic processes that shape the interlanguage of a learner.

3 Corpora

The corpus used in our study is based on a subset from the EF Cambridge Open Language Database - EFCAMDAT [10] developed at the University of Cambridge in collaboration with EF Education. The corpus consists from texts of various proficiency levels, submitted at Englishtown, the online school of EF Education [7].

The size of our extracted corpus has a total 18 million tokens and has essays collected from learners of 29 different countries: Argentina, Austria, Belgium, Brazil, Chile, People's Republic of China (PRC), Republic of China (Taiwan), Colombia, Costa Rica, Egypt, France, Germany, Indonesia, Italy, Japan, Kuwait, Mexico, Peru, Portugal, Russia, Saudi Arabia, South Korea, Spain, Switzerland, Thailand, Turkey, Ukraine, United Arab Emirates and Venezuela.

Out of the entire set of extracted texts, we have compiled the following corpora :

B13: learner texts from every unit in each level from one to six
- thirteen different countries: Brazil, Turkey, Italy, Mexico, People's Republic of China (PRC), France, Germany, Saudi Arabia, Colombia, Japan, Taiwan, Russia, South Korea
- for each country the sentences are merged and split into chunks of 1000 tokens
- each class is represented by the same number of chunks
- the total size of the corpus is approximately two million tokens

LB_Lang: groups of corpora to study the linguistic background hypothesis
- level one to six texts grouped from English learners that share similar mother tongues
- for each country the sentences are merged and split into chunks of 500 tokens
- each class is represented by the same number of chunks
- the term "Lang" is used to describe the native language
- (Table 1) contains the size and countries included for each language

For each level we have selected only learners who completed every unit to ensure as much as possible a uniform spread of topic and proficiency. The units found in level one are fairly basic and cover topics like greetings, family, jobs, describing people, food and drinks, etc. We could not control for a uniform distribution of levels across the documents from each country and we assume that a certain bias may have been introduced.

Among these topics, learners are required to describe facts about their place of birth which can reveal the first language of the learner.

Table 1. The corpora used to investigate the linguistic background hypothesis

Corpus	Countries	Total nr. of tokens
LB_Ar	Egypt, Kuwait, Saudi Arabia, United Arab Emirates (UAE)	174,000
LB_Ch1	People's Republic of China (PRC), Taiwan	1,000,000
LB_Ch2	Hong Kong (HK), PRC, Taiwan	132,000
LB_Ge	Austria, Germany, Switzerland	102,000
LB_RuUk	Russia, Ukraine	66,000
LB_Sp	Argentina, Colombia, Costa Rica, Mexico, Peru, Spain, Venezuela	199,500

We used the Stanford named entity recognition (NER) tool [9] trained on the CoNLL 2003 shared task data [24] to remove locations, person names, organizations and misc entities found in the texts. Moreover, we removed language names that were not identified by the NER system to avoid having biased classifications when using character n-grams. If a speaker claims to know Italian, then it's more likely for him to have a European mother tongue.

The corpus also comes with manual annotations of errors [10] together with the corrected alternatives. The most common type of errors encountered are the misuse of punctuation, capitalization and spelling errors. In total, there are 23 types of annotated errors [10], the least common are expressions of idioms, the use of possessive and the use of singular.

Each sub-corpus is extracted for distinct experiments, apart from this, it contains similar error annotations which allows us to have results consistent across various experiments. The extracted, pre-processed corpus is freely available at request from the author.

4 Features of Interlanguage

Chomsky [2] traces the starting point of language learning to a simple, basic (universal) grammar to which all language users have access. Strong similarities were observed between the sequential process of acquiring a mother tongue and a second language: *pidgin*, *baby talk*, *simplified registers* [21]. There is also an important distinction between the two learning processes: children always succeed in completely acquiring their native language, but adults only very rarely succeed in completely acquiring a second language [28]. The notion of *fossilization* as defined by Selinker and Rutherford [26] designates the permanent cessation of TL learning before the learner has attained the TL norms at all levels of linguistic structure.

Furthermore, the speed of learning a second language is highly correlated with the mother tongue [3]. The shared grammatical similarities between NL and TL can facilitate or impede the learning process. A so called positive language transfer phenomenon can intervene, *facilitating* a more rapid discovery of mother tongue-like features in the target language. A negative language transfer can also occur when a learner wrongly applies already acquired grammatical rules from his native language to express meanings in the TL. *Language transfer* is a key phenomenon that shapes the form of interlanguage.

In order to acquire a second language, Selinker [27] hypothesized that adults make use of a latent psychological structure, i.e. an already formulated arrangement in the brain, which is activated whenever an adult or child tries to produce meaning in the TL. The psycholinguistic processes of this structure together with brief examples are further provided:

native language transfer	NL-specific syntactic structures combined with target language words
overgeneralization / simplification	learners have the tendency to extensively use already acquired TL rules; for example, the use of past tense marker "-ed" for all verbs
transfer of training	e.g. fossilization can occur more rapidly for *street* learners compared to *classroom* learners [33], the former may successfully communicate to suit their needs albeit with lexical and syntactic errors
strategies of communication	resorting to more general nouns ("kind of", "sort", "thing") when the TL word is not known; the use of anaphoric shell nouns
strategies of learning	particular to learner: use of mnemonics, associations with cognates

5 Classification Approach

5.1 Classification Features

For classifying documents, we experiment with different features to cover every psycholinguistic aspect of interlanguage.

POS n-grams: part of speech bigrams and trigrams
character n-grams: bigrams, trigrams, 4-grams
function words: the closed class of words for English - connectors, determiners, particles, prepositions, adverbs, etc.
shell nouns: anaphoric nouns used to encapsulate more complex pieces of information [14, 25] - "fact", "thing", "task", "goal", "act", etc.

errors: manually annotated errors available in the EFCAMDAT corpus [10]
positional token frequencies: tokens appearing on the first two and last three positions in each sentence [34]

To cover the language transfer phenomenon, we consider function words and POS n-grams to be good features for two main reasons. (1) These types of features are (as much as possible) topic independent, unlike character n-grams or positional tokens. (2) Native language syntactic chunks have a tendency to transfer and influence the interlanguage. Function words reveal syntactic constructs, they are used unconsciously to tie sentences and create meaning. Hence, they were successfully used in a wide variety of text classification tasks from authorship attribution and analysis of style [5, 15], gender identification [19] to translation studies [34]. Münte et al. [22] argue that different brain functions are used to process the closed class and the open class of words.

POS n-grams are a type of shallow syntactic chunks that can be used as an indicator of the learner's coherence [4]. Apart from this, in combination with function words, POS n-grams can be used to reconstruct the phylogenetic tree of language similarities from learner texts [23] or to increase the accuracy of native language identification [13, 31].

Neither of these features are completely topic-independent, for example, literary and argumentative essays often employ different types of syntactic constructs that influence the way documents are classified, a fact observed [1] on the ICLE corpus as well.

Character n-grams have been successfully used for the task of NLI before, either in combination with words [13] or as standalone features in a kernel machine [12]. Widely used [20, 29], character n-grams have the advantage of covering language transfer and overgeneralization by encompassing both syntactic and morphologic features. The main drawback relies in the content it covers. Beside cultural particularities, learners often utter named entities like person names, organizations or locations which *betray* their actual native language. The features space usually grows to greatly outnumber the training/testing examples, making a feature selection process difficult, if not impossible.

We also investigate the importance of anaphoric shell nouns for the NLI task. Schmid [25] provides a list of shell nouns classified by six semantic classes: circumstantial, linguistic, modal, eventive, factual and mental. These features are quasi-topic-independent and we use them in combination with function words to increase the classification accuracy.

Positional frequencies [34] are obtained by counting the number of occurrences of tokens on the first, second and the last three positions. They can be an indicator that learners have certain ways of starting and ending a sentence (strategies of communication) which might be mother-tongue related.

5.2 Classifier

In our experiments, we use an L2-regularized L2-loss support vector classification machine with further parameter selection for C [8]. We adopt the log-entropy

weighting scheme to construct feature vectors from documents. This weighting method also increased the classification accuracies in previous studies [6, 13].

The log-entropy weighting is frequently encountered in latent semantic indexing [18], its purpose is to reduce the importance of high frequency features, and increase the weight for the ones that are good discriminants between documents. We compute the entropy for a feature i by the following formula:

$$g_i = 1 + \sum_{j=1}^{\mathcal{N}} \frac{p_{ij} \log 1 + p_{ij}}{\log \mathcal{N}} \tag{1}$$

where $\mathcal{N}$ is the number of documents in the corpus and p_{ij} is defined by the normalized frequency of term i in document j.

To normalize the p_{ij} values, we divide by the global frequency in the corpus:

$$\mathrm{gf}_i = \sum_{j=1}^{\mathcal{N}} \mathrm{tf}_{ij}$$

in consequence, the value of p_{ij} becomes: $p_{ij} = \frac{\mathrm{tf}_{ij}}{\mathrm{gf}_i}$.

The final weight of a feature is computed by multiplying the entropy with the log weight:

$$logent_{ij} = g_i \log(\mathrm{tf}_{ij} + 1) \tag{2}$$

6 Results and Interpretation

We have conducted multiple 10-fold cross-validation experiments corresponding to each combination of feature and corpus. The classifier was optimized with a search for the best parameter C which, in the majority of cases, attains low optimal values.

We have distinguished between topic sensitive/dependent features like character n-grams or positional token frequencies and topic independent features like function words, annotated errors or POS n-grams. (Table 2) contains the complete results for each set of experiments.

6.1 B13 Corpus for Native Language Identification

The B13 set of experiments represents the standard NLI task in which we evaluated a 13-class classifier to detect the native language/country of different English chunks.

On this sub-corpus, we obtained the best overall classification accuracy with character 4-grams (99.89%), closely followed by character trigrams.

Among the features that are topic dependent, positional token frequencies achieved a reasonable accuracy of 97.42% which indicates a correlation between the nativeness (mother tongues) of individuals and the way they start or end sentences, i.e. some *strategies of communication* may be determined by the native language.

Table 2. Average accuracy for each combination of feature and corpus. The highlighted values on each column represent the best scores for topic sensitive and independent features.

		Average accuracy for data set						
	Feature	**B13**	**LB_Ar**	**LB_Ch1**	**LB_Ch2**	**LB_Ge**	**LB_RuUk**	**LB_Sp**
independent	POS bigrams	75.43	67.04	88.10	70.18	76.58	90.22	67.25
	POS trigrams	87.20	75.07	92.35	83.01	**82.43**	**97.74**	72.25
	function words (FW)	86.06	**97.42**	99.5	94.71	60.48	83.45	**95.25**
	errors (E)	47.52	39.54	98.45	66.03	79.02	88.72	26.0
	FW and shell nouns	88.08	96.84	99.4	95.47	60.97	82.70	94.5
	FW + E + shell nouns	**93.75**	97.42	**99.85**	**96.22**	74.63	93.23	94.25
dependent	positional frequency	97.42	86.24	98.95	89.43	85.85	95.48	78.75
	char bigrams	94.47	93.98	98.75	88.3	83.41	93.98	86.25
	char trigrams	99.79	97.7	**99.9**	**97.35**	**90.24**	**97.74**	94.75
	char 4-grams	**99.89**	**99.14**	99.75	96.6	88.78	97.74	**96.25**

The best topic independent features were a combination of function words, errors and shell nouns which achieved an average cross-validation accuracy of (93.75%).

Table 3. Confusion matrix containing the rounded percentages of correctly classified B13 documents. A combination of function words, annotated errors and shell nouns were used as classification features.

	Brazil	Turkey	Italy	Mexico	PRC	France	Germany	S. Arabia	Colombia	Japan	Taiwan	Russia	S. Korea
Brazil	99	0	0	0	0	0	1	0	0	0	0	0	0
Turkey	1	96	0	2	1	0	0	0	0	0	0	0	0
Italy	0	0	83	1	0	7	1	1	3	2	2	0	1
Mexico	0	0	0	99	1	0	0	0	0	0	0	0	0
PRC	0	0	0	0	100	0	0	0	0	0	0	0	0
France	0	0	13	0	0	70	6	1	2	3	0	3	1
Germany	0	0	2	0	0	3	83	1	0	2	0	4	5
S. Arabia	1	0	0	0	0	0	0	96	0	0	1	3	0
Colombia	0	0	1	0	0	3	1	1	91	2	1	0	0
Japan	1	0	6	0	0	6	5	0	0	63	3	1	15
Taiwan	0	0	1	0	0	0	0	0	0	0	99	0	0
Russia	0	0	2	0	1	0	1	1	1	0	0	94	1
S. Korea	0	0	1	1	0	1	9	1	1	11	1	3	72

We regard the number of misclassified documents as a measure of similarity between two classes, (Table 3) contains the resulted confusion matrix for the B13 corpus. If native language relatedness can explain the 13% of the French documents misclassified as Italian, it cannot explain why native Mexican-Spanish and Colombian-Spanish do not trigger any confusion at all. Learners from distinct families of languages (Japanese and Korean, French and German) coming from related geographic areas/linguistic backgrounds evidence more similarities through the percentage of classification confusion. Similar confusions were also observed at the NLI 2013 shared task in which pairs of non-related languages - Japanese-Korean and Telugu-Hindi [13] exhibit confusion because learners belong to related linguistic or cultural backgrounds.

6.2 Linguistic Background Analysis

We have experimented with texts coming from learners that share the same mother tongue including variations or dialects (Russian and Ukrainian). The LB_Lang columns reflect the classification accuracy for these sub corpora.

For the different varieties of Arabic spoken in Egypt, Kuwait, United Arab Emirates (UAE) and Saudi Arabia, we show that the classifier is able to distinguish between the English texts of these learners. Function words achieved the best accuracy for topic independent features (97.42%) whereas errors or shell nouns did not improve the classification results. Among the topic dependent features, positional token frequencies obtained the lowest accuracy (lower than function words), hence, in (Table 4) we render the resulted confusion matrix.

Table 4. Confusion matrix containing the rounded percentages of correctly classified L_Ar documents using positional token frequencies.

	Egypt	Kuwait	UAE	S. Arabia
Egypt	91	2	6	1
Kuwait	5	82	11	2
UAE	9	6	77	8
S. Arabia	1	0	3	95

Table 5. Confusion matrix containing the rounded percentages of correctly classified L_Ch2 documents using function words, errors and shell nouns.

	HK	Taiwan	PRC
HK	90	6	4
Taiwan	1	99	0
PRC	0	0	100

The confusion matrix in (Table 4) shows that a significant amount of learner English from Egypt was misclassified as United Arab Emirates and vice-versa. Documents from Kuwait are also frequently confused as being from UAE (11%) in contrast, Saudi Arabian English can be differentiated from the remaining texts - 95% correctly classified. Positional token frequencies cover similar types of starting and ending a sentence, a good classification result could indicate that the differences do not necessarily emerge due to specific language variations getting transfered onto English, but rather because of different strategies of teaching/learning English in these countries.

The LB_Ch1 corpus contains one million tokens equally extracted from learners from People's Republic of China (PRC) and Taiwan. Standard Chinese (the Mandarin dialect) is spoken in both countries, with the mention that in People's Republic of China at least 13 more major dialects exists specific for different provinces. In (Table 2) we can observe that classifying English from Taiwan and English from PRC can be done with high accuracy values - using only function words, we get a 99.5% accuracy. Almost every type of feature (including errors) can act as excellent discriminants. For this particular instance we cannot be sure whether diverse dialects within PRC transfer to English yielding texts that are structurally different from the ones coming from Taiwan, or whether distinct learning methods are being used within the two countries.

The LB_Ch2 corpus is smaller including additional documents from English learners from Hong Kong. (Table 5) renders the confusion matrix which shows that only 10% of the learner English from Hong Kong is confused as being

from Taiwan or PRC. The overall accuracy for topic dependent features is of 97.35% with character trigrams while function words, errors and shell nouns used together obtain a 96.22% accuracy.

Table 6. Confusion matrix containing the rounded percentages of correctly classified LB_Ge documents using POS trigrams

	Switzerland	Austria	Germany
Switzerland	76	12	12
Austria	7	86	7
Germany	13	1	86

Table 7. First nine feature-selected character n-grams sorted by their corresponding F-score in the LB_Ge corpus

trigram	**F-score**	**examples**
"hi "	2.25	hi
"pu "	1.93	punctuation error
"pe "	0.54	type, hope
"oon"	0.31	soon, afternoon
"ope"	0.29	hope, open, opera
"tab"	0.25	table, vegetables,
"wn "	0.24	down, brown, town,
" af"	0.23	after, afternoon
"hit"	0.22	white

Lower accuracy values were obtained for countries in which German varieties are commonly spoken (Austria, Germany and Switzerland) - POS trigrams achieved 82.43% which is the best accuracy for the topic independent features. In (Table 6) we can observe the confusion matrix obtained with function words combined with errors: a significant number of documents are uniformly misclassified to each of the other countries.

Character trigrams - topic dependent features - attain the best overall accuracy value of 90.24%. Naturally, we are interested to observe which character n-grams increase the accuracy of the classifier. As a result, we investigate the n-grams with the highest F-score given the feature selection method proposed by Yi-Wei and Chih-Jen [35]. After extracting the most relevant trigrams for classification, we search for their occurrences in texts to find the most frequent examples. As (Table 7) indicates, the most discriminant trigrams cover topic independent features such as punctuation errors, function words ("soon", "after") and shell nouns ("type"). Yet, these features also cover content related words for example: "white", "brown", "afternoon", "opera", "table", "brown", etc. Under these circumstances, character n-grams do not necessarily reveal only interlanguage markers, but also hidden content that is not uniform for different groups of learners.

Even though Russian and Ukrainian are considered dialects or separate languages, we investigated whether the classifier can distinguish between English essays written by natives of these countries. The penultimate column in (Table 3) surprisingly indicates that both topic dependent and independent features achieve similar classification accuracies (97.74%). As in the case of PRC versus Taiwan, learners could also be influenced by different varieties of languages spoken across Russia, a fact which can determine separate linguistic backgrounds.

Last but not least, we carried a 7-class classification of texts coming from different regions of the Spanish-speaking world (LB_Sp): Spain, Mexico, Costa Rica, Peru, Colombia, Venezuela and Argentina. Learner English from these

countries can be classified with a 95.25% accuracy using only the list of function words while shell nouns or errors slightly decrease the value. (Table 8) contains the confusion matrix of the classification results using only the function words which obtained an overall accuracy of 95.25%. Character 4-grams can increase this accuracy with only 1%.

Table 8. Confusion matrix containing the rounded percentages of correctly classified LB_Sp documents using function words.

	Colombia	Mexico	Peru	Argentina	Venezuela	Costa Rica	Spain
Colombia	98	2	0	0	0	0	0
Mexico	0	100	0	0	0	0	0
Peru	0	0	100	0	0	0	0
Argentina	0	0	0	96	2	2	0
Venezuela	0	0	0	4	93	4	0
Costa Rica	0	0	0	9	11	79	2
Spain	0	0	0	0	0	0	100

English texts from Argentina, Colombia, Mexico, Peru and Spain can be distinguished almost perfectly from the rest while Costa Rica and Venezuela share the largest amounts of classification confusion using function words (topic independent features).

These results indicate that students from each country go through similar stages of learning English and possibly any foreign language. For example, the learners from Mexico may be influenced by linguistic and political factors (USA being a neighboring country) so that they achieve good proficiency levels at earlier stages of learning English, compared to students from other countries which experience less interaction with native English communities. Our investigation does not account the different grades students had for the Cambridge examination which, we assume, might also be a factor of influence. Furthermore, the distributions of different levels across the corpus can also be a factor of influence and more work is prepared in this direction.

7 Conclusions

In this paper we provide an analysis of the linguistic features that are suitable for the task of native language identification. We research our claims on a subset of the EFCAMDAT corpus [10] from which named entities and references to language names were removed. In addition to the standard classification features used in literature such as character n-grams, part of speech n-grams, function words or annotated errors, we further prove that anaphoric shell nouns and positional token frequencies represent interlanguage markers that contribute to the overall classification accuracies. Our results also suggest that topic sensitive features tend to obtain the best results across different corpora. However, we recommend additional care when employing these features since texts may contain hidden topics that can determine misleading classifications.

Our data includes error annotated documents from different countries in which the same language is spoken by a majority. Apart from this, the corpus is compiled from medium-low proficiency English texts that exhibit a significant amount of errors and interlanguage features, therefore, facilitating the classification tasks.

The novelty of our study does not only rely on the experimental analysis of interlanguage features but also on the investigation of the inner dissimilarities within a group of learners that share the same mother tongue. To explain the differences that appear between learners with distinct native countries and similar native languages, we conjecture the existence of a linguistic background which can be determined by the previous languages learned and possibly cultural and political factors. The linguistic background interacts with the process of learning, complementary to the learner's native language.

On one hand, language relatedness can explain the classification confusions that emerge between similar languages e.g. French and Italian. On the other hand, this phenomenon cannot explain why Spanish from Mexico and Spanish from Colombia do not trigger confusion or why learners from distinct families of languages (Japanese and Korean, French and German, Telugu and Hindi [13]) coming from neighboring geographic areas evidence more similarities through the percentage of misclassified documents.

We are inclined to believe these similarities fade as the learner proficiency increases, but the corpus required to investigate this hypothesis is not available yet and its development is part of our current and future work. Our results trace the existence of a linguistic background. Nevertheless, a more thorough investigation would be necessary to fully analyze and understand the roots of this phenomenon.

Acknowledgments. I would like to address special thanks to Anca Bucur for her helpful suggestions and support in improving this paper. Needless to say, any remaining errors are mine alone.

References

[1] Brooke, J., Hirst, G.: Native language detection with 'cheap' learner corpora. In: Conference of Learner Corpus Research (LCR 2011). Presses universitaires de Louvain, Louvain-la-Neuve (2011)

[2] Chomsky, N.A.: Linguistics and philosophy. In: Hook, S. (ed.) Language and Philosophy. New York University Press (1969)

[3] Corder, S.P.: Language distance and the magnitude of the language learning task. Studies in Second Language Acquisition 2, 27–36 (1979)

[4] Dinu, A.: On classifying coherent/incoherent romanian short texts. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008). European Language Resources Association (ELRA), Marrakech (2008)

[5] Dinu, L.P., Niculae, V., Şulea, O.M.: Pastiche detection based on stopword rankings: exposing impersonators of a romanian writer. In: Proceedings of the Workshop on Computational Approaches to Deception Detection, EACL 2012, pp. 72–77. Association for Computational Linguistics, Stroudsburg (2012)
[6] Dumais, S.: Improving the retrieval of information from external sources. Behavior Research Methods, Instruments and Computers 23(2), 229–236 (1991)
[7] Englishtown: Education first (2012), http://www.englishtown.com/
[8] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
[9] Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Knight, K., Ng, H.T., Oflazer, K. (eds.) ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, University of Michigan, USA, June 25-30. The Association for Computer Linguistics (2005)
[10] Geertzen, J., Alexopoulou, T., Korhonen, A.: Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In: Proceedings of the 31st Second Language Research Forum (SLRF). Cascadilla Press, MA (2013)
[11] Granger, S., Dagneaux, E., Meunier, F.: The International Corpus of Learner English: Handbook and CD-ROM, version 2. Presses Universitaires de Louvain, Louvain-la-Neuve (2009)
[12] Ionescu, T.R., Popescu, M., Cahill, A.: Can characters reveal your native language? a language-independent approach to native language identification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1363–1373. Association for Computational Linguistics (2014)
[13] Jarvis, S., Bestgen, Y., Pepper, S.: Maximizing classification accuracy in native language identification. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 111–118. Association for Computational Linguistics, Atlanta (2013)
[14] Kolhatkar, V., Zinsmeister, H., Hirst, G.: Interpreting anaphoric shell nouns using antecedents of cataphoric shell nouns as training data. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 300–310. Association for Computational Linguistics (2013)
[15] Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. J. Am. Soc. Inf. Sci. Technol. 60(1), 9–26 (2009)
[16] Koppel, M., Schler, J., Zigdon, K.: Automatically determining an anonymous author's native language. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 209–217. Springer, Heidelberg (2005)
[17] Koppel, M., Schler, J., Zigdon, K.: Determining an author's native language by mining a text for errors. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 624–628. ACM, Chicago (2005)
[18] Landauer, T., McNamara, D., Dennis, S., Kintsch, W.: Handbook of Latent Semantic Analysis. Taylor and Francis (2013)
[19] Lim, C., Lee, K., Kim, G.: Multiple sets of features for automatic genre classification of web documents. Information Processing and Management 41(5), 1263–1276 (2005)
[20] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. J. Mach. Learn. Res. 2, 419–444 (2002)

[21] Long, M.H.: Maturational constraints on language development. Studies in Second Language Acquisition 12, 251–285 (1990)
[22] Münte, T.F., Wieringa, B.M., Weyerts, H., Szentkuti, A., Matzke, M., Johannes, S.: Differences in brain potentials to open and closed class words: class and frequency effects. Neuropsychologia 39(1), 91–102 (2001)
[23] Nagata, R., Whittaker, E.W.D.: Reconstructing an indo-european family tree from non-native english texts. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Volume 1: Long Papers, Sofia, Bulgaria, August 4-9, pp. 1137–1147 (2013)
[24] Sang, E.F.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, pp. 142–147 (2003)
[25] Schmid, H.U.: English Abstract Nouns As Conceptual Shells: From Corpus to Cognition. In: Topics in English Linguistics 34. De Gruyter Mouton, Berlin (2000)
[26] Selinker, L., Rutherford, W.: Rediscovering Interlanguage. Applied Linguistics and Language Study. Routledge (2014)
[27] Selinker, L.: Interlanguage. International Review of Applied Linguistics in Language Teaching 10(1-4), 209–232 (1972)
[28] Tarone, E.: Interlanguage. Blackwell Publishing Ltd. (2012)
[29] Tetreault, J., Blanchard, D., Cahill, A.: A report on the first native language identification shared task. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications. Association for Computational Linguistics, Atlanta (2013)
[30] Tetreault, J., Blanchard, D., Cahill, A., Chodorow, M.: Native tongues, lost and found: Resources and empirical evaluations in native language identification. In: Proceedings of COLING 2012, pp. 2585–2602. The COLING 2012 Organizing Committee, Mumbai (2012)
[31] Tsur, O., Rappoport, A.: Using Classifier Features for Studying the Effect of Native Language on the Choice of Written Second Language Words. In: Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition, pp. 9–16. Association for Computational Linguistics, Prague (2007)
[32] Tsvetkov, Y., Twitto, N., Schneider, N., Ordan, N., Faruqui, M., Chahuneau, V., Wintner, S., Dyer, C.: Identifying the l1 of non-native writers: the cmu-haifa system. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 279–287. Association for Computational Linguistics, Atlanta (2013)
[33] Valette, R.M.: Proficiency and the prevention of fossilization an editorial. The Modern Language Journal 75(3), 325–328 (1991)
[34] Volansky, V., Ordan, N., Wintner, S.: On the features of translationese. Literary and Linguistic Computing (2013)
[35] Yi-Wei, C., Chih-Jen, L.: Combining svms with various feature selection strategies. Feature Extraction 207, 315–324 (2006)

Author Index