

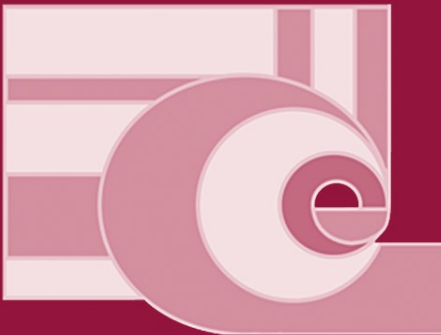
Alexander Gelbukh (Ed.)

LNCS 7181

Computational Linguistics and Intelligent Text Processing

13th International Conference, CICLing 2012
New Delhi, India, March 2012
Proceedings, Part I

1
Part I



 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

13th International Conference, CICLing 2012
New Delhi, India, March 11-17, 2012
Proceedings, Part I

Volume Editor

Alexander Gelbukh
Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Col. Nueva Industrial Vallejo, CP 07738, Mexico D.F., Mexico
E-mail: gelbukh@gelbukh.com

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-28603-2 e-ISBN 978-3-642-28604-9
DOI 10.1007/978-3-642-28604-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012932159

CR Subject Classification (1998): H.3, H.4, F.1, I.2, H.5, H.2.8, I.5

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

CICLing 2012 was the 13th 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 volume numbers 2004, 2276, 2588, 2945, 3406, 3878, 4394, 4919, 5449, 6008, 6608, and 6609.

The set has been structured into 13 sections:

- NLP System Architecture
- Lexical Resources
- Morphology and Syntax
- Word Sense Disambiguation and Named Entity Recognition
- Semantics and Discourse
- Sentiment Analysis, Opinion Mining, and Emotions
- Natural Language Generation
- Machine Translation and Multilingualism
- Text Categorization and Clustering
- Information Extraction and Text Mining
- Information Retrieval and Question Answering
- Document Summarization
- Applications

The 2012 event received a record high number of submissions. A total of 307 papers by 575 authors from 46 countries were submitted for evaluation by the International Program Committee, see Tables 1 and 2. This two-volume set contains revised versions of 88 papers selected for presentation; thus the acceptance rate for this set was 28.6%.

The book features invited papers by

- Srinivas Bangalore, AT&T, USA
- John Carroll, University of Sussex, UK
- Marie-Francine Moens, Katholieke Universiteit Leuven, Belgium
- Salim Roukos, IBM, USA

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

Table 1. Statistics of submissions and accepted papers by country or region

Country or region	Authors		Papers ¹	Country or region	Authors		Papers ¹
	Subm.	Subm.	Accp.		Subm.	Subm.	Accp.
Argentina	1	0.5	–	Japan	25	11.5	3.5
Australia	3	1	1	Kazakhstan	10	6	–
Belgium	2	1	1	Korea, Republic of	10	5.25	2
Brazil	3	2	1	Lebanon	3	2	1
Canada	3	2.5	–	Macao	4	2	–
Chile	3	1	1	Mexico	14	7.41	1.2
China	29	12.5	5.5	Norway	1	0.5	–
Colombia	4	3	–	Poland	10	7	2
Croatia	2	1	1	Portugal	6	2	–
Cuba	1	0.33	0.33	Romania	11	10	2
Czech Republic	5	3	2	Russian Federation	9	5	–
Denmark	1	1	–	Saudi Arabia	4	2	–
Finland	7	3	2	Spain	36	11.85	8.57
France	30	12.9	7.4	Sri Lanka	4	1	1
Germany	20	8.83	4.33	Sweden	12	5	2
Greece	5	2	–	Switzerland	1	1	–
Hong Kong	1	1	1	Taiwan	2	2	–
Hungary	2	1	1	Turkey	3	1.5	1
India	196	120	18.75	United Arab Emirates	5	2	1
Indonesia	7	3	–	UK	14	4.92	2.67
Iran	11	15	2	USA	33	13.75	7.5
Ireland	2	1	1	Uruguay	5	1	1
Italy	11	4.25	2.25	Viet Nam	4	1.5	–
				<i>Total:</i>	575	307	89

¹ 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.

With this event we continued with our policy of giving preference to papers with verifiable and reproducible results: we encouraged the authors to provide, in electronic form, a proof of their claims or a working description of the suggested algorithm, in addition to the verbal description given in the paper. If the paper claimed experimental results, we encouraged the authors to make available to the community all the input data necessary to verify and reproduce these results; if it claimed to advance human knowledge by introducing an algorithm, we encouraged the authors to make the algorithm itself, in some programming language, available to the public. This additional electronic material will be permanently stored on 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 greater in size than the claim itself); each theorem – and not just its descrip-

Table 2. Statistics of submissions and accepted papers by topic²

Accepted	Submitted	% accepted	Topic
20	44	45	Text mining
18	61	30	Information extraction
18	45	40	Semantics and discourse
18	44	41	Lexical resources
16	63	25	Information retrieval
13	40	33	Practical applications
13	29	45	Opinion mining
11	35	31	Clustering and categorization
11	21	52	Acquisition of lexical resources
8	19	42	Syntax and chunking (linguistics)
8	17	47	Word sense disambiguation
8	14	57	Summarization
7	21	33	Formalisms and knowledge representation
7	16	44	Symbolic and linguistic methods
6	50	12	Other
6	23	26	Statistical methods (mathematics)
5	23	22	Morphology
5	18	28	Named entity recognition
5	15	33	POS tagging
4	30	13	Machine translation and multilingualism
4	17	24	Question answering
4	12	33	Noisy text processing and cleaning
4	5	80	Textual entailment
3	12	25	Text generation
3	10	30	Cross-language information retrieval
3	8	38	Spelling and grammar checking
2	13	15	Natural language interfaces
2	7	29	Emotions and humor
2	6	33	Parsing algorithms (mathematics)
1	9	11	Anaphora resolution
1	6	17	Computational terminology
–	4	0	Speech processing

² As indicated by the authors. A paper may belong to several topics.

tion or general idea – is completely and precisely presented to the reader. Electronic media allow computational linguists to provide material analogous to the proofs and formulas in mathematics in full length – which can amount to megabytes or gigabytes of data – separately from a 12-page description published in the book. A more detailed argumentation for this new policy can be found on www.CICLing.org/why_verify.htm.

To encourage the provision of algorithms and data along with the published papers, we selected the winner 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 documenta-

tion, 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 – those parameters were evaluated for the paper itself and not for the data.

The following papers received the Best Paper Awards, the Best Student Paper Award, as well as the Verifiability, Reproducibility, and Working Description Award, correspondingly (the best student paper was selected from papers of which the first author was a full-time student, excluding the papers that received a Best Paper Award):

- 1st Place: *Automated Detection of Local Coherence in Short Argumentative Essays Based on Centering Theory*, by Vasile Rus and Nopal Nirauala, USA;
- 2nd Place: *Corpus-Driven Hyponym Acquisition for Turkish Language*, by Savaş Yıldırım and Tuğba Yıldız, Turkey;
- 3rd Place: *Towards Automatic Generation of Catchphrases for Legal Case Reports*, by Filippo Galgani, Paul Compton, and Achim Hoffmann, Australia;
- Student: *Predictive Text Entry for Agglutinative Languages Using Unsupervised Morphological Segmentation*, by Miikka Silfverberg, Krister Lindén, and Mirka Hyvärinen, Finland;
- Verifiability: *Extraction of Relevant Figures and Tables for Multi-document Summarization*, by Ashish Sadh, Amit Sahu, Devesh Srivastava, Ratna Sanyal, and Sudip Sanyal, India.

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 their high scientific level, one of the success factors of the CICLing conferences is their excellent cultural program. The attendees of the conference had a chance to visit the main tourist attractions of the marvellous, mysterious, colorful, and infinitely diverse India: Agra with the famous Taj Mahal, Jaipur, and Delhi. They even enjoyed riding elephants!

I would like to thank all those involved in the organization of this conference. Most importantly 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, Software Reviewing Committee, Award Selection Committee, as well as additional reviewers, for their hard and very professional work. Special thanks go to Rada Mihalcea, Ted Pedersen, and Grigori Sidorov, for their invaluable support in the reviewing process.

I would like to cordially thank the Indian Institute of Technology Delhi, for hosting the conference. With deep gratitude I acknowledge the support of Prof. B.S. Panda, the Head of Department of Mathematics, IIT Delhi. My most special thanks go to Prof. Niladri Chatterjee for his great enthusiasm and hard work

on the organization of the conference, as well as to the members of the local Organizing Committee for their enthusiastic and hard work, which has led to the success of the conference.

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 record short time – it is always a great pleasure to work with Springer.

February 2012

Alexander Gelbukh

Organization

CICLing 2012 was hosted by the Indian Institute of Technology Delhi and organized by the CICLing 2012 Organizing Committee, in conjunction with the Natural Language and Text Processing Laboratory of the CIC (Center for Computing Research) of the IPN (National Polytechnic Institute), Mexico.

Organizing Chair

Niladri Chatterjee

Organizing Committee

Niladri Chatterjee (Chair)
Pushpak Bhattacharyya
Santanu Chaudhury
K.K. Biswas
Alexander Gelbukh
Arsh Sood

Ankit Prasad
Avikant Bhardwaj
Mehak Gupta
Pramod Kumar Sahoo
Renu Balyan
Susmita Chakraborty

Program Chair

Alexander Gelbukh

Program Committee

Sophia Ananiadou
Bogdan Babych
Ricardo Baeza-Yates
Sivaji Bandyopadhyay
Srinivas Bangalore
Roberto Basili
Anja Belz
Pushpak Bhattacharyya
António Branco
Nicoletta Calzolari
Sandra Carberry
Dan Cristea
Walter Daelemans
Alex Chengyu Fang
Anna Feldman
Alexander Gelbukh

Gregory Grefenstette
Eva Hajicova
Yasunari Harada
Koiti Hasida
Graeme Hirst
Aleš Horák
Nancy Ide
Diana Inkpen
Hitoshi Isahara
Aravind Joshi
Sylvain Kahane
Alma Kharrat
Philipp Koehn
Leila Kosseim
Krister Lindén
Aurelio Lopez

Cerstin Mahlow
Sun Maosong
Yuji Matsumoto
Diana McCarthy
Helen Meng
Rada Mihalcea
Ruslan Mitkov
Dunja Mladenic
Marie-Francine Moens
Masaki Murata
Vivi Nastase
Roberto Navigli
Kjetil Nørvåg
Constantin Orăsan
Patrick Saint-Dizier
Maria Teresa Paziienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Stelios Piperidis
Irina Prodanof
Arne Ranta
Victor Raskin
Fuji Ren

German Rigau
Fabio Rinaldi
Horacio Rodriguez
Vasile Rus
Horacio Saggon
Kepa Sarasola
Serge Sharoff
Grigori Sidorov
Thamar Solorio
John Sowa
Ralf Steinberger
Vera Lúcia Strube De Lima
Tomek Strzalkowski
Jun Suzuki
Christoph Tillmann
George Tsatsaronis
Junichi Tsujii
Dan Tufiş
Hans Uszkoreit
Felisa Verdejo
Manuel Vilares Ferro
Haifeng Wang
Bonnie Webber

Software Reviewing Committee

Ted Pedersen
Florian Holz
Miloš Jakubčíek

Sergio Jiménez Vargas
Miikka Silfverberg
Ronald Winnemöller

Award Committee

Alexander Gelbukh
Eduard Hovy
Rada Mihalcea

Ted Pedersen
Yorick Wiks

Additional Referees

Adrián Blanco González
Ahmad Emami
Akinori Fujino
Alexandra Balahur
Alvaro Rodrigo
Amitava Das

Ana Garcia-Serrano
Ananthakrishnan Ramanathan
Andrej Gardon
Aniruddha Ghosh
Antoni Oliver
Anup Kumar Kolya

Arantza Casillas-Rubio
 Arkaitz Zubiaga
 Bing Xiang
 Binod Gyawali
 Blaz Novak
 Charlie Greenbacker
 Clarissa Xavier
 Colette Joubarne
 Csaba Bodor
 Daisuke Bekki
 Daniel Eisinger
 Danilo Croce
 David Vilar
 Delia Rusu
 Diana Trandabat
 Diman Ghazi
 Dipankar Das
 Egoitz Laparra
 Ekaterina Ovchinnikova
 Enrique Amigó
 Eugen Ignat
 Fazel Keshtkar
 Feiyu Xu
 Francisco Jose Ribadas Pena
 Frederik Vaassen
 Gabriela Ferraro
 Gabriela Ramirez De La Rosa
 Gerold Schneider
 Gorka Labaka
 Guenter Neumann
 Guillermo Garrido
 H M Ishrar Hussain
 Håkan Burden
 Hendra Setiawan
 Hiroya Takamura
 Hiroyuki Shindo
 Ingo Glöckner
 Ionut Cristian Pistol
 Irina Chugur
 Irina Temnikova
 Jae-Woong Choe
 Janez Brank
 Jirka Hana
 Jirka Hana
 Jordi Atserias
 Julian Brooke
 K.V.S. Prasad
 Katsuhito Sudoh
 Kishiko Ueno
 Kostas Stefanidis
 Kow Kuroda
 Krasimir Angelov
 Laritza Hernández
 Le An Ha
 Liliana Barrio-Alvers
 Lorand Dali
 Luis Otávio De Colla Furquim
 Luz Rello
 Maite Oronoz Anchordoqui
 Maria Kissa
 Mario Karlovcec
 Martin Scaiano
 Masaaki Nagata
 Matthias Reimann
 Maud Ehrmann
 Maya Carrillo
 Michael Piotrowski
 Miguel Angel Rios Gaona
 Miguel Ballesteros
 Mihai Alex Moruz
 Milagros Fernández Gavilanes
 Milos Jakubicek
 Miranda Chong
 Mitja Trampus
 Monica Macoveiciuc
 Najeh Hajlaoui
 Natalia Konstantinova
 Nathan Michalov
 Nattiya Kanhabua
 Nenad Tomasev
 Niyu Ge
 Noushin Rezapour Asheghi
 Oana Frunza
 Oier Lopez De Lacalle
 Olga Kolesnikova
 Omar Alonso
 Paolo Annesi
 Peter Ljunglöf
 Pinaki Bhaskar
 Prokopis Prokopidis

Rainer Winnenburg
Ramona Enache
Raquel Martínez
Richard Forsyth
Robin Cooper
Rodrigo Aggeri
Roser Morante
Ryo Otoguro
Samira Shaikh
Santanu Pal
Shamima Mithun
Sharon Small
Simon Mille
Simone Paolo Ponzetto
Siva Reddy
Somnath Banerjee
Tadej Štajner
Thierry Declerck

Ting Liu
Tom De Smedt
Tommaso Caselli
Tong Wang
Toshiyuki Kanamaru
Tsutomu Hirao
Ulf Hermjakob
Upendra Sapkota
Vanessa Murdock
Victor Darriba
V́ctor Peinado
Vít Baisa
Vojtech Kovar
Wilker Aziz
Yulia Ledeneva
Yvonne Skalban
Zuzana Neverilova

Website and Contact

The webpage of the CICLing conference series is www.CICLing.org. It contains information about past CICLing conferences and their satellite events, including published papers or their abstracts, photos, video recordings of keynote talks, as well as information about the forthcoming CICLing conferences and contact options.

Table of Contents – Part I

NLP System Architecture

Thinking Outside the Box for Natural Language Processing (Invited Paper)	1
<i>Srinivas Bangalore</i>	

Lexical Resources

A Graph-Based Method to Improve WordNet Domains	17
<i>Aitor González, German Rigau, and Mauro Castillo</i>	
Corpus-Driven Hyponym Acquisition for Turkish Language (Best Paper Award, Second Place)	29
<i>Savaş Yıldırım and Tuğba Yıldız</i>	
Automatic Taxonomy Extraction in Different Languages Using Wikipedia and Minimal Language-Specific Information	42
<i>Renato Domínguez García, Sebastian Schmidt, Christoph Rensing, and Ralf Steinmetz</i>	
Ontology-Driven Construction of Domain Corpus with Frame Semantics Annotations	54
<i>He Tan, Rajaram Kaliyaperumal, and Nirupama Benis</i>	
Building a Hierarchical Annotated Corpus of Urdu: The URDU.KON- TB Treebank	66
<i>Qaiser Abbas</i>	

Morphology and Syntax

A Morphological Analyzer Using Hash Tables in Main Memory (MAHT) and a Lexical Knowledge Base	80
<i>Francisco J. Carreras-Riudavets, Juan C. Rodríguez-del-Pino, Zenón Hernández-Figueroa, and Gustavo Rodríguez-Rodríguez</i>	
Optimal Stem Identification in Presence of Suffix List	92
<i>Vasudevan N. and Pushpak Bhattacharyya</i>	
On the Adequacy of Three POS Taggers and a Dependency Parser	104
<i>Ramadan Alfared and Denis Béchet</i>	

Will the Identification of Reduplicated Multiword Expression (RMWE) Improve the Performance of SVM Based Manipuri POS Tagging?	117
<i>Kishorjit Nongmeikapam, Aribam Umananda Sharma, Laishram Martina Devi, Napoleon Keisam, Khangengbam Dilip Singh, and Sivaji Bandyopadhyay</i>	
On Formalization of Word Order Properties	130
<i>Vladislav Kuboň, Markéta Lopatková, and Martin Plátek</i>	
Core-Periphery Organization of Graphemes in Written Sequences: Decreasing Positional Rigidity with Increasing Core Order	142
<i>Md. Izhar Ashraf and Sitabhra Sinha</i>	
Discovering Linguistic Patterns Using Sequence Mining	154
<i>Nicolas Béchet, Peggy Cellier, Thierry Charnois, and Bruno Crémilleux</i>	
What about Sequential Data Mining Techniques to Identify Linguistic Patterns for Stylistics?	166
<i>Solen Quiniou, Peggy Cellier, Thierry Charnois, and Dominique Legallois</i>	
Resolving Syntactic Ambiguities in Natural Language Specification of Constraints	178
<i>Imran Sarwar Bajwa, Mark Lee, and Behzad Bordbar</i>	
A Computational Grammar of Sinhala	188
<i>Chamila Liyanage, Randil Pushpananda, Dulip Lakmal Herath, and Ruwan Weerasinghe</i>	
Automatic Identification of Persian Light Verb Constructions	201
<i>Bahar Salehi, Narjes Askarian, and Afsaneh Fazly</i>	
Word Sense Disambiguation and Named Entity Recognition	
A Cognitive Approach to Word Sense Disambiguation	211
<i>Sudakshina Dutta and Anupam Basu</i>	
A Graph-Based Approach to WSD Using Relevant Semantic Trees and N-Cliques Model	225
<i>Yoan Gutiérrez, Sonia Vázquez, and Andrés Montoyo</i>	
Using Wiktionary to Improve Lexical Disambiguation in Multiple Languages	238
<i>Kiem-Hieu Nguyen and Cheol-Young Ock</i>	

Two Stages Based Organization Name Disambiguity	249
<i>Shu Zhang, Jianwei Wu, Dequan Zheng, Yao Meng, Yingju Xia, and Hao Yu</i>	
Optimizing CRF-Based Model for Proper Name Recognition in Polish Texts	258
<i>Michał Marcińczuk and Maciej Janicki</i>	
Methods of Estimating the Number of Clusters for Person Cross Document Coreference Task	270
<i>Octavian Popescu and Roberto Zanolì</i>	
Coreference Resolution Using Tree CRFs	285
<i>Vijay Sundar Ram R. and Sobha Lalitha Devi</i>	
Arabic Entity Graph Extraction Using Morphology, Finite State Machines, and Graph Transformations	297
<i>Jad Makhlouta, Fadi Zaraket, and Hamza Harkous</i>	
Integrating Rule-Based System with Classification for Arabic Named Entity Recognition	311
<i>Sherief Abdallah, Khaled Shaalan, and Muhammad Shoaib</i>	
Semantics and Discourse	
Space Projections as Distributional Models for Semantic Composition	323
<i>Paolo Annesi, Valerio Storch, and Roberto Basili</i>	
Distributional Models and Lexical Semantics in Convolution Kernels . . .	336
<i>Daniò Croce, Simone Filice, and Roberto Basili</i>	
Multiple Level of Referents in Information State	349
<i>Gábor Alberti and Márton Károly</i>	
Inferring the Scope of Negation in Biomedical Documents	363
<i>Miguel Ballesteros, Virginia Francisco, Alberto Díaz, Jesús Herrera, and Pablo Gervás</i>	
LDA-Frames: An Unsupervised Approach to Generating Semantic Frames	376
<i>Jiří Materna</i>	
Unsupervised Acquisition of Axioms to Paraphrase Noun Compounds and Genitives	388
<i>Anselmo Peñas and Ekaterina Ovchinnikova</i>	
Age-Related Temporal Phrases in Spanish and Italian	402
<i>Sofía N. Galicia-Haro and Alexander Gelbukh</i>	

Can Modern Statistical Parsers Lead to Better Natural Language Understanding for Education?	415
<i>Umair Z. Ahmed, Arpit Kumar, Monojit Choudhury, and Kalika Bali</i>	
Exploring Classification Concept Drift on a Large News Text Corpus . . .	428
<i>Artur Šilić and Bojana Dalbelo Bašić</i>	
An Empirical Study of Recognizing Textual Entailment in Japanese Text	438
<i>Quang Nhat Minh Pham, Le Minh Nguyen, and Akira Shimazu</i>	
Automated Detection of Local Coherence in Short Argumentative Essays Based on Centering Theory (Best Paper Award, First Place) . . .	450
<i>Vasile Rus and Nobal Niraula</i>	
A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra-sentence Discourse Segments in Spanish	462
<i>Iria da Cunha, Eric SanJuan, Juan-Manuel Torres-Moreno, M. Teresa Cabré, and Gerardo Sierra</i>	
Sentiment Analysis, Opinion Mining, and Emotions	
Feature Specific Sentiment Analysis for Product Reviews	475
<i>Subhabrata Mukherjee and Pushpak Bhattacharyya</i>	
Biographies or Blenders: Which Resource Is Best for Cross-Domain Sentiment Analysis?	488
<i>Natalia Ponomareva and Mike Thelwall</i>	
A Generate-and-Test Method of Detecting Negative-Sentiment Sentences	500
<i>Yoonjung Choi, Hyo-Jung Oh, and Sung-Hyon Myaeng</i>	
Roles of Event Actors and Sentiment Holders in Identifying Event-Sentiment Association	513
<i>Anup Kumar Kolya, Dipankar Das, Asif Ekbal, and Sivaji Bandyopadhyay</i>	
Applying Sentiment and Social Network Analysis in User Modeling	526
<i>Mohammadreza Shams, Mohammadtaghi Saffar, Azadeh Shakery, and Hesham Faili</i>	
The 5W Structure for Sentiment Summarization-Visualization-Tracking	540
<i>Amitava Das, Sivaji Bandyopadhyay, and Björn Gambäck</i>	

The Naive Bayes Classifier in Opinion Mining: In Search of the Best Feature Set	556
<i>Liviu P. Dinu and Iulia Iuga</i>	
A Domain Independent Framework to Extract and Aggregate Analogous Features in Online Reviews	568
<i>Archana Bhattarai, Nobal Niraula, Vasile Rus, and King-Ip Lin</i>	
Learning Lexical Subjectivity Strength for Chinese Opinionated Sentence Identification	580
<i>Xin Wang and Guohong Fu</i>	
Building Subjectivity Lexicon(s) from Scratch for Essay Data.....	591
<i>Beata Beigman Klebanov, Jill Burstein, Nitin Madnani, Adam Faulkner, and Joel Tetreault</i>	
Emotion Ontology Construction from Chinese Knowledge	603
<i>Peilin Jiang, Fei Wang, Fuji Ren, and Nanning Zheng</i>	
Author Index	615

Table of Contents – Part II

Natural Language Generation

Exploring Extensive Linguistic Feature Sets in Near-Synonym Lexical Choice	1
<i>Mari-Sanna Paukkeri, Jaakko Väyrynen, and Antti Arppe</i>	
Abduction in Games for a Flexible Approach to Discourse Planning	13
<i>Ralf Klabunde, Sebastian Reuße, and Björn Schlünder</i>	

Machine Translation and Multilingualism

Document-Specific Statistical Machine Translation for Improving Human Translation Productivity (Invited Paper)	25
<i>Salim Roukos, Abraham Ittycheriah, and Jian-Ming Xu</i>	
Minimum Bayes Risk Decoding with Enlarged Hypothesis Space in System Combination	40
<i>Tsuyoshi Okita and Josef van Genabith</i>	
Phrasal Syntactic Category Sequence Model for Phrase-Based MT	52
<i>Hailong Cao, Eiichiro Sumita, Tiejun Zhao, and Sheng Li</i>	
Integration of a Noun Compound Translator Tool with Moses for English-Hindi Machine Translation and Evaluation	60
<i>Prashant Mathur and Soma Paul</i>	
Neoclassical Compound Alignments from Comparable Corpora	72
<i>Rima Harastani, Béatrice Daille, and Emmanuel Morin</i>	
QAlign: A New Method for Bilingual Lexicon Extraction from Comparable Corpora	83
<i>Amir Hazem and Emmanuel Morin</i>	
Aligning the Un-Alignable — A Pilot Study Using a Noisy Corpus of Nonstandardized, Semi-parallel Texts	97
<i>Florian Petran</i>	
Parallel Corpora for WordNet Construction: Machine Translation vs. Automatic Sense Tagging	110
<i>Antoni Oliver and Salvador Climent</i>	
Method to Build a Bilingual Lexicon for Speech-to-Speech Translation Systems	122
<i>Keiji Yasuda, Andrew Finch, and Eiichiro Sumita</i>	

Text Categorization and Clustering

A Fast Subspace Text Categorization Method Using Parallel Classifiers	132
<i>Nandita Tripathi, Michael Oakes, and Stefan Wermter</i>	
Research on Text Categorization Based on a Weakly-Supervised Transfer Learning Method	144
<i>Dequan Zheng, Chenghe Zhang, Geli Fei, and Tiejun Zhao</i>	
Fuzzy Combinations of Criteria: An Application to Web Page Representation for Clustering	157
<i>Alberto Pérez García-Plaza, Víctor Fresno, and Raquel Martínez</i>	
Clustering Short Text and Its Evaluation	169
<i>Prajol Shrestha, Christine Jacquín, and Béatrice Daille</i>	

Information Extraction and Text Mining

Information Extraction from Webpages Based on DOM Distances	181
<i>Carlos Castillo, Héctor Valero, José Guadalupe Ramos, and Josep Silva</i>	
Combining Flat and Structured Approaches for Temporal Slot Filling or: How Much to Compress?	194
<i>Qi Li, Javier Artiles, Taylor Cassidy, and Heng Ji</i>	
Event Annotation Schemes and Event Recognition in Spanish Texts	206
<i>Dina Wonsever, Aiala Rosá, Marisa Malcuori, Guillermo Moncecchi, and Alan Descoins</i>	
Automatically Generated Noun Lexicons for Event Extraction	219
<i>Béatrice Arnulphy, Xavier Tannier, and Anne Vilnat</i>	
Lexical Acquisition for Clinical Text Mining Using Distributional Similarity (Invited Paper)	232
<i>John Carroll, Rob Koeling, and Shivani Puri</i>	
Developing an Algorithm for Mining Semantics in Texts	247
<i>Minhua Huang and Robert M. Haralick</i>	
Mining Market Trend from Blog Titles Based on Lexical Semantic Similarity	261
<i>Fei Wang and Yunfang Wu</i>	

Information Retrieval and Question Answering

Ensemble Approach for Cross Language Information Retrieval	274
<i>Dinesh Mavaluru, R. Shriram, and W. Aisha Banu</i>	

Web Image Annotation Using an Effective Term Weighting	286
<i>Vundavalli Srinivasarao and Vasudeva Varma</i>	
Metaphone-pt_BR: The Phonetic Importance on Search and Correction of Textual Information	297
<i>Carlos C. Jordão and João Luís G. Rosa</i>	
Robust and Fast Two-Pass Search Method for Lyric Search Covering Erroneous Queries Due to Mishearing	306
<i>Xin Xu and Tsuneo Kato</i>	
Bootstrap-Based Equivalent Pattern Learning for Collaborative Question Answering	318
<i>Tianyong Hao and Eugene Agichtein</i>	
How to Answer Yes/No Spatial Questions Using Qualitative Reasoning?	330
<i>Marcin Walas</i>	
Question Answering and Multi-search Engines in Geo-Temporal Information Retrieval	342
<i>Fernando S. Peregrino, David Tomás, and Fernando Llopis Pascual</i>	
Document Summarization	
Using Graph Based Mapping of Co-occurring Words and Closeness Centrality Score for Summarization Evaluation	353
<i>Niraj Kumar, Kannan Srinathan, and Vasudeva Varma</i>	
Combining Syntax and Semantics for Automatic Extractive Single-Document Summarization	366
<i>Araly Barrera and Rakesh Verma</i>	
Combining Summaries Using Unsupervised Rank Aggregation	378
<i>Girish Keshav Palshikar, Shailesh Deshpande, and G. Athiappan</i>	
Using Wikipedia Anchor Text and Weighted Clustering Coefficient to Enhance the Traditional Multi-document Summarization	390
<i>Niraj Kumar, Kannan Srinathan, and Vasudeva Varma</i>	
Extraction of Relevant Figures and Tables for Multi-document Summarization (Verifiability Award)	402
<i>Ashish Sadh, Amit Sahu, Devesh Srivastava, Ratna Sanyal, and Sudip Sanyal</i>	
Towards Automatic Generation of Catchphrases for Legal Case Reports (Best Paper Award, Third Place)	414
<i>Filippo Galgani, Paul Compton, and Achim Hoffmann</i>	

Applications

A Dataset for the Evaluation of Lexical Simplification (Invited Paper)	426
<i>Jan De Belder and Marie-Francine Moens</i>	
Text Content Reliability Estimation in Web Documents: A New Proposal	438
<i>Luis Sanz, Héctor Allende, and Marcelo Mendoza</i>	
Fine-Grained Certainty Level Annotations Used for Coarser-Grained E-Health Scenarios: Certainty Classification of Diagnostic Statements in Swedish Clinical Text	450
<i>Sumithra Velupillai and Maria Kvist</i>	
Combining Confidence Score and Mal-rule Filters for Automatic Creation of Bangla Error Corpus: Grammar Checker Perspective	462
<i>Bibekananda Kundu, Sutanu Chakraborti, and Sanjay Kumar Choudhury</i>	
Predictive Text Entry for Agglutinative Languages Using Unsupervised Morphological Segmentation (Best Student Paper Award)	478
<i>Miikka Silfverberg, Krister Lindén, and Mirka Hyvärinen</i>	
Comment Spam Classification in Blogs through Comment Analysis and Comment-Blog Post Relationships	490
<i>Ashwin Rajadesingan and Anand Mahendran</i>	
Detecting Players Personality Behavior with Any Effort of Concealment	502
<i>Fazel Keshtkar, Candice Burkett, Arthur Graesser, and Haiying Li</i>	
Author Index	515

Thinking Outside the Box for Natural Language Processing

Srinivas Bangalore

AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
`srini@research.att.com`

Abstract. Natural Language Processing systems are often composed of a sequence of transductive components that transform their input into an output with additional syntactic and/or semantic labels. However, each component in this chain is typically error-prone and the error is magnified as the processing proceeds down the chain. In this paper, we present details of two systems, first, a speech driven question answering system and second, a dialog modeling system, both of which reflect the theme of tightly incorporating constraints across multiple components to improve the accuracy of their tasks.

1 Introduction

Speech and natural language processing system typically aim to transform an input natural language utterance into a representation that can be acted upon by a domain specific inference engine to affect the world of a given application. This transformation from the input utterance to the target representation might be carried out through a series of processing components (but sometimes may be achieved through a direct mapping). Each component adds additional labels to its input and passes the result as an output to the next component in the processing chain. These labels are typically linguistically interpretable, however, the labels could be designed specifically for the application at hand as well. Some examples of such processing chains include: (a) Part-of-speech (POS) tagging, syntactic parsing and semantic role labeling (b) Speech recognition, named entity detection and call type classification.

While the individual components of an NLP chain have been compartmentalized for software modularity, it is widely acknowledged that there is interdependence between components with information from upstream and downstream components need to interact to provide an optimal solution. However, most NLP systems typically pass down a single hypothesis from each component to the next. Given that the component modules in a chain are error-prone at their task, the error is progressively magnified as the input is processed down the chain. In this paper, we discuss the strengths and weaknesses of a few NLP architectures that are designed to address this issue. We will present the details of a system which instantiates one of these architectures.

2 NLP Architectures that Alleviate Error Propagation

In this section, we discuss three different architectures for NLP systems that address the issue of error propagation in pipeline architectures.

1. **Broadening the pipe:** In this architecture, each component in the pipeline passes down more than one solution to its downstream component, with the expectation that the correct solution might be among one of the multiple hypotheses, even if not the first hypothesis. The multiple hypotheses might be presented as alternate inputs to the next component, or as a compactly encoded single input, for example, as word lattices or forests. The component consuming the input would have to be able to cope with such multiple hypotheses and produce multiple hypotheses of its own to be passed downstream. With each component represented as a discriminative model that uses features extracted from preceding component, the incorporation of multiple hypotheses from the preceding component into the feature extraction of the discriminative model is non-trivial. It often amounts to running the model multiple times once for each input hypothesis.
2. **Constraint programming:** In this architecture, an NLP task is formulated as a constraint programming problem. Variables are introduced that represent the entities of interest in the task and constraints are encoded for each component as well as for the interdependence between the components. Given these ingredients, a constraint solver is used to obtain the values for the variables that optimize the objective function. This architecture is a radical departure from the pipeline architecture and has only been explored in very few NLP tasks (e.g. anaphora resolution). With the increasing speed and computational efficiency of constraint solvers, this approach shows promise.
3. **Finite-state models:** In this architecture, an NLP task is modeled using finite state models which provide the unique feature of being modular while at the same time guaranteeing optimal solution. If an NLP problem can be represented (or approximated) as a finite-state transduction, then the benefits of finite-state composition can be exploited to arrive at a modular architecture and yet produce an optimal solution. One such instance of tight coupling between several transductive components to achieve improved accuracy is presented in the context of speech-driven question retrieval system detailed below.

3 Speech-Driven Question Retrieval System

We describe the speech-driven query retrieval application (Qme!) [14]¹ in this section. The user of this application provides a spoken language query to a mobile device intending to find an answer to her question. User input could be either a short business listing query (e.g., *pizza hut near urbana illinois*) or a

¹ Parts of this section are derived from [13].

complete question (e.g., *how do I fix a leaky dishwasher, what movies are playing in los angeles california*).

As shown in Figure 1, we classify questions to Qme! into three categories and apply different retrieval methods.

1. Business listing questions: These questions typically contains a business name, optionally followed by a city and state to indicate the location of the business, such as *pizza hut near urbana illinois*. User input with a business category (*laundromats in madison*) and without location information (*hospitals*) are some variants supported by this application. The result from speech recognition is parsed to identify the business name (aka SearchTerm) and the city/state segments that the user is interested in. The parsed information is used to search a business listing database of over 10 million entries to retrieve the entries pertinent to the user query.
2. Static questions: These are general user queries that are not constrained to be of any specific question type such as *what, where, when, how* questions. The answers to these questions remain mostly the same irrespective of when and where the questions are asked. Some examples of such queries include *what is the fastest animal in water?, how do you make a bloody mary mix?, when is George Washington's birthday?, why is the sky blue?*. The result of the speech recognizer is used to search a large corpus of human generated question-answer pairs to retrieve the answers pertinent to the user's static questions. This ensures high accuracy for the answers retrieved (if found in the archive) and also allows us to retrieve related questions on the user's topic of interest.
3. Dynamic questions: The answers to these questions depend on when and where they are asked. For such questions, the above method results in less than satisfactory and sometimes inaccurate answers. Examples of such questions are *what is the stock price of General Motors?, who won the baseball game last night?, what is playing at the theaters near me?* For the dynamic questions, the answers are retrieved by querying a web form from the appropriate web site (for example, www.fandango.com for movie information).

For all user queries, the result from the speech recognizer can be a single-best string or a weighted word lattice. The retrieved results are then ranked.

4 Tightly Coupling Speech Recognition and Search

Most of the speech-driven search systems [15,19] use the 1-best output from the speech recognizer (ASR) as the query for the search component. Given that ASR 1-best output is likely to be erroneous, this serialization of the ASR and search components might result in sub-optimal search accuracy. As will be shown in our experiments, the oracle word/phrase accuracy using n -best hypotheses is far greater than the 1-best output. However, using each of the n -best hypothesis as a query to the search component is computationally sub-optimal since the strings in the n -best hypotheses usually share large subsequences with each other. A

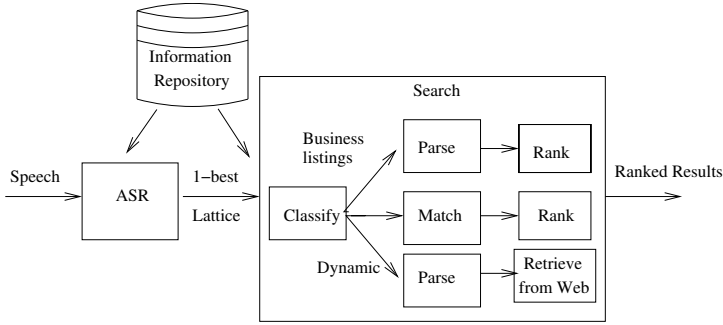


Fig. 1. The architecture of the speech-driven question-answering system

lattice representation of the ASR output, in particular, a word-confusion network (WCN) transformation of the lattice, compactly encodes the n -best hypothesis with the flexibility of pruning alternatives at each word position. An example WCN is shown in Figure 2. The weights on the arcs are to be interpreted as costs and the best path in the WCN is the lowest cost path from the start state to the final state. Note that the 1-best path in Figure 2 is *how old is mama*, while the input speech was *how old is obama* which also is in the WCN, but at a higher cost.

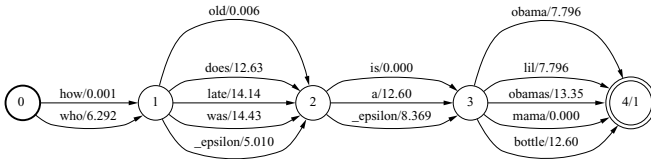


Fig. 2. A sample word confusion network with arc costs as negative logarithm of the posterior probabilities

4.1 Representing Search Index as an FST

In order to exploit WCNs for Search, we have implemented our own search engine instead of using an off-the-shelf search engine such as Lucene [11]. We index each business listing (d) in our data that we intend to search using the words (w_d) in that listing. The pair (w_d, d) is assigned a weight ($c_{(w_d, d)}$) using different metrics, including the standard $tf * idf$, as explained below. This index is represented as a weighted finite-state transducer ($SearchFST$) as shown in Figure 3 where w_d is the input symbol, d is the output symbol and $c_{(w_d, d)}$ is the weight of that arc.

The FST search index for the question answer repository is built as follows. We index each question-answer (QA) pair from our repository ((q_i, a_i) , qa_i for short) using the words (w_{q_i}) in question q_i . This index is represented as a weighted finite-state transducer ($SearchFST$) as shown in Figure 3. Here a word w_{q_i} (e.g

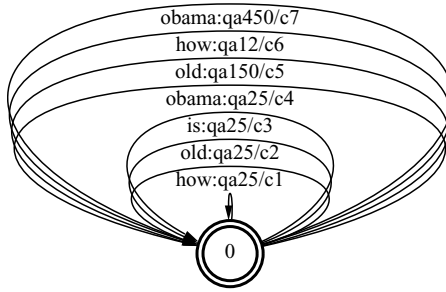


Fig. 3. An example of an FST representing the search index related to the QA repository

old) is the input symbol for a set of arcs whose output symbol is the index of the QA pairs where *old* appears in the question. The weight of the arc $c_{(w_{q_i}, q_i)}$ is one of the similarity based weights discussed in Section 4.1. As can be seen from Figure 3, the words *how*, *old*, *is* and *obama* contribute a score to the question-answer pair *qa25*; while other pairs, *qa150*, *qa12*, *qa450* are scored by only one of these words.

4.2 Query Relevance Metrics

For a given user query, we retrieve documents (i.e., either business listings or question-answer pairs) from the information repository based on the similarity of match between the user’s query and each of the documents (d) in the repository.

The problem of answering user queries that are classified by the system as static questions is formulated as follows². Given a question-answer archive $\mathbf{QA} = \{(q_1, a_1), (q_2, a_2), \dots, (q_N, a_N)\}$ of N question-answer pairs, and a user’s question q_u , the task is to retrieve a subset $\mathbf{QA}^r = \{(q_1^r, a_1^r), (q_2^r, a_2^r), \dots, (q_M^r, a_M^r)\}$ $M \ll N$ using a selection function *Select* and rank the members of \mathbf{QA}^r using a scoring function *Score* such that $Score(q_u, (q_i^r, a_i^r)) > Score(q_u, (q_{i+1}^r, a_{i+1}^r))$. Here, we assume $Score(q_u, (q_i^r, a_i^r)) = Score(q_u, q_i^r)$, that is, the relevance score of a question-answer pair to a user’s question is approximated as the similarity score of the question in the corpus to the user’s question.

The *Select* function is intended to select the matching questions that have high “semantic” similarity to the user’s question. For this purpose, we use **TF-IDF** metric [16] which involves term frequency (*tf*) and inverse document frequency (*idf*).

$$Score(d) = \sum_{w \in Q} tf_{w,d} idf_w \quad (1)$$

² User queries that are classified as dynamic questions are answered by querying the Deep web using web forms.

Ranking of the members of the retrieved set can be based on the scores computed during the selection step or can be independently computed based on other criteria such as popularity of the question, credibility of the source, temporal recency of the answer, geographical proximity to the answer origin.

4.3 Search Process Using FSTs

Given a user’s speech query, the process of obtaining search results is defined by equations 2 through 7. In these equations, \circ indicates composition of two finite state transducers (or one transducer and one acceptor); and π_2 indicates that the output of the given transducer is projected (or preserved) to obtain a new automaton.

The user’s speech query, after speech recognition (either 1-best or WCN), is represented as an FSA Q . The n -grams of Q and their counts are constructed using the function *CountNgrams* and the result is represented as another FSA (N_Q) as shown in (2). For this construction, we follow the algorithm presented in 4. The result is that, if Q is an unweighted FSA with one path (e.g. a 1-best ASR output), N_Q contains the set of n -grams of Q as paths and the frequencies of those n -grams as the path weights. If Q is a weighted FSA (e.g. a WCN), the paths in the N_Q represent the set of n -grams in the WCN and the path weights correspond to the weighted frequencies of the n -grams of Q .

The weighted FSA N_Q is then composed with the *SearchFST* (3). For numerical stability, the weights are converted as negative logarithm scores, and the composition operation adds these scores. We retrieve the 1-best path (R_1) from the result of the composition D_1 (4). R_1 represents that particular question with the best score (a combination of acoustic, language model and search score) in the repository that also contains the n -grams in N_Q . We use R_1 as the rescored output from the ASR, compute its n -grams N_{R_1} (5) and compose N_{R_1} with the *SearchFST* (6) to obtain all the arcs ($w_q, d_{w_q}, c_{(w_q, d_{w_q})}$) where w_q is a query term, d_{w_q} is either a QA index or a business listing index containing the query term and, $c_{(w_q, d_{w_q})}$ is the weight associated with that pair. Using this information, we aggregate the weight for a document (which is either a QA pair or a business listing) (d_q) across all query words (using *fsmdeetermine*) and rank the retrieved QAs in the descending order of this aggregated weight.³ We select the top N QA pairs (7) from this ranked list.

$$N_Q = \text{CountNgrams}(Q) \quad (2)$$

$$D_1 = \pi_2(N_Q \circ \text{SearchFST}) \quad (3)$$

$$R_1 = \text{fsmbestpath}(D_1, 1) \quad (4)$$

³ The FSA is converted into the *real semiring* before *fsmdeetermine* to allow for aggregation of weights.

$$N_{R_1} = \text{CountNgrams}(R_1) \quad (5)$$

$$D_2 = \pi_2(N_{R_1} \circ \text{SearchFST}) \quad (6)$$

$$\text{Top}N = \text{fsmbestpath}(\text{fsmdeterminize}(D_2), N) \quad (7)$$

While the computational complexity of the n -gram counting is $O(|q|)$, where $|q|$ is the number of arcs in q , the composition operation on two FSTs is $O(m * n)$ where m is the number of states of the first FST and n is the number of states of the second FST. Determinization of FSTs in general is $O(2^p)$ where p is the number of states, however, in our approach, the FST D_2 has $n+1$ states for an n -gram based model. We have built this model on over 30 million documents in the *SearchFST*, without any scalability problems. Extensions to this model using lazy evaluation techniques [15] can be used in cases where scalability becomes an issue.

4.4 Experiments and Results

In this section, we present the results of our experiments on tight coupling the ASR and Search components. We show that we not only improve the relevance of search results, but we also improve the speech recognition accuracy. We illustrate this for both the business listing queries as well as the general static queries.

Speech-driven query retrieval. We have a fairly large data set consisting of over a million question-answer pairs collected by harvesting the web. In order to evaluate the retrieval methods discussed earlier, we use a test set of 645 QA pairs where the queries may not match any question in the database exactly. The questions, however, also have a human generated answer that is used in our evaluations. We use a different set of 250 *speech* queries as the development set. In Table 1, we show the Word and Sentence Accuracy measures for the best path in the WCN before and after the composition of *SearchFST* with the WCN on the development and test sets. We note that by integrating the constraints from the search index, the ASR accuracies can be improved by about 1% absolute.

Table 1. ASR accuracies of the best path before and after (in parentheses) the composition of SearchFST

Set	# of utterances	Word Accuracy	Sentence Accuracy
Dev Set	250	77.1(78.2)	54(54)
Test Set	645	70.8(72.1)	36.7(37.1)

In order to investigate the impact of WCNs on search accuracy, we computed the search results by integrating the ASR WCNs with the *SearchFST* on the speech utterances of the *Unseen* set. The integration of the ASR WCNs with the *SearchFST* produces higher 1-best and 20-best search accuracy (70.12/78.52) compared to ASR 1-best (69.13/75.81). This indicates that the WCNs have information that can be utilized effectively to improve Search retrieval accuracy.

Dynamic questions are questions whose answers vary based on the place and time of the question, for example, *what movies are playing tonight*, *where is the closest walmart*, *when is the next train to new york city*. In order to answer such questions, we first classify a question as either static or dynamic using semi-supervised methods. For the static questions, we use the finite-state transducer-based search technique described in the previous section. For the questions classified as dynamic, we first identify the topic of the question (for example, *movies*), and then we parse the query to obtain relevant parameters that are needed to query a content aggregator web site. For example, if the question is *what movies are playing in glendale california?*, we parse out the city and state information and use it to query a movie aggregator site like www.fandango.com. A detailed report of our work on answering dynamic questions is presented in [14]. The finite-state transducer-based parser is described in the following section.

5 Finite-State Transducer-Based Parser

In this section, we present a method for parsing the segments of a user’s utterance to identify the parameters to be passed to query the information repository. These methods work not only on 1-best ASR output but can transparently scale to take word lattices from ASR as input. We encode the problem of parsing as a weighted finite-state transducer (FST). This encoding allows us to apply the parser on ASR 1-best as well as ASR WCNs using the composition operation of FSTs.

We formulate the parsing problem as associating with each token of the input a label indicating whether that token belongs to one of a search term (*st*), location term (*lt*) or neither (*null*). Thus, given a word sequence ($W = w_1, \dots, w_n$) output from ASR, we search of the most likely label sequence ($T = t_1, \dots, t_n$), as shown in Equation 8. We use the joint probability $P(W, T)$ and approximate it using an k -gram model as shown in equations 9 and 10. The sequence w_{i-1}^{i-k+1} represents the $k - 1$ consecutive words of the history and t_{i-1}^{i-k+1} represents the $k - 1$ consecutive tags of this history.

$$T^* = \underset{T}{\operatorname{argmax}} P(T|W) \quad (8)$$

$$= \underset{T}{\operatorname{argmax}} P(W, T) \quad (9)$$

$$= \underset{T}{\operatorname{argmax}} \prod_i^n P(w_i, t_i | w_{i-1}^{i-k+1}, t_{i-1}^{i-k+1}) \quad (10)$$

A k -gram model can be encoded as a weighted finite-state acceptor (FSA) [3]. The states of the FSA correspond to the k -gram histories, the transition labels to

the pair (w_i, t_i) and the weights on the arcs are $-\log(P(w_i, t_i | w_{i-1}^{i-k+1}, t_{i-1}^{i-k+1}))$. The FSA also encodes back-off arcs for purposes of smoothing with lower order k -grams. An annotated corpus of words and labels is used to estimate the weights of the FSA. A sample corpus is shown in Table 2.

Table 2. A Sample set of annotated sentences

1. pizza_st hut_st new_lt york_lt new_lt york_lt
2. home_st depot_st around_null san_lt francisco_lt
3. please_null show_null me_null indian_st restaurants_st in_null chicago_lt
4. pediatricians_st open_null on_null sundays_null
5. hyatt_st regency_st in_null honolulu_lt hawaii_lt

The FSA on the joint alphabet is converted into an FST. The paired symbols (w_i, t_i) are reinterpreted as consisting of an input symbol w_i and output symbol t_i . The resulting FST (M) is used to parse the 1-best ASR (represented as FSTs (I)), using composition of FSTs and a search for the lowest weight path, as shown in (11). The output symbol sequence (π_2) from the lowest weight path is T^* .

$$T^* = \pi_2(\text{Bestpath}(I \circ M)) \quad (11)$$

Equation (11) shows a method for parsing the 1-best ASR output using the FST. However, a similar method can be applied for parsing WCNs. The WCN arcs are associated with a posterior weight that needs to be scaled suitably to be comparable to the weights encoded in M . We represent the result of scaling the weights in WCN by a factor of λ as WCN^λ . The value of the scaling factor is determined empirically. Thus the process of parsing a WCN is represented by (12).

$$T^* = \pi_2(\text{Bestpath}(WCN^\lambda \circ M)) \quad (12)$$

5.1 Experiments

We have access to text query logs consisting of 18 million queries annotated with SearchTerm and LocationTerm labels. In addition to these logs, we have access to 11 million unique business listing names and their addresses. We use the combined data to train the parameters of the parsing model as discussed in the previous section. We tested our approach on three data sets, which in total include 2686 speech queries. These queries were collected from users using mobile devices from different time periods. Labelers transcribed and annotated the test data using SearchTerm and LocationTerm tags.

We use an ASR with a trigram-based language model trained on the query logs. Table 3 shows the ASR word accuracies on the three data sets. The accuracy is the lowest on Test1, in which many users were non-native English speakers and a large percentage of queries are not intended for business search.

Table 3. ASR Performance on three Data Sets

Data Sets	Number of Speech Queries	Word Accuracy
Test1	1484	70.1
Test2	544	82.9
Test3	658	77.3

We measure the parsing performance in terms of extraction accuracy on the two non-filler slots: SearchTerm and LocationTerm. Extraction accuracy computes the percentage of the test set where the string identified by the parser for a slot is exactly the same as the annotated string for that slot.

In Tables 4 and 5, we report the parsing performance for the FST-based approach. We note that the FST-based parser on a WCN improves the SearchTerm and LocationTerm extraction accuracy over ASR 1-best, an improvement of about 1.5%. The performance under “Oracle path” shows the upper bound for the parser using the oracle path⁴ from the pruned WCN. We pruned WCN by only retaining arcs that are within *thresh* (=4 in our experiments) of the lowest cost arc between two states.

Table 4. SearchTerm extraction accuracy using the FST approach

Data Sets	SearchTerm Extraction Accuracy			
Input	ASR 1-best	WCN	Oracle Path 4	Transcription
Test1	56.9	57.4	65.6	92.2
Test2	69.5	71.0	81.9	98.0
Test3	59.2	60.6	69.3	96.1

Table 5. LocationTerm extraction accuracy using the FST approach

Data Sets	LocationTerm Extraction Accuracy			
Input	ASR 1best	WCN	Oracle Path 4	Transcription
Test1	79.8	79.8	83.8	95.1
Test2	89.4	89.4	92.7	98.5
Test3	87.1	87.1	89.3	97.3

We evaluated the impact of parsing performance on search accuracy. In order to measure search accuracy, we first collected a reference set of search results for our test utterances. For this purpose, we submitted the human annotated two-field data to the search engine (<http://www.yellowpages.com/>) and extracted

⁴ Oracle text string is the path in the WCN that is closest to the reference string in terms of Levenshtein edit distance.

the top 5 results from the returned pages. The returned search results are either business categories such as “Chinese Restaurant” or business listings including business names and addresses. We considered these results as the reference search results for our test utterances.

In order to evaluate our voice search system, we submitted the two fields resulting from the query parser on the ASR output (1-best/WCN) to the search engine. We extracted the top 5 results from the returned pages and we computed the Precision, Recall and F1 scores between this set of results and the reference search set. In Table 6, we report the search performance. The overall improvement in search performance is not as large as the improvement in the slot accuracies between using ASR 1-best and WCNs.

Table 6. Search performances using the FST approach

Data Sets		Precision	Recall	F1
ASR	Test1	71.6	64.3	67.8
1-best	Test2	79.6	76.0	77.7
	Test3	72.9	67.2	70.0

	Test1	70.5	64.7	67.5
WCN	Test2	80.3	77.3	78.8
	Test3	72.9	68.1	70.3

6 Human-Machine Dialog Modeling

There is another sense in which “thinking outside the box” is appropriate for NLP. It is to do with features or constraints that are beyond the confines of a component and yet have an influence on the output of a component. An example of such a system is dialog modeling where the interpretation of a user’s utterance and the generation of the system’s output are tightly coupled. Human-machine dialog modeling has been traditionally thought of a sequence of components starting with speech recognition, followed by spoken language understanding, dialog management and finally language generation and speech synthesis. In previous work [18], which we summarize here, we have designed models of dialog that tightly couple all the components in a dialog system without the ill-effects of error propagation.

In this work, we adopt the shared-plan model of dialog [12]. In this model, a task-oriented dialog is result of incremental creation of a shared plan by the dialog participants. The shared plan is represented as a single tree T that incorporates the task/subtask structure, dialog acts, syntactic structure and lexical content of the dialog, as shown in Figure 4. A task is a sequence of subtasks. A subtask is a sequence of subtasks and/or of dialog acts. Each dialog act corresponds to one clause spoken by one speaker, customer (c^u) or agent (c^a), for which we may have acoustic, lexical, syntactic and semantic representations. For example, we have annotated each utterance/clause of a corpus of human-human

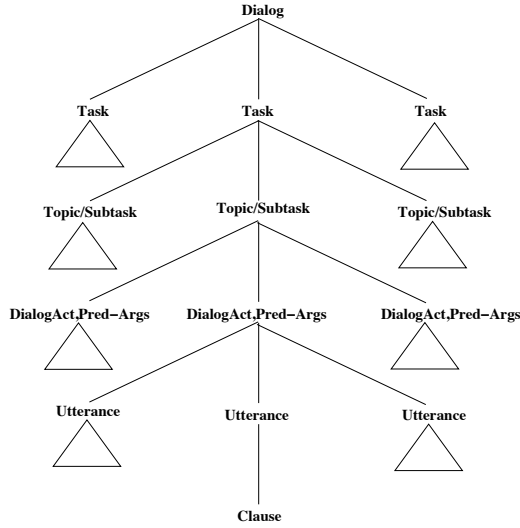


Fig. 4. A schema of a shared plan tree for a dialog

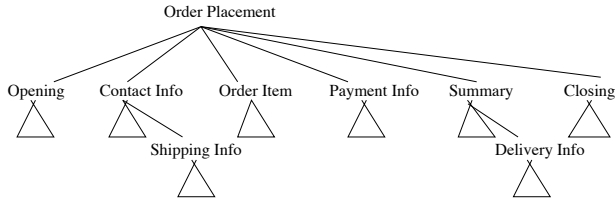


Fig. 5. Sample output (subtask tree) from a parse-based model for the catalog ordering domain

dialogs (CHILD) in the catalog ordering domain with the speaker, the words, the part of speech tags, the supertags, occurrences of task-related entities, and coreference information. Figure 5 shows the subtask tree for a sample dialog in the CHILD corpus.

In a departure from our previous work (e.g. [8,6]), we distinguish between two types of dialog act: *task-level* dialog acts and *grounding* dialog acts.

- Task-level dialog acts contribute to a subtask.
- Grounding dialog acts, by contrast, indicate the degree to which the recipient of a task-level dialog act understands and agrees with its content.

As the dialog proceeds, an utterance from a participant is accommodated into the subtask tree in an incremental manner, much like an incremental syntactic parser accommodates the next word into a partial parse tree [2]. An illustration of the incremental evolution of dialog structure is shown in Figure 6.

Based on this model, the dialog management functionalities we need to support are:

- Prediction of system utterances – generation of output utterances is driven by the right frontier of the subtask tree. First, a subtask label is predicted (either to continue the current subtask, or to start a new one). Then, task-level and grounding-level dialog acts are predicted (the act predicted may be *None*). We will use **STP** to refer to subtask label prediction, **TDAP** to refer to task-level dialog act prediction and **GDAP** to refer to grounding-level dialog act prediction.
- Interpretation of user utterances – interpretation of input utterances is constrained by the right frontier of the subtask tree. Each input utterance is assigned grounding-level and task-level dialog acts (the act assigned may be *None*). Then, the input utterance is assigned a subtask label (either a current open subtask, or a new one), and incorporated into the subtask tree. We will use **GDAC** to refer to grounding-level dialog act assignment/classification, **TDAC** to refer to task-level dialog act classification and **STC** to refer to subtask label classification.
- Maintenance of the dialog state – in this case, maintenance of the dialog state amounts to maintenance of the subtask tree.

In a dialog system, it can be helpful to have an idea of what the user is likely to say next (for example, to set the language model in the speech recognizer [20]). So we also support prediction of the grounding- and task-level dialog acts of user utterances based on the dialog thus far.

6.1 The Dialog Manager

We implement our dialog model as an incremental parser operating over the subtask tree for a dialog. This has several advantages: it is tightly tied to the dialog model; we can use known parsing methods that have known efficiency properties; and maintenance of dialog state comes automatically as the output of the parser. A dialog parser can produce a “shallow” or “deep” tree structure. A shallow parse is one in which utterances are grouped together into subtasks, but the dominance relations among subtasks are not tracked. We call this model a *chunk-based* dialog model [7]. The chunk-based model has limitations. For example, dominance relations among subtasks are important for dialog processes such as anaphora resolution [9]. Also, the chunk-based model is representationally inadequate for center-embedded nestings of subtasks, which do occur in both CHILD and MapTask, although less frequently than the more prevalent “tail-recursive” structures.

We use the term *parse-based* dialog model to refer to deep parsing models for dialog which not only segment the dialog into chunks but also predict dominance relations among chunks. For this paper, we used a **shift-reduce** parser [10,17]. This parser operates on the subtask tree for the dialog incrementally, from left-to-right, with access only to the preceding dialog context, as shown in Figure 6. The

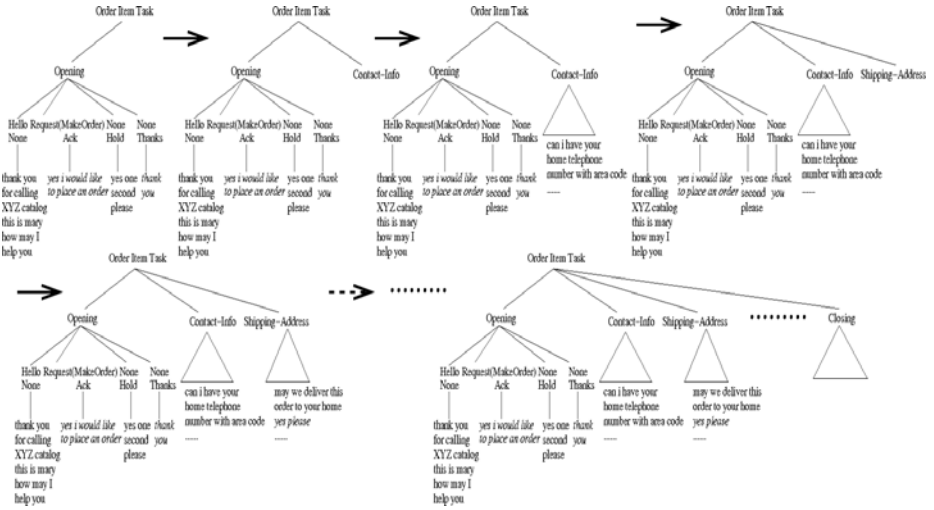


Fig. 6. An illustration of incremental evolution of dialog structure. Internal nodes in the dialog tree are labeled with tasks/subtasks. Leaves are labeled with task-level dialog acts (top) and grounding dialog acts (bottom), as well as with utterances/clauses.

parser shifts each utterance on to the stack. It then inspects the stack and decides whether to do one or more reduce actions that result in the creation of subtrees in the subtask tree. The parser maintains two data structures – a stack and a tree. The actions of the parser change the contents of the stack and create nodes in the dialog tree structure. The actions for the parser include *unary-reduce-X*, *binary-reduce-X* and *shift-X*, where X is each of the non-terminals (subtask labels) in the tree. *Shift-X* pushes non-terminal X onto the stack; *binary-reduce-X* pops two tokens off the stack and pushes the non-terminal X; and *unary-reduce-X* pops one token off the stack and pushes the non-terminal X. Each type of *reduce* action creates a constituent X in the dialog tree and the tree(s) associated with the reduced elements as subtree(s) of X. At the end of the dialog, the output is a subtask tree.

The steps taken by our dialog parser to incorporate an utterance into the subtask tree depend on whether the utterance was produced by the *system* or the *user* (as shown in Figure 7). Consider the example subdialog A: *would you like a free magazine?* U: *no*. The processing of this dialog using our shift-reduce dialog parser would proceed as follows: first, STP predicts *shift-special-offer* for st^a ; TDAP^s predicts *YNP(Promotions)* for tda^a and GDAP^s predicts *None* for gda^a ; the generator outputs *would you like a free magazine?*; and the parser shifts a token representing this utterance onto the stack. Second, GDAP^u predicts *None* for gda^*u , and TDAP^u predicts either *Yes* or *No* for tda^*u (this information can be used to set language models or adjust parser weights). Third, the customer says *no*. GDAC classifies gda^u as *None* and TDAC classifies tda^u as *No*; STC

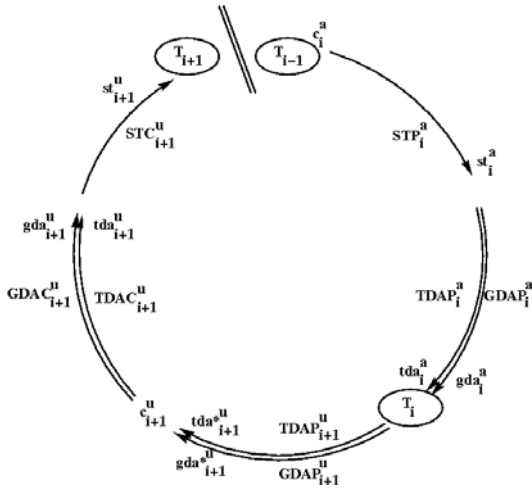


Fig. 7. Dialog management process

classifies st^u as *shift-special-offer* and *binary-reduce-special-offer*; and the parser shifts a token representing the utterance onto the stack, before popping the top two elements off the stack and adding the subtree for *special-offer* into the dialog’s subtask tree.

In our implementation, the subtasks of system utterance prediction, user utterance prediction and user utterance interpretation are each handled by a classifier trained on data from a corpus of human-human dialogs with annotation for dialog structure. Details of the experiments for the dialog model is presented in [8].

7 Discussion

In this paper, we have shown the benefits of modeling an NLP task as a series of finite-state models in terms of tight coupling through finite-state composition resulting in improved accuracy for both speech recognition and search.

However, not all NLP tasks can be accurately modeled as finite-state transductions. There may be approximations of the task that are finite-state transductions and it might be worth considering such approximations for the benefits derived from tight coupling.

We have also discussed a model for human-machine dialog, that in contrast with traditional multi-component models of dialog, tightly couples speech understanding and language generation tasks of a dialog manager with the aim of better exploiting constraints from each component and minimize the ill-effects of error propagation.

Acknowledgments. The work presented in this paper is a result of several collaborations with colleagues at AT&T Labs-Research, especially, Giuseppe di-Fabrizio, Amanda Stent, and Taniya Mishra. I wish to thank each of my collaborators for our joint work.

References

1. Acero, A., Bernstein, N., Chambers, R., Ju, Y., Li, X., Odell, J., Nguyen, P., Scholtz, O., Zweig, G.: Live search for mobile: Web services by voice on the cell-phone. In: Proceedings of ICASSP 2008, Las Vegas (2008)
2. Alexandersson, J., Reithinger, N.: Learning dialogue structures from a corpus. In: Proceedings of Eurospeech (1997)
3. Allauzen, C., Mohri, M., Riley, M., Roark, B.: A generalized construction of speech recognition transducers. In: ICASSP, pp. 761–764 (2004)
4. Allauzen, C., Mohri, M., Roark, B.: Generalized algorithms for constructing statistical language models. In: Proceedings of ACL (2003)
5. Bacchiani, M., Beaufays, F., Schalkwyk, J., Schuster, M., Strophe, B.: Deploying GOOG-411: Early lessons in data, measurement and testing. In: Proceedings of ICASSP 2008, Las Vegas (2008)
6. Bangalore, S., DiFabrizio, G., Stent, A.: Learning the structure of task-driven human-human dialog. *IEEE Transactions on Audio, Speech, and Language Processing Special Issue on New Approaches to Statistical Speech and Text Processing* 16(7), 1249–1259 (2008)
7. Bangalore, S., Fabrizio, G.D., Stent, A.: Learning the structure of task-driven human-human dialogs. In: Proceedings of COLING/ACL (2006)
8. Bangalore, S., Stent, A.: Incremental parsing models for dialog task structure. In: Proceedings of the ACL (2009)
9. Grosz, B., Sidner, C.: Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3) (1986)
10. Hall, J., Nivre, J., Nilsson, J.: Discriminative classifiers for deterministic dependency parsing. In: Proceedings of COLING/ACL (2006)
11. Hatcher, E., Gospodnetic, O.: *Lucene in Action*. In Action series. Manning Publications Co., Greenwich (2004)
12. Lochbaum, K.: A collaborative planning model of intentional structure. *Computational Linguistics* 24(4), 525–572 (1998)
13. Mishra, T., Bangalore, S.: Finite-state models for speech-based search on mobile devices. *Journal of Natural Language Engineering* 17, 243–264 (2010)
14. Mishra, T., Bangalore, S.: Qme!: A speech-based question-answering system on mobile devices. In: NAACL 2010 (2010)
15. Mohri, M., Pereira, F., Riley, M.: A Rational Design for a Weighted Finite-State Transducer Library. In: Wood, D., Yu, S. (eds.) WIA 1997. LNCS, vol. 1436, pp. 144–158. Springer, Heidelberg (1998)
16. Robertson, S.: Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation* 60, 503–520 (2004)
17. Sagae, K., Lavie, A.: A best-first probabilistic shift-reduce parser. In: Proceedings of COLING/ACL (2006)
18. Stent, A., Bangalore, S.: Incremental parsing models for dialog task structure. In: Proceedings of EACL (2009)
19. vlingo.com (2009), <http://www.vlingomobile.com/downloads.html>
20. Xu, W., Rudnicky, A.: Language modeling for dialog system. In: Proceedings of ICSLP (2000)

A Graph-Based Method to Improve WordNet Domains

Aitor González¹, German Rigau¹, and Mauro Castillo²

¹ IXA group UPV/EHU, Donostia, Spain
agonzalez278@ikasle.ehu.com,
german.rigau@ehu.com

² UTEM, Santiago de Chile, Chile
mcast@informatica.utem.cl

Abstract. WordNet Domains (WND) is a lexical resource where synsets have been semi-automatically annotated with one or more domain labels from a set of 170 hierarchically organized domains. The uses of WND include the power to reduce the polysemy degree of the words, grouping those senses that belong to the same domain. This paper presents a novel automatic method to propagate domain information through WordNet. We compare both labellings (the original and the new one) allowing us to detect anomalies in the original WND labels. We also compare the quality of both resources (the original labelling and the new one) in a common Word Sense Disambiguation task. The results show that the new labelling clearly outperform the original one by a large margin.

1 Introduction

Building large and rich knowledge bases is a very costly effort which involves large research groups for long periods of development. For instance, hundreds of person-years have been invested in the development of wordnets for various languages [1].

WordNet Domains [1] (WND) is a lexical resource where synsets have been semi-automatically annotated with one or more domain labels from a set of 165 hierarchically organized domains [2,3]. WND allows to reduce the polysemy degree of the words, grouping those senses that belong to the same domain [4].

But the semi-automatic method used to develop this resource was not free of errors and inconsistencies. For instance, noun synset <diver_n¹ frogman_n¹ underwater_diver_n¹> defined as *someone who works underwater* has domain *history* because it inherits from its hypernym <explorer_n¹ adventurer_n²>. WND has never been verified manually. Additionally, WND is aligned to WordNet 1.6 [5], and there is no version for 3.0.

We suggest a novel graph-based approach for improving WND. As a result we obtained a new semantic resource derived from WordNet Domains and aligned to WordNet 3.0.

¹ <http://wndomains.fbk.eu/>

After this short introduction, Section 2 describes a very simple method of inheritance used to fill the gaps that have arisen due to the porting process from WordNet 1.6 to 3.0. In section 3 we describe our novel graph-based method, based on the UKB algorithm, used to generate new domain labels aligned to WordNet 3.0. Section 4 presents an example of how to evaluate in a semi-automatic way the quality of the domain labels assigned in the original WND. Finally, section 5 presents an evaluation of the new domain labelling based on a common Word Sense Disambiguation task.

2 Domain Inheritance

WND was developed using WordNet 1.6. One consequence of the automatic mapping that we used to upgrade version 1.6 to 3.0 is that many synsets were left unlabeled (because there are new synsets, changes in the structure, etc.).

Thus, the first tasks undertaken has been to fill these gaps. For them, we have carried out a propagation of the labels by inheritance of nominal and verbal synsets. In WordNet, the adjectives are organized in terms of binary oppositions (antonymy) and similarity of meaning (synonymy). The structure of WordNet for adjectives and adverbs makes this spread not trivial. Therefore this simple process has been not carried out neither for adjectives nor for adverbs.

Consider the example shown in Figure 1. For nouns and verbs, we have worked on the assumption that synsets are mostly correctly labeled, and therefore we have worked exclusively on those synsets that had no labels at all. We inherited the label or labels from its hypernyms. If a synset has more than one hypernym, the domain labels are taken from all of them. During this phase has been taken into account the incompatibility between domain labels, preventing the same synset can be, for instance, both *factotum* and *biology*.

This process increased our domain information by nearly a 18-19%, as shown in Tables 1 and 2.

Table 1. Number of synsets with domain labels

PoS	Before	After	Increase
Nouns	66,595	83,286	+25%
Verbs	12,219	14,224	+16%
All	100,315	119,011	+19%

However, this process may also have propagated inappropriate domain labels to unlabeled synsets. In the next section we present some examples using a new graph-based method for propagating domain labels through WordNet. Additionally, the method can also be used to detect anomalies in the original WND labels.

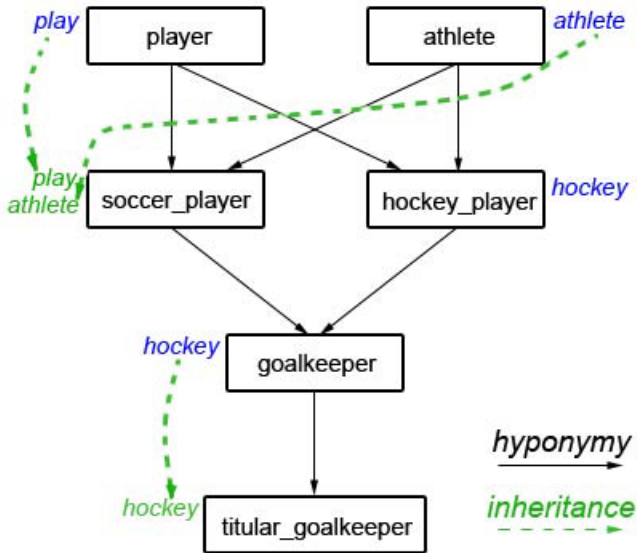


Fig. 1. Example of inheritance of domain labels

Table 2. Total number of domain labels

PoS	Before	After	Increase
Nouns	87,938	108,665	+24%
Verbs	13,026	15,051	+16%
All	124,551	146,899	+18%

3 A New Graph Based Method

UKB² algorithm [6] applies personalized PageRank on a graph derived from a wordnet. This algorithm has proven to be very competitive on Word Sense Disambiguation tasks and it is easily portable to other languages that have a wordnet [7]. Now, we present a novel use of the UKB algorithm for propagating information through a wordnet structure.

Given an input context, 'ukb_ppv' (*Personalized PageRank Vector*) algorithm outputs a ranking vector over the nodes of a graph, after applying a *Personalized PageRank* over it. We just need to use a wordnet as a knowledge base and pass to the application the contexts we want to process, performing a kind of *spreading activation* through the structure of a wordnet.

² <http://ixa2.si.ehu.es/ukb/>

As a context we used those synsets labelled with a particular domain. Thus, for each of the 169³ domain labels included in the MCR we generated a context. Each file contains the list of offsets corresponding to those synsets with a particular domain label. After creating the context file, we just need to execute 'ukb_ppv' that will return a ranking of the weights for each wordnet synset with respect to that particular domain.

Once made the process for all domains we will have the weight of each synset for each of the domains. Therefore, we know which are the highest weights for each domain and the highest weights for each synset. This allows us to estimate which synsets are more representative of each domain (those who have more weight in the ranking) and which domains are best for each synset (those who have attained a higher weight for that synset).

Basically, what we do is to mark some synsets with a domain (using the labels we already know from the original porting process) and use the wordnet graph to propagate the new labelling. We work on the assumption that a synset directly related to several synsets labelled with a particular domain (i.e. *biology*) would itself possibly be also related somehow to that domain (i.e. *biology*). Therefore, it makes no sense to use the domain *factotum* for this technique.

3.1 Propagating Domain Labels

We have generated two different knowledge bases. The first one only contains the original WordNet relations. The second one, also contains the relationships between glosses, increasing the size and richness of the knowledge base. Instructions for preparing the binary databases for UKB using WordNet relations are inside the downloadable file⁴ of the UKB package.

It has been necessary to generate a context file for each domain. Generating a context is as simple as creating a text file with the synset offsets that have the domain label. An example of a context file for the *rugby* domain can be seen in Figure 2. We can see a list of offsets representing synset of the Table 3.



Fig. 2. View of the format of a context file

One of the problems that comes up when analyzing the results is that the own domain labels of a synset have an unbalanced weight on the final ranking of that synset. Almost always the own labels of a synset appear in the top positions. In

³ Excluding *factotum* labels.

⁴ <http://ixa2.si.ehu.es/ukb/>

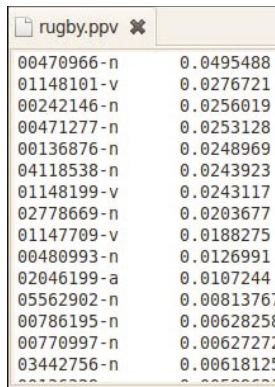
Table 3. List of synset with "rugby" as domain label

Synset	Variants
eng-30-00136876-n	goal-kick
eng-30-00242146-n	scrum, scrummage
eng-30-00470966-n	rugby, rugby_football, rugger
eng-30-00471277-n	knock_on
eng-30-01148101-v	hack
eng-30-01148199-v	hack
eng-30-04118538-n	rugby_ball

order to avoid this undesired effect, we generated new contexts, specific to each synset, and each domain. Thus, a synset can not vote for its own domains and only the rest of synsets decide the final weights of the ranking.

3.2 Post-processing

Once generated the context files, the UKB algorithm is executed. The result is a list with the weight for each synset for a domain. The next step is to sort the file by weight, highlighting those synsets that are more representative of the domain (Figure 3).



Synset	Weight
00470966-n	0.0495488
01148101-v	0.0276721
00242146-n	0.0256019
00471277-n	0.0253128
00136876-n	0.0248969
04118538-n	0.0243923
01148199-v	0.0243117
02778669-n	0.0203677
01147709-v	0.0188275
00480993-n	0.0126991
02046199-a	0.0107244
05562902-n	0.00813767
00786195-n	0.00628258
00770997-n	0.00627272
03442756-n	0.00618125

Fig. 3. Result of a PPV ranking sorted by weight (only the first lines are shown)

Furthermore, we can sort the result by synset. This allows us to, once we have a file for each domain, put them together in a matrix. Each line of this matrix will represent a synset, and the columns will be weights corresponding to each domain. The highest values of a line (synset) will be the more representative domains for that synset.

Table 4 shows the first ten domains and weights resulting from the application of this method on synset $\langle \text{diver}_n^1 \text{ frogman}_n^1 \text{ underwater_diver}_n^1 \rangle$ originally labeled as *history*, which seem to be incorrect. The suggestions of the algorithm seems to improve the current labeling because it suggests *sub* (possibly the best one) and *diving* (possibly, the second best option). Moreover, the method suggests the wrong label with a much lower weight.

Table 4. PPV weight rankings for sense diver_n^1

Weight	Domain
0.0144335:	sub
0.0015939:	diving
0.0001725:	swimming
0.0001297:	history
0.0000557:	nautical
0.0000529:	fashion
0.0000412:	jewellery
0.0000315:	ethnology
0.0000274:	archaeology
0.0000204:	gas

4 Analyzing Ranking Changes

It seems that the algorithm is able to generate a ranking in which the most appropriate labels obtain larger weights and also that avoiding the own labels of a synset reduces the weights for incorrect domain labels.

In the next experiment we study how to evaluate in a semi-automatic way the quality of the original labelling. To do that we check the domain labels of the synsets, taking into account the position they occupy in the weight vector. If a synset has ' n ' domain labels, the displacement is calculated for every label. For example, if a synset has two labels and one of the domains occupies the first position and the other the third one, they receive an offset of +0 and +1 respectively. That is, we calculate how many positions they moved from its original place. All those labels with an offset of six or greater are considered in the same group. Possibly, this test will allow us to discover wrong labeled synsets (or at least delimit the search) or to create a group of labels with a high value of reliability.

Therefore we tested the process for each PoS. The results obtained are in the Table 5.

Detecting the labels that have been displaced six or more positions (Table 5) allows us to recognize possible synset that have been labeled incorrectly. An example can be seen in Table 6.

Table 5. Method WN+gloss: Displacement of domain labels regarding their current position (separated by PoS)

PoS	Offset						
	0	1	2	3	4	5	6+
Nouns	55.52%	18.51%	10.06%	5.19%	1.95%	1.95%	6.82%
Verbs	40.46%	15.95%	13.39%	7.69%	4.56%	0.85%	17.09%
Adjectives	51.04%	21.35%	8.85%	2.60%	1.56%	4.17%	10.42%
Adverbs	60.40%	13.86%	5.94%	0.99%	4.95%	2.97%	10.89%
Total	54.48%	18.60%	10.07%	5.04%	2.06%	2.10%	7.65%

Results for 'ili-30-00747215-n':

- **Variants:** pornography_1 porno_1 porn_1 erotica_1 smut_5
- **Gloss:** creative activity (writing or pictures or films etc.) of no literary or artistic value other than to stimulate sexual desire
- **Domains:** law

Table 6. Method WN+gloss: UKB weight rankings for sense 1 of "porno"

Method WN+gloss	
Weight	Domain
0.000123453:	sexuality
0.000112444:	cinema
0.000077780:	theatre
0.000075525:	painting
0.000062377:	telecommunication
0.000060640:	publishing
0.000050370:	psychological_features
0.000047003:	photography
0.000046853:	artisanship
0.000040458:	graphic_arts

The example in Table 6 shows how the label *law* (incorrectly assigned) disappears from the first ten positions of the list. Instead, the algorithm suggests *sexuality* and *cinema*, which in this case seems to be much more appropriate.

5 Evaluation

To evaluate the new resources, we decided to compare the original labelling against the new domain labels that we have generated in a common Word Sense Disambiguation task.

Senseval-3 task 12 *Word-Sense Disambiguation of WordNet Glosses*⁵ was designed as an all-words task using as a gold standard the handtagged words provided by the eXtended WordNet ⁸.

Similarly, we selected as a gold standard, a random subset of 933 disambiguated words from the semantically disambiguated WordNet glosses⁶. This sample is available at <http://adimen.si.ehu.es/web/XWND>. The task is to try to select the correct sense of a target word appearing in its gloss. For example, consider synset $\langle \text{tortoiseshell}_n^3 \text{ tortoiseshell-cat}_n^1 \text{ calico_cat}_n^1 \rangle$ defined as a *cat having black and cream-colored and yellowish markings*. In this case, we should try to disambiguate which of the seven senses of the word *cat* is the one used in the gloss.

Our approach follows heuristic 5 from ⁹. Having a synset with a particular WordNet Domain label, this method selects those synsets from the target word of the gloss having the same Domain label. According to ⁹ this heuristic obtained a precision of 69.7%, a recall of 18.9% and it was applied only 27.1% of the cases on the WordNet 2.0 dataset.

The technique consists in choosing the synset that shares the domain labels with the synset defined by the gloss we are trying to disambiguate. At this point we must differentiate between the original labelling and those generated using our graph technique. One advantage of our labelling is that we have a ranking of the 169 domain labels, while the old labelling provides a limited number of labels. In the case of the original labels of WordNet Domains (WND) all available domain labels are used for the disambiguation task (varying between one and four domain labels per synset). In the case of the new labeling, we will check the results obtained using between one and five labels for the disambiguation task. We also use the *scorer2* software available on the Senseval-3 website⁷.

For those cases of multiple matches in the candidate synsets (when more than one synset shares the domain labels) we will choose those sharing more labels (where possible). That is, if we are using three domain labels to disambiguate, we will select as candidates those synsets that share the three labels with the synset defined by the gloss we are trying to disambiguate. In the case of ties, we choose all those synsets that matches (which decreases the score obtained after applying the software *scorer2*). If any of the candidate synsets shares the three domain labels, we will select all those who share two labels, and so on.

Following the example of the synset $\langle \text{tortoiseshell}_n^3 \text{ tortoiseshell-cat}_n^1 \text{ calico_cat}_n^1 \rangle$, labeled with *animals* and *biology* domain labels, the chosen senses will be synsets $\langle \text{cat}_n^1 \text{ true_cat}_n^1 \rangle$ and $\langle \text{cat}_n^7 \text{ big_cat}_n^1 \rangle$. The two chosen synsets share two domain labels with the synset $\langle \text{tortoiseshell}_n^3 \text{ tortoiseshell-cat}_n^1 \text{ calico_cat}_n^1 \rangle$ (*animals* and *biology*). None of the other senses of *cat* shares any of the domain labels with the synset $\langle \text{tortoiseshell}_n^3 \text{ tortoiseshell-cat}_n^1 \text{ calico_cat}_n^1 \rangle$.

⁵ <http://www.clres.com/SensWDisamb.html>

⁶ <http://wordnet.princeton.edu/glosstag.shtml>

⁷ <http://www.senseval.org/senseval3>

After performing this operation with the 933 glosses we will get the values for precision, recall and F1 score (Table 7). Nomenclature employed is as follows:

- **Method 0:** Disambiguation performed using the original WordNet Domains.
- **Method 1:** Disambiguation performed using the new labelling obtained using UKB and WordNet relations as a knowledge base (WN).
- **Method 2:** Disambiguation performed using the new labelling obtained using UKB and WordNet relations enriched with relations between glosses (WN+gloss).

Table 7. Precision, recall and F1 values obtained using *scorer2*

Label #	Method 0			Method 1			Method 2		
	P	R	F1	P	R	F1	P	R	F1
1	-	-	-	0.779	0.242	0.369	0.796	0.283	0.418
2	-	-	-	0.739	0.373	0.496	0.795	0.509	0.621
3	-	-	-	0.720	0.435	0.542	0.807	0.654	0.722
4	-	-	-	0.695	0.474	0.564	0.793	0.693	0.740
5	-	-	-	0.682	0.504	0.580	0.796	0.745	0.770
All	0.668	0.319	0.432	-	-	-	-	-	-

Additionally, if we look at the plots with all the values obtained for the new domains we will see that both methods outperform the original WordNet Domains (the line for the *Method 0* is shown as baseline). The plots are shown in Figure 4 (precision), 5 (recall) and 6 (F1 score). *Method 2* seems to be the most robust of the three, reaching an F1 score of 0.770 when using three domain labels.

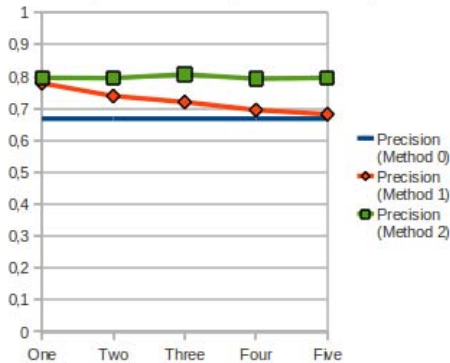


Fig. 4. Graphic showing *precision* values

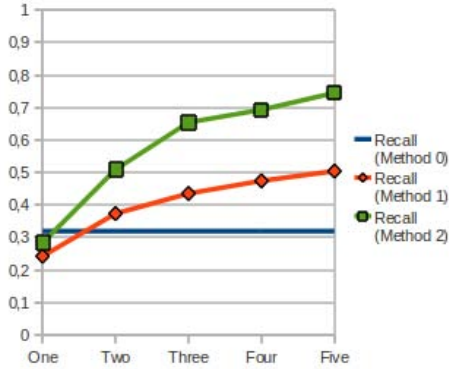


Fig. 5. Graphic showing *recall* values

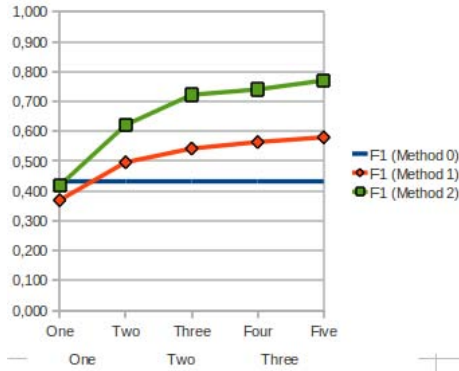


Fig. 6. Graphic showing *F1 score* values

6 Concluding Remarks

We have presented a new robust graph-based method which propagates domain information through WordNet. Firstly, we described a simple inheritance mechanism to complete unlabelled synsets from WordNet 3.0. Secondly, we provide some examples of the new domain labellings focussing on those synsets which provided larger variations. Thirdly, an empirical evaluation has been carried out in a common Word Sense Disambiguation task. On this task, the heuristic using the new WordNet Domains clearly outperforms by a large margin the one using the original WordNet Domains.

After these initial empirical tests, we drawn some preliminary conclusions:

1. The propagation method seems to provide some interesting results which deserve more research.
2. The gloss relations seems to provide useful knowledge for propagating domain information through WordNet.

Obviously, some improvements and further investigation are needed with these new resources. For instance, we need to develop an automatic method to select which label or labels finally assign to a particular synset. Moreover, not all domains affect in the same way due to its initial distribution through the WordNet structure. We also need to investigate different combinations of relations for creating the knowledge base used by UKB. For instance, using only gloss relations, or a particular subset of WordNet relations.

We also plan to try different combinations of methods and resources to improve the final result. For instance, we also plan to derive domain information from Wikipedia by exploiting WordNet++ [\[10\]](#).

Acknowledgments. We thank the IXA NLP group from the Basque Country University. This work was been possible thanks to its support withing the framework of the KNOW2 (TIN2009-14715-C04-04) and PATHS (FP7-ICT-2009-6-270082) projects.

We also wish to thank the reviewers for their valuable comments.

References

1. Vossen, P.: EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publishers (1998)
2. Magnini, B., Cavagli, G.: Integrating subject field codes into wordnet. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC), Athens, Greece (2000)
3. Bentivogli, L., Forner, P., Magnini, B., Pianta, E.: Revising WordNet Domains hierarchy: Semantics, coverage, and balancing. In: Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources, pp. 101–108 (2004)
4. Magnini, B., Satraparava, C., Pezzulo, G., Gliozzo, A.: The role of domains informations. In: Word Sense Disambiguation, Treto, Cambridge (2002)
5. Fellbaum, C.: WordNet. An Electronic Lexical Database. Language, Speech, and Communication. The MIT Press (1998)
6. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European chapter of the Association for Computational Linguistics (EACL 2009), Athens, Greece (2009)
7. Agirre, E., Cuadros, M., Rigau, G., Soroa, A.: Exploring knowledge bases for similarity. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), pp. 373–377. European Language Resources Association, ELRA (2010)
8. Mihalcea, R., Moldovan, D.: eXtended WordNet: Progress Report. In: Proceedings of NAACL Workshop WordNet and Other Lexical Resources: Applications, Extensions and Customizations, Pittsburg, PA, USA, pp. 95–100 (2001)

9. Castillo, M., Real, F., Asterias, J., Rigau, G.: The TALP systems for disambiguating WordNet glosses. In: Mihalcea, R., Edmonds, P. (eds.) *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 93–96. Association for Computational Linguistics, Barcelona (2004)
10. Navigli, R., Ponzetto, S.P.: Building a very large multilingual semantic network. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 216–225 (2010)

Corpus-Driven Hyponym Acquisition for Turkish Language

Savaş Yıldırım and Tuğba Yıldız

Department of Computer Science, Istanbul Bilgi University
Dolapdere, 34440 Istanbul, Turkey
{savasy,tdalyan}@cs.bilgi.edu.tr

Abstract. In this study, we propose a method for acquisition of hyponymy relations for the Turkish Language. This integrated method relies on both lexico-syntactic pattern and semantic similarity. Once the model has extracted the items using patterns it applies similarity based elimination of the incorrect ones in order to increase precision. We show that the algorithm based on a particular lexico-syntactic pattern for Turkish language can retrieve many hyponymy relations and also demonstrate that elimination based on semantic similarity gives promising results. We discuss how we measure the similarity between the concepts. The objective is to get better relevance and more precise results. The experiments show that this approach gives successful results with high precision.

Keywords: Corpus-based method, Hypernym/Hyponym, Semantic Lexicon, Semantic Similarity.

1 Introduction

This paper describes a pattern-based method to automatically extract hyponym relations of nouns from a Turkish corpus [1]. The Hypernym/Hyponym relation is one of the semantic relations that play an important role for supporting many NLP applications. The term hyponym used in this paper refer to the definition summarized as “hyponym is (a kind) of hypernym” [2].

In recent years, many algorithms and models have been developed to build semantic lexicon using corpus driven techniques. The main aim for building semantic lexicon is to label the words with semantic classes and to make a relation between the words (synsets). Lexical databases such as WordNet [2, 4, 5] include synsets with their relations between other synsets, where synsets refers to synonyms sharing identical meaning. Hypernym, hyponym, meronym and holonym are the main relations between the nouns. However, hand-built resources can be limited and misleading. So that automatically acquisition of semantic lexicon from a given corpus has many advantages when compared to hand-build lexicon. This approach automatically updates the lexicon and generates domain specific lexicon.

In our study, we focus on acquisition of is-a relation between nouns. In other words, the proposed model extracts hypernym/hyponym relations from a given corpus. First, it exploits particular linguistic patterns such as “NPs gibi CLASS“ (CLASS such as NPs), and produces hypernym/hyponym pairs. Several patterns has been used for several languages so far.

The observations show that four important patterns can be used to extract the hypernym/hyponym relation in Turkish. However three patterns are very misleading and their accuracy are very low. Due to the fact, we used only the most productive and reliable pattern “NPs gibi CLASS“ (CLASS such as NPs).

At the second phase, the model eliminates the spurious candidates according to some assumptions. We determined some rules to apply elimination process. Another elimination is based on semantic similarity at final step. All candidates are scored by their co-occurrence or vectors in a Word Space. According to semantic similarity score, insufficient candidates are eliminated.

As described, once the model has extracted hyponyms for a given hypernym, it applies some elimination methods to acquire more precise candidates as many as possible. All implementation and the results are described in the further sections.

2 Related Work

Of the projects in this field, Cyc [3] and WordNet [2, 4, 5] are the most useful to provide resources for NLP applications. Although these efforts are reliable and effective, they are also costly, time-consuming, limited and insufficient in some cases. To overcome these types of problems, various techniques have been proposed to automatically acquire semantic information.

In this study, we defined a corpus-driven method using lexico-syntactic patterns and integrated it with semantic similarity. Related previous pattern-based approaches are cited frequently in the literature. Previous attempts [6-10] used patterns to extract lexical information from Machine Readable Dictionaries (MRDs). Because of the limitations of MRDs, Hearst was the first to apply a pattern-based method to extract hyponyms from Grolier’s American Academic Encyclopedia [11, 12]. Another approach to build semantic lexicons for specific categories with sets of seed words [13, 14]. Caraballo also used conjunctions and appositives that appearing in the Wall Street Journal to build a hypernym-labeled noun hierarchy like Wordnet [15]. Another similar technique to Hearst’s was applied to look for patterns inside texts to extend an ontology and compare with WordNet [16]. Hearst patterns were used and refined to increase recall in Information Extraction (IE) systems, KNOWITALL [17-20].

Moreover, several researchers also used corpus-driven pattern-based methods to find Hypernym/Hyponyms [21-25]. Pattern-based methods have also been applied to web documents. Methods for acquiring named entities apply lexico-syntactic patterns to Web documents [26]. Many recent studies have been performed to construct semantic relations using web documents and wiki pages [27-30, 33]. There have been significant studies, which present unsupervised,

statistical, graph-based methods for automatic extraction of semantic relations [31, 32, 34, 35].

There have been few studies for acquiring semantic relations in Turkish. BalkaNet [36] is the first project to develop of a multilingual lexical database for Balkan languages such as Turkish WordNet. Although the project is not yet completed, it can be used for comparison with other studies. One attempt to extract only Hypernym/Hyponym relations [37], described in more detail in [38], used a dictionary TDK¹.

Another study applied a rule-based method in order to extract Hypernym relations between words in a dictionary [39]. Recent studies about deriving semantic word relations in Turkish from dictionary definitions have been developed [40, 41]. All studies on Turkish are based on a Turkish dictionary and compared with BalkaNet. Our study is the first major attempt using a pattern-based approach in a Turkish corpus.

The rest of the paper is organized into 3 sections. We explain our methodology of the study in Section 3. Implementation details are demonstrated in Section 4 and experimental results are shown in Section 5. Our conclusion is stated in Section 6.

3 Methodology

The methodology employed here proceeds in three stages to acquire the hyponyms for a given hypernym from the corpus.

3.1 STEP 1: Candidate Hyponym

The most precise acquisition methodology earlier applied by Hearst [11] relies on lexico-syntactic patterns. We start with the same idea of using the most productive and reliable pattern to acquire hyponyms with high accuracy. For the Turkish text, we observed the most precise pattern as follows:

"NPs gibi CLASS" (CLASS such as NPs)

It gives strong indication of is-a hierarchy. Given the syntactic pattern above, the algorithm extracts the candidate hyponyms that are recorded with their occurring frequency in the patterns. This count will be input to the scoring function for a later stage.

3.2 STEP 2: Elimination Based on Assumptions

Although the idea of relying on lexico-syntactic patterns can extract a sufficient number of instances, it can be risky because incorrect hyponyms are often extracted due to parsing and other errors. Since the extracted list can contain non-hyponym words which are strongly associated with hypernyms, we need to filter them out. For example, looking at the word kus/bird, the algorithm extracted migration, photo, lake and other associated words along with the bird

¹ Türk Dil Kurumu. <http://www.tdk.gov.tr>

types. Also, unassociated words can be retrieved by mistake due to polysemy or parsing errors. The objective of this step is to exclude these kinds of non-hyponyms and to acquire more precise candidates. According to our findings the partial exclusion can be performed by making some simple assumptions. The assumptions which we applied for this step are as follows:

- In our reliable pattern "NPs gibi CLASS" (CLASS such as NPs), we observed a grammatical rule that the actual hyponyms appears in nominative case and without any suffix. If a noun that appeared in the pattern NPs gibi CLASS is in nominative case, it will be passed. Or, the candidates in other cases such as accusative, dative, genitive etc. will be eliminated. This rule is very effective especially for an agglutinative language such as Turkish, since agglutinative languages have highly productive inflectional and derivational morphology.
- Hyponyms tend to have lower document frequency than their hypernyms.

3.3 STEP 3: Statistical Elimination

Although we filtered out non-hyponyms based on the assumptions above, we can still have erroneous words. The objective of this step is to eliminate some of these words by relying on semantic similarity. Our expectation at this phase is that non-hyponym words share low semantic similarity with other candidates and hypernyms, while real hyponyms must have strong semantic relations with their class. The candidates with low similarity scores are likely to be erroneous. This semantic similarity of two words can be reduced to their frequency of co-occurrence in a corpus. The more frequently two words occur together, the higher their similarity is. For the similarity, the marginal total of the words must be taken into account.

In order to compute the similarity between concepts and eliminate incorrect candidates, we used the cosine similarity measurement based on word space model which is a representational Vector Space as proposed in [42]. Schütze derives the word space model from the nearest neighbors of a given word w in the corpus. In brief, all words can be described through taking their co-occurrence relation with the words in a given window or a larger context. In a matrix, each row represents a word vector and each column represents a word. The $cell_{ij}$ records the number of times $word_i$ and $word_j$ co-occur together.

Dimension. For the Word Space model, the most significant task is how we determine the dimension and which words will be chosen. In our study, we applied the following selectional criteria. For a given hypernym, the words as the dimension of Word Space are derived depending on their semantic similarity with given hypernym or how they associate with it. The dimension words can be extracted from a global corpus or a local context. To build the dimension and to select the words of the dimension, we mine a global corpus size of 490M-tokens.

The words can be ranked by their statistical measures of association. We calculate bi-gram values of word pairs from the corpora and compute their chi-square coefficient. Chi-square, jaccard, dice are among the main measurements

for word similarity. In our experiment, we observed that the chi-square coefficient helps to gather the coherent words. Using the bi-grams in which the target word occurs ranked by their coefficient, we can obtain plausible word lists which could be clue/key for acquisition of hyponyms. The nouns, the adjective and the verbs are better potential indicators for understanding the meaning of the text than are other pos tags. Actually, to analyze how much each pos type contributes could be another direction of study on semantic similarity.

For the dimension, nouns, verbs and adjectives are chosen. Also some studies [34] use only verbs as their dimension. To balance the number of word types, we select K number of words for each type. Finally, the most associated K number of neighbors are chosen as the dimension of the space. In this study we selected K as 20.

A vector for a candidate hyponym is constructed from the nearest neighbor words which must be from our 60-word dimension obtained at the previous step. In brief we create a matrix in which the rows are vectors of each candidate and likewise each column represents a word from 60-word dimension. Each cell can refer to chi-square statistical score between a candidate hyponym and a word in the dimension. The target word (hypernym) is also added to the matrix along with these possible hyponyms to measure similarity between them.

Similarity Score. Briefly, all candidate hyponyms and the hypernym are represented in a candidate-by-word-list matrix. For this step, we need to measure similarities between the hyponyms and the target word as well. The following procedure can be applied to measure the closeness of each candidate to the centroid or target hypernym word.

For each candidate c_i in CHL:

$$\text{simple_score}(c_i) = \sum_{j=1}^N c_{ij} \quad (1)$$

where CHL is a candidate hyponym list, c_i is an element of the list and N is the size of dimension. Since the dimension extracted the word depending on similarity to hypernym, the summation can be considered as semantic similarity to hypernym. And the candidates can be ranked by that score and eliminated.

On the other hand, we observed that cosine similarity-based ranking gives more precision and recall values than the summation procedure described above. Cosine similarity is a well-known measurement using vectors as follows:

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (2)$$

Once the similarities between candidate list (and target word as well), have been calculated, we get CHL X CHL up triangle matrix or symmetric matrix. Using these similarity scores, an average similarity score (or centroid) is calculated. Summation of a row or column gives the closeness of a concept to the centroid; The closer, the more chance.

For the elimination phase, we denote that average similarity score as *sim-2nd*, the similarity between the candidate and the target hypernym as *sim-hypernym* and the number of occurrence in lexico-syntactic pattern in Step-1 as *freq*. Finally we apply the following scoring function:

$$score(cand) = \begin{cases} Pass, & if(cand-freq > K1) \\ Pass, & if(freq*sim-2nd*sim-hypernym > K2) \\ Fail, & otherwise \end{cases} \quad (3)$$

where K1 and K2 are specific thresholds for the domain. According to our observations K1=3 gives good performance; When the candidates appear more than three times in the patterns, they are more likely to be correct hyponyms and are automatically considered to have passed the test. In a list produced by lexico-syntactic patterns for fruit, 20 out of 56 candidates retrieved from pattern matching at Step-1 occur more than 3 times and all of them are correct candidates. Moreover, most of the words can appear in the pattern by mistake due to error in data, polysemy or parsing error. Scoring function defined above works like a decision list. The instances are checked against conditions in order. The first satisfied condition determines the output. So, if a candidate occurs less than 4, the defined formula will be checked, where *sim-2nd* and *sim-hypernym* weights are normalized. Thus, the score can be in the range [0-3]. According to the observation, K2 can be specified as 0.2. However, we have used only four classes. In order to more reliable system, the value should be optimized by using more than four categories. Finally, the candidates that have a poor score value and are less frequent will be eliminated.

4 Implementation and Results

4.1 Language Resources

In our experiments, we used the BOUN Web corpus and language resource prepared [1]. They propose a set of language resources for Turkish language processing applications. They presents an implementation of a morphological parser based on two-level morphology, an averaged perceptron-based morphological disambiguator with accuracy of %98, and a web corpus.

The BOUN Web corpus contains four sub-corpora. Three of them named NewsCor are from three major Turkish news portals and the other corpus named GenCor is a general sampling of web pages in the Turkish Language. For encoding of the xml files, XML Corpus Encoding Standard is used, XCES. The corpus is tokenized and encoded in paragraph and sentence levels. And other symbols are also tagged. The corpus is about the size of 490M tokens.

4.2 Experiments

We conducted several experiments on unparsed corpora, and parsed it with the parser as described in previous section. Each word in the raw text is converted into the form of surface value /root /pos tag.

Algorithm 1. Dimension

NAME: getDIM
 INPUT: C, H
 OUTPUT: DIM
 PURPOSE: For a given hypernym list, it outputs appropriate dimension of Word Space Model

```

for each h in H
  list-bigram=retrieve bigram information for h in C
  ranked=rank list-bigram according to chi-square coefficient
  balanced-list=balanced-pos(ranked,20)
  return balanced-list

```

Fig. 1. Create Dimension for a given hypernym list. C= Corpus, H=Hypernym List, DIM= Dimension

We selected four hypernyms for the test phase; fruit, country, vegetable and fish. For a given hypernym, the algorithm searches the parsed corpora to match the pattern using regular expression. In order to calculate unigram, bigram information and statistical values, TextNSP [43] library is used. For other calculations such as the pattern extraction process and second order representation, the necessary algorithms are implemented in the Java and Python programming languages. The steps of the algorithm are as follows:

Algorithm 2. Producing Hyponyms

NAME: ProduceHyponym
 INPUT: C, H, P
 OUTPUT: Hyponym List for each h in H
 PURPOSE: For a given hypernym list, it decides the hyponyms

```

for each h in H
  chl-initial= apply-pattern(h,P,C)
  chl-elim= applying-eliminatio-criteria(chl-initial)
  chl-by-word-matrix= creatWordSpace(chl-elim,getDIM(h))
  chl-by-chl-matrix= cosine-similarity(chl-by-word-matrix)
  final-hyponym-list= eliminate(chl-by-chl, threshold)
  return final-hyponym-list

```

Fig. 2. Create Hyponym List for each Hyponym in H. C= Corpus, H=Hypernym List, P= Pattern, chl= Candidate Hyponym List

The algorithm is simply summarized in Figure 1 and Figure 2. The output of the first algorithm is used in the second algorithm. The second algorithm produces hyponym candidates for each hypernym in a given list.

5 Results and Evaluation

In order to compare results, we used two different sources. Firstly, we used the BOUN Web Corpus and secondly, we searched the web and manually found the list of a given target hypernym. Our examples were fruit, vegetable, fish and country. Once we have checked whether an item in the list appears in our corpus or not, we compared the results in a more realistic environment by checking against the web list. Because of corpus-driven approach, we especially checked our resulting hypernyms against the list in the corpus, since the approach mines only the corpus. Table 1 shows what is the possible size of the hyponym list in web and corpus as well. And it also shows us the number of retrieved items for each steps.

Table 1. The number of items in web, corpus and in the output of the algorithm

Class	Web	Corpus	S1	S2	S3
Fruit	40	32	69	42	31
Vegetable	46	41	86	53	47
Country	189	137	1525	560	172
Fish	69	41	55	35	32

Looking at the first row, 40 items are retrieved from web, 32 occurred in the corpus. There are 69, 42 and 31 items proposed by the Step1, Step2 and Step3 respectively. The precision/recall based evaluation method can be used to analyze the results. Table 2 shows precision and recall values for each hypernym.

Two different recall values are evaluated. $Recall_1$ value shows the number of successfully retrieved items divided by the number of items existing in the web. Similarly, to calculate the $Recall_2$, we divide the number of successfully retrieved items by the number of items in the corpus. Our algorithm uses only corpus data. The success is measured by looking at the $Recall_2$. On the other hand, $Recall_2$ represents capacity of the model and its performance ratio against the corpus.

As shown in Table 2, a significant increasing in precision values is produced by applying elimination at each step. For instance, the list in Step1 for Fruit hypernym includes some incorrect relevant items such as vitamin/vitamin or porsiyon/portion. In next step, portion remains but vitamin is eliminated.

Finally, portion is eliminated in Step3. The second important result is that the decrease in recall is very small. Such a decrease is inevitable because we apply statistical elimination rather than statistical expansion.

During the lexico-syntactic pattern phase, we observed that the more frequent items tend to be correct hyponyms. Therefore we apply a rule retrieving the candidates that occur at least 4 times. Lexico-syntactic patterns can be risky since incorrect hyponyms are often retrieved. But such incorrect candidates are mostly less frequent. So the challenge at this step is the elimination. The semantic similarity measurement is the main solution for the elimination problem.

Table 2. Precision and Recall values for Fruit, Vegetable, Country and Fish

Hypernym	Recall-Prec.	S1	S2	S3
Fruit	<i>Recall₁</i>	0.69	0.66	0.58
	<i>Recall₂</i>	0.78	0.75	0.65
	<i>Precision</i>	0.44	0.71	0.84
Vegetable	<i>Recall₁</i>	0.88	0.76	0.76
	<i>Recall₂</i>	0.91	0.79	0.79
	<i>Precision</i>	0.52	0.70	0.79
Country	<i>Recall₁</i>	0.72	0.70	0.61
	<i>Recall₂</i>	0.96	0.95	0.83
	<i>Precision</i>	0.09	0.25	0.71
Fish	<i>Recall₁</i>	0.42	0.40	0.37
	<i>Recall₂</i>	0.61	0.58	0.54
	<i>Precision</i>	0.66	0.94	0.97
Average	<i>Recall₁</i>	0.68	0.63	0.58
	<i>Recall₂</i>	0.81	0.77	0.70
	<i>Precision</i>	0.69	0.65	0.83

Taking the result into account, the following are our observations and findings;

1. The more frequent items in the patterns tend to be correct hyponyms. Less-frequent candidates in the patterns can easily be eliminated by making some simple assumptions. Similarity measurement is an efficient way to select correct hyponyms.
2. Corpus-based studies and algorithms suffer from data sparseness. For hyponym detection, there is big difference in the fraction of the siblings / hyponyms. For instance, while the hyponym 'apple' appears in the patterns 30 times, only a few kiwi instances are found. To cope with such cases, other similarity measurement such as pmi, jaccard, or dice can be analyzed and compared.
3. To create the dimension of word space, the algorithm can extract many unrelated words causing noise. Therefore, it is an important challenge to select the distinctive words and ignore misleading words. Moreover, the number of pos type (Noun/Verb/Adj) must be balanced as word space.

6 Conclusion

In this study, we presented a method for acquiring hyponyms of a given hypernym. We integrated the method relying on lexico-syntactic pattern and semantic similarity using a word space model. Once the model has applied the former method to extract an initial list and applied the latter function to eliminate the erroneous items in order to increase the precision. We showed that an algorithm based on a particular lexico-syntactic pattern can retrieve a significant number of hyponymy relations with some assumptions that are particularly applicable to agglutinative language such as Turkish. We also demonstrated that this integration could give promising results for Turkish.

After applying pattern matching, the resulting hyponym list can contain many errors or incorrect instances. To eliminate such cases, we applied a model based semantic similarity using second-order word representation and cosine similarity measurement. The objective is to get better relevance and more precise results. Some other studies especially focused on syntactical expansion to retrieve as many hyponyms as possible. This could be among our future work. We observed that semantic similarity measurement in second order word space gives successful results.

We use a corpus size of 490 M words in which it is easy to match the patterns. Here, we succeeded in increasing the precision without decreasing the recall. Looking at the result table for each case, while recall values slightly decrease or remain the same, precision scores get better. After semantic similarity based improvement, the average precision is increased by %92. However, the recall values are decreased by %10. These results indicate that our methodology can robustly capture hyponymy relationships for Turkish.

We conclude that pattern frequency, document frequency and semantic similarity score are the main properties of hyponym candidates. There exist rules that can successfully eliminate unsatisfactory candidates. However, bad design of the word dimension can lead to model failure. Since candidates having low document frequency are a major problem, they must be covered with a different approach. Distinctive and related words must be reserved.

This is the first study on corpus-driven hyponymy acquisition for Turkish Language. Extraction other types of relationships, such as part-of, synonymy and group-of will be our future work. Secondly, since corpus-based approach has some limitations such as sparseness, we plan to use the web as a corpus and to apply statistical expansion to increase both precision and recall.

References

1. Sak, H., Güngör, T., Saraçlar, M.: Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus. In: Nordström, B., Ranta, A. (eds.) *GoTAL 2008*. LNCS (LNAI), vol. 5221, pp. 417–427. Springer, Heidelberg (2008)
2. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: WordNet: An online lexical database. *International Journal of Lexicography* 3, 235–244 (1990)

3. Lenat, D., Prakash, M., Shepherd, M.: CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI Magazine* 6(4), 65–85 (1986)
4. Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM* 38(11), 39–41 (1995)
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
6. Alshawi, H.: Processing Dictionary Definitions with Phrasal Pattern Hierarchies. *American Journal of Computational Linguistics* 13(3-4), 195–202 (1987)
7. Markowitz, J., Ahlswede, T., Evens, M.: Semantically Significant Patterns in Dictionary Definitions. In: *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics, ACL 1986*, vol. 13, pp. 112–119. Association for Computational Linguistics (1986)
8. Jensen, K., Binot, J.: Disambiguating Prepositional Phrase Attachments by Using On-Line Dictionary Definitions. *American Journal of Computational Linguistics* 13(3-4), 251–260 (1987)
9. Nakamura, J., Nagao, M.: Extraction of semantic information from an ordinary English dictionary and its evaluation. In: *Proceedings of the 12th International Conference on Computational Linguistics, COLING 1988*, vol. 2, pp. 459–464. Association for Computational Linguistics (1988)
10. Ahlswede, T., Evens, M.E.: Parsing vs. Text Processing in the Analysis of Dictionary Definitions. In: *Proceedings of the 26th Annual Meeting on Association for Computational Linguistics, ACL 1988*, vol. 1, pp. 217–224. Association for Computational Linguistics (1988)
11. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: *Proceedings of the 14th Conference on Computational Linguistics, COLING 1992*, vol. 2, pp. 539–545. Association for Computational Linguistics (1992)
12. Hearst, M.A.: Automated Discovery of WordNet Relations. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database and Some of its Applications*, pp. 131–152. MIT Press, Cambridge (1998)
13. Riloff, E., Shepherd, J.: A corpus-based approach for building semantic lexicons. In: *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 117–124 (1997)
14. Roark, B., Charniak, E.: Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, ACL 1998*, vol. 2, pp. 1110–1116. Association for Computational Linguistics, Montreal (1998)
15. Caraballo, S.A.: Automatic Construction of a Hypernym-Labeled Noun Hierarchy From Text. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL 1999*, pp. 120–126. Association for Computational Linguistics (1999)
16. Alfonseca, E., Manandhar, S.: Improving an Ontology Refinement Method with Hyponymy Patterns. In: *Proceedings of Language Resources and Evaluation (LREC 2002)*, pp. 235–239 (2001)
17. Etzioni, O., Cafarella, M.J., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in knowitall (preliminary results). In: *Proceedings of the 13th International Conference on World Wide Web, WWW 2004*, pp. 100–110. ACM, New York (2004a)

18. Etzioni, O., Cafarella, M.J., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Methods for domain-independent information extraction from the Web: An experimental comparison. In: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI 2004, pp. 391–398. AAAI Press (2004b)
19. Etzioni, O., Cafarella, M.J., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* 165(1), 91–134 (2005)
20. Ritter, A., Soderl, S., Etzioni, O.: What is this, anyway: Automatic hypernym discovery. In: Proceedings of AAAI 2009 Spring Symposium on Learning, pp. 88–93. AAAI Press (2009)
21. Rydin, S.: Building a Hyponymy Lexicon with Hierarchical Structure. In: Proceedings of the ACL 2002 Workshop on Unsupervised Lexical Acquisition, ULA 2002, pp. 26–33. Association for Computational Linguistics (2002)
22. Cederberg, S., Widdows, D.: Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 111–118. Association for Computational Linguistics (2003)
23. Ando, M., Sekine, S., Ishizaki, S.: Automatic Extraction of Hyponyms from Newspapers Using Lexico-syntactic Patterns. In: Fourth International Conference on Language Resource and Evaluation, LREC 2004, Lisbon, Portugal (2004)
24. Snow, R., Jurafsky, D., Ng, A.Y.: Learning Syntactic Patterns for Automatic Hyponym Discovery. In: Advances in Neural Information Processing Systems, vol. 17. MIT Press, Cambridge (2005)
25. Tjong Kim Sang, E.F., Hofmann, K.: Automatic Extraction of Dutch Hyponym-Hyponym Pairs. In: Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands, LOT, Netherlands Graduate School of Linguistics (2007)
26. Paşca, M.: Acquisition of Categorized Named Entities for Web Search. In: CIKM 2004: Proceedings of The Thirteenth ACM International Conference on Information and Knowledge Management, pp. 137–145. ACM Press, New York (2004)
27. Tjong Kim Sang, E.F.: Extracting Hyponym Pairs from the Web. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 165–168. Association for Computational Linguistics, Prague (2007)
28. Kozareva, Z., Riloff, E., Hovy, E.: Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In: Proceeding of ACL 2008, pp. 1048–1056. The Association for Computational Linguistics (2008)
29. Elghamry, K.: Using the Web in Building a Corpus-Based Hyponymy-Hyponymy Lexicon with Hierarchical Structure for Arabic. In: The Sixth International Conference on Informatics and Systems, INFOS 2008, Cairo, Egypt (2008)
30. Sombatsrisomboon, R., Matsuo, Y., Ishizuka, M.: Acquisition of Hyponyms and Hyponyms from the WWW. In: Proceedings of the 2nd International Workshop on Active Mining, Japan (2003)
31. Chodorow, M.S., Byrd, R.J., Heidorn, G.E.: Extracting Semantic Hierarchies from a Large On-Line Dictionary. In: Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics, ACL 1985, pp. 299–304. Association for Computational Linguistics (1985)
32. Widdows, D., Dorow, B.: A Graph Model for Unsupervised Lexical Acquisition. In: Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002, vol. 1, pp. 1093–1099. Association for Computational Linguistics (2002)

33. Sumida, A., Kentaro, T.: Hacking wikipedia for hyponymy relation acquisition. In: Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP), pp. 883–888. Association for Computational Linguistics (2008)
34. Shinzato, K., Torisawa, K.: Acquiring hyponymy relations from web documents. In: Proceedings of HLT-NAACL, vol. 80, pp. 73–80 (2004)
35. Imsombut, A., Kawtrakul, A.: Automatic building of an ontology on the basis of text corpora in Thai. *Language Resources and Evaluation* 42(2), 137–149 (2008)
36. Bilgin, O., Çetinoğlu, Ö., Oflazer, K.: Building a wordnet for Turkish. *Romanian Journal of Information Science and Technology* 7(1-2), 163–172 (2004)
37. Amasyalı, M.F.: Türkçe Wordnet'in Otomatik Olarak Oluşturulması. In: SIU 2005, Kayseri (2005)
38. Yazıcı, E., Amasyalı, M.F.: Automatic Extraction of Semantic Relationships Using Turkish Dictionary Definitions. *EMO Bilimsel Dergi* 1(1), 1–13 (2011)
39. Güngör, O., Güngör, T.: Türkçe Bir Sözlükteki Tanımlardan Kavramlar Arasındaki Üst-kavram İlişkilerinin Çıkarılması. *Akademik Bilişim Konferansı 2007* 1(1), 1–13 (2007)
40. İberbetçi, A., Orhan, Z., Pehlivan, İ.: Extraction of Semantic Word Relations in Turkish from Dictionary Definitions. In: Proceedings of the ACL 2011 Workshop on Relational Models of Semantics (RELMS 2011), pp. 11–18. Association for Computational Linguistics, Portland (2011)
41. Orhan, Z., Pehlivan, İ., Uslan, V., Önder, P.: Automated Extraction of Semantic Word Relations in Turkish Lexicon. *Mathematical and Computational Applications* 16(1), 13–22 (2011)
42. Schütze, H.: Automatic Word Sense Discrimination. *Computational Linguistics - Special Issue on Word Sense Disambiguation* 24(1), 97–123 (1998)
43. Pedersen, T., Banerjee, S., Kohli, S., Joshi, M., McInnes, B.T., Liu, Y.: The Ngram Statistics Package (Text::NSP) - A Flexible Tool for Identifying Ngrams, Collocations, and Word Associations. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011), pp. 131–133. Association for Computational Linguistics, Portland (2011)

Automatic Taxonomy Extraction in Different Languages Using Wikipedia and Minimal Language-Specific Information

Renato Domínguez García, Sebastian Schmidt,
Christoph Rensing, and Ralf Steinmetz

Multimedia Communications Lab - Technische Universität Darmstadt
64283 Darmstadt - Germany

{renato.dominguez.garcia,sebastian.schmidt,christoph.rensing,
ralf.steinmetz}@kom.tu-darmstadt.de

<http://www.kom.tu-darmstadt.de>

Abstract. Knowledge bases extracted from Wikipedia are particularly useful for various NLP and Semantic Web applications due to their coverage, actuality and multilingualism. This has led to many approaches for automatic knowledge base extraction from Wikipedia. Most of these approaches rely on the English Wikipedia as it is the largest Wikipedia version. However, each Wikipedia version contains socio-cultural knowledge, i.e. knowledge with relevance for a specific culture or language. In this work, we describe a method for extracting a large set of hyponymy relations from the Wikipedia category system that can be used to acquire taxonomies in multiple languages. More specifically, we describe a set of 20 features that can be used for Hyponymy Detection without using additional language-specific corpora. Finally, we evaluate our approach on Wikipedia in five different languages and compare the results with the WordNet taxonomy and a multilingual approach based on interwiki links of the Wikipedia.

Keywords: Hyponymy Detection, Multilingual large-scale taxonomies, Wikipedia Mining, NLP.

1 Introduction

Natural language processing (NLP) covers all steps of processing natural language from the syntactical representation (or audio representation) to the discourse. While the first steps aim at breaking down and analyzing the structure of the text, the latter steps cope with/handle reassembling and understanding. All those steps require human knowledge in machine processable form to be executable. Whereas the knowledge required for the first steps is of very local scope, which means the processing of the single tokens is only minimally dependent on neighbouring tokens and only requires a small number of rules, subsequent steps require more and more context. This holds for both, the context of the text in

the document itself and for the general knowledge required to derive the understanding of the text. For the latter, a structured knowledge base on the specific topic enables a machine to derive knowledge and put it into an abstract context. One of those knowledge bases is a taxonomy. Within a taxonomy, relations of the type *is-a* are contained, creating a tree-like structure of real world concepts. One example of such a *is-a* relation is the tuple (*juice*, *beverage*) because juice is a beverage.

In some fields of knowledge, like biology, elaborated taxonomies already exist. But there are still many domains without such explicit taxonomies. Additionally, these taxonomies are usually only defined in one language. Therefore, although a variety of taxonomies are existing in English, other languages lack these. This impedes the application of taxonomies in several languages and fields of knowledge impossible. Within this paper we present an approach to create taxonomies in different languages automatically from the category and article structure of Wikipedia. Our approach uses structural properties of Wikipedia and syntactical structure of single categories and articles, and requires only minimal language-specific information.

After giving an overview of existing work on automatic taxonomy creation (section 2), we will present our machine learning approach in section 3. Its evaluation is shown in chapter 4 and concluded in the last section.

2 Related Work

As one of the key challenges for NLP applications is to allow the extraction of machine-usable knowledge from written language, a lot of research on the topic has been conducted. In this section, we will give a short overview on approaches using for this purpose. We focus on approaches based on Wikipedia and exclude approaches based on Text Mining as these approaches rely on lexical patterns. Lexical patterns can be applied on texts in different languages in order to obtain taxonomies from scratch. but they are strongly language-dependent.

WordNet [8] is a knowledge base consisting of English words with short definitions and both lexical and semantic relations between those words. Semantic relations comprise hypernymy, hyponymy and synonymy among others. This enables the direct extraction of a taxonomy from WordNet. Within the Universal WordNetProject [2] based on wordnets in different languages and other information sources first an initial graph was built which is afterwards enriched by adding missing links and then iteratively refined by making use of machine learning techniques. The result is a multilingual lexical database of terms in combination with their meanings, containing relations between the terms. However, WordNet is a manually built resource and has to be catered for by linguistic experts. Thus, its growth is slow and novel, domain-specific or trending topics are usually not covered. The same holds for other manually created knowledge bases. Therefore, in scenarios that depend on the availability of a domain-generic set of concepts and up-to-date knowledge, these approaches do not suffice.

Sumida and Torisawa [16] make use of the structure of Wikipedia articles to extract *hyponymy* relations. Often, Wikipedia articles are structured in a way, that subsections in Wikipedia articles describe a *hyponym* of the wrapping section (e.g. the article *Sense* has the section *Senses* with the subsections *Sight*, *Taste* etc.). Sumida and Torisawa use this for discovering *hyponymy* relations by applying language dependent pattern matching and use of machine learning techniques to differentiate between *hyponymy* and *non-hyponymy* relations.

Ponzetto und Strube [13] use of this structure and aim to identify *hypernymy* and *hyponymy* between Wikipedia categories. Wikipedia categories build a large network containing links of different types. In many cases there is a subtype relation between two categories (e.g. the category *Juice* is connected with the category *Non-alcoholic beverages*), but in general it can be any kind of semantic relation (e.g. the category *Titles* is connected with *Sociolinguistics*). Therefore, they identify that can disambiguate taxonomic relations from others. For applying the approach to other languages the algorithm itself has to be modified because most of the features are strongly dependent on the used language. Further, the Tipster corpus [6] which is used for one step of *hyponymy* relation detection is not on hand in other languages than English. Kassner et al. [7] adapt the approach to be used with the German Wikipedia. In addition to the smaller size of the German Wikipedia they had to face the challenge of a more complicate word composition in German compared to English.

Navigli and Ponzetto present BabelNet [11], a multilingual semantic network created by the aggregation of WordNet, Wikipedia and SemCor. Additionally to those resources, the Google Translator API¹ is used to translate article names of the Wikipedia which do not have a correspondent in other languages. We see three problems with this approach. First, it relies on Statistical Machine Translation and its ability to translate to other languages. Second, the Google Translator has strong usage restrictions, which makes it not suitable for a publicly available resource. Finally, BabelNet use *interwiki links*² to build the multilingual semantic network. However, *interwiki links* do not exist for many articles in Wikipedia.

Another approach currently developed for creating a taxonomy from the English Wikipedia is WikiNet [10]. By analyzing categories and articles of the English Wikipedia a monolingual concept network is created. Afterwards, for all included concepts the *interwiki links* are examined and a multilingual concept network is created by adding all articles being interlinked by those links. The authors describe the portability of this approach to other languages. However, the impact of combining category systems in different languages is not clear.

A common disadvantage of the previously presented approaches is the loss of socio-cultural knowledge which is available in Wikipedia. Some artifacts of knowledge are only relevant for a single region with a single spoken language. Those artifacts are often only covered in the Wikipedia of the respective

¹ <http://code.google.com/apis/language/> - retrieved 28.10.2011

² Links from a Wikipedia article in one language to an article in another language describing a similar concept.

language. When creating taxonomies only based on the English Wikipedia, all socio-cultural knowledge described in other Wikipedia versions can not be transferred into the taxonomy.

MENTA [3] addresses this issue by providing a multilingual taxonomy consisting of entities in various languages. To this end, all similar entities from different Wikipedia versions are merged and afterwards, both syntactical and structural properties of Wikipedia and WordNet are used to determine taxonomic relations between the entities. This approach is not fully automatic but some linguistic exceptions for syntactical rules need to be specified manually.

As presented, there are several approaches to create taxonomies in different languages (semi-)automatically. They all show to have advantages, but none of them is at the same time easily adaptable to different languages, accurate and dynamic in terms of trending topics. With our approach, we aim at targeting those challenges.

3 Language-Independent Acquisition of Hierarchical Relations

Two preliminary studies are the basis of our approach: In the first study [4] we analyzed the feasibility of Hyponymy detection in different languages using simple heuristics and in the second study [5], we described our application scenario for taxonomies in different languages and performed first experiments with a machine learning approach. In the following subsections we present our approach which involves of a set of 20 features to recognize *is-a* relations from Wikipedia categories. These features are described in this section.

3.1 The Feature Set for Recognizing *is-a* relations

We take pairs of categories (in following denoted as c_1 and c_2) from the Wikipedia category graph and apply our features to each of these pairs, called *links*. The returned values are used to build the *feature vector*, which is used by a classifier to determine if there is an *is-a* relation between both categories. In table II, we present an overview of the used features. In the following, these features are described in detail.

Preprocessing Features. These features are used to detect links containing nodes to administrative or refinement categories and evaluate to true, if the category belongs to one of those. Administration categories contain prefixes like `Wikipedia` or `User`. Refinement links are used in Wikipedia to organize multiple categories using the pattern `X by Y` (e.g. "Companies by country"). Their purpose is to structure and simplify the category graph. The prefixes of administration categories and the preposition used in the refinement links are used in all Wikipedia versions independently of the language. However, they have to be adapted to the respective language. For instance, the prefix `category` is translated to `Kategorie` in German and `Categoría` in Spanish.

Table 1. Overview of the used features

	Name	Value type	Feature type
1	adminCatFeature	binary	preprocessing
2	refinementLinkFeature	binary	preprocessing
3	positionOfHeadFeature	{2, 1, 0, -1}	syntactic
4	cooccurrenceOfWordsFeature	N	syntactic
5	cooccurrenceArticleFeature	binary	structural
6	commonArticleFeature	{1, 0, -1}	structural
7	c1c2IncomingLinksFeature	{1, 0, -1}	structural
8	c1c2OutgoingLinksLinksFeature	{1, 0, -1}	structural
9	c1distanceCommonAncestorFeature	N	structural
10	c2distanceToCommonAncestorFeature	N	structural
11	c1NumberOfSubcategoriesFeature	N	structural
12	c1NumberOfSuperCategoriesFeature	N	structural
13	c2NumberOfSubcategoriesFeature	N	structural
14	c2NumberOfSuperCategoriesFeature	N	structural
15	CommonWikilinksFeature	N	structural
16	firstSentenceFeature	binary	article
17	RedirectFeature	{1, 0, -1}	article
18	c2Incl1Feature	N	article
19	c1ArticleFeature	binary	article
20	c2ArticleFeature	binary	article

Syntactic Features. Syntactic features use string matching of syntactic components to differentiate between *is-a* and *not-is-a* links. We distinguish between two different syntactic features. The `positionOfHeadFeature` uses the fact that the lexical head of two category names is a very effective method for labeling *is-a* links [13]. This feature returns a value $[-1, 2]$ for pairs of categories c_1 and c_2 representing the position of the lexical head of the superordinate category. We differentiate between the following cases:

$$f_3(c_1, c_2) = \begin{cases} 2 & \text{if lexical head of } c_2 \text{ is at the end of } c_1 \\ 1 & \text{if lexical head of } c_2 \text{ is in the middle of } c_1 \\ 0 & \text{if lexical head of } c_2 \text{ is at the beginning of } c_1 \\ -1 & \text{else, i.e. no occurrence} \end{cases}$$

For instance, the value of this feature for $c_1 = \text{"French Revolution"}$ and $c_2 = \text{"Revolution"}$ is 2. In the English Wikipedia, the lexical head of a category is usually the last word. However, there are some exceptions, for example for categories containing prepositions e.g. "Campaign for nuclear disarmament" or containing refinement brackets e.g. "Sport (Ireland)". We cope with this issue by recognizing prepositions heuristically, i.e. matching preposition and using the term before the preposition. This heuristic works for other languages as well: In Arabic, for instance, where the lexical head is at the beginning of a category name or in German, where the lexical head is "hidden" inside noun compounds where the multiple noun modifies the meaning given by the last one, e.g. "Baumhaus" (Eng. tree house). If the position of the head is not the head position in a given language then we assume that there is a *not-is-a* relation between c_1 and c_2 . `cooccurrenceOfWords` represents cooccurrences of words in both category names. This feature should match cases, in which two category labels have more words in common than the lexical head.

Structural Features. These features exploit the structure of the category graph and the wikilink graph³. `cooccurrenceFeature` returns `true` for pairs of categories which have at least one article in common [13]. Further, `commonArticle-Feature` returns the number of articles in common between both categories. `c1c2IncomingLinksFeature` and `c1c2OutgoingLinksFeature` measure the strength of the relation between both categories. For this purpose, the number of articles in c_1 is counted, which have at least one incoming or outgoing wikilink to any article in c_2 [1]. The features `c1distanceCommonAncestorFeature` and `c2distanceCommonAncestorFeature` calculate the distance between of given categories c_1 and c_2 to the first common ancestor c_A of both categories. Both distances are calculated separately, i.e. `c1distanceCommonAncestorFeature` calculates the distance of c_1 to c_A and feature `c2distanceCommonAncestorFeature` the distance of c_2 to c_A . If $c_i = c_A$ then the distance for c_i is 0. `c1NumberOfSubcategories`, `c1NumberOfSupercategories`, `c2NumberOfSubcategories` and `c2NumberOfSupercategories` just counts the number of sub- and supercategories of c_1 and c_2 . Categories having a huge number of subcategories usually represent more abstract concepts which can be referenced by many other categories, e.g. "Science". Finally, `CommonWikilinksFeature` counts the number of common wikilinks between c_1 and c_2 .

Article Features. This set of features is applied to the content of articles. The first sentence of an article has a special meaning for taxonomic applications as it usually contains a definition of the concept [12]. This fact is used by `definitionSentenceFeature` to recognize *is-a* relations in the first sentence. This means that if an article a belongs to a category c_1 with both having the same label, then we search for occurrences of lexical heads of c_2 in a in the first sentence of the article. For instance, if $c_1 = \text{"Mice"}$ and $c_2 = \text{"Pet Rodent"}$, we test if the first sentence of the article "Mice" contains the term "rodent". If the check is positive, then this feature returns `true`. An advantage of this method is that a language-dependent search of patterns is not needed, and thus it can be applied to different languages. Further, the feature `c2Inc1Feature` counts the number of occurrences of the lexical head of c_2 in the rest of the article of c_1 . `c1ArticleFeature` and `c2ArticleFeature` match c_1 and c_2 to Wikipedia articles. If c_2 can be matched to an article then we assume that this category is an existing concept, otherwise it may be a category used to structure the category graph like e.g. lists. In this case `true` is returned. Finally, the feature `RedirectFeature` returns `true` if the article corresponding to c_1 is redirected in Wikipedia to the article corresponding to c_2 . This represents a strong relation between both categories. This information is stored in Wikipedia in so called *redirect pages*.

³ This graph is built by iterating over the Wikipedia articles and adding all links between two articles in the same language version of the Wikipedia.

3.2 Language-Specific Information Needed by Our Approach

This approach is applicable as such to very different languages without modifying the features with minimal language-specific information. Specifically, it can be used to derive taxonomies from different languages with little information about a language needed. The mandatory data is the following:

1. A list of prefixes of meta-categories that Wikipedia uses in this language, e.g. `wikipedia`, `user` or `articles`.
2. The preposition contained in refinement-links, e.g. `by` in English or `nach` in German.
3. A list of prepositions of a language in order to match the lexical head of categories containing prepositions heuristically, e.g. "Battalions of the Canadian Expeditionary Force".

The language-independency of our approach is restricted by the 281 existing Wikipedia versions (i.e. we can not acquire taxonomies from other languages) and by the input of the prefixes and the prepositions mentioned before. Using only this information it is possible to generate taxonomies in different Wikipedia languages as we show in the next section.

4 Evaluation

We evaluated our approach in four different languages: three European languages (English, German and Spanish) and two language with non-latin characters (Arabic and Russian). We used a manually labelled corpus for each language to obtain results by applying our approach for multiple languages.

4.1 Building the Different Corpora

Our corpus consists of 1000 randomly selected Wikipedia articles and categories. We extracted the corpus using Wikipedia's export page⁴ and following method:

1. Get random article a_i using the "Random page"-link⁵ and add all links of a to all its categories in our corpus.
2. Choose a random category c of a and add all links of c to all its super categories $c_{s,i}$ in our corpus. As our corpus should contain 1000 articles, we filter out categories that have more than 100 super categories in order to have enough articles and categories from different domains.
3. Choose randomly a super category $c_{s,j}$ of $c_{s,i}$ and all links of $c_{s,j}$ and insert it into our corpus.
4. Repeat step 3. until the root category⁶ or an already visited category is reached moving to the top of the category graph.
5. Go to step 1, until corpus has 1000 articles.

⁴ <http://en.wikipedia.org/wiki/Special:Export> - retrieved 28.10.2011.

⁵ <http://en.wikipedia.org/wiki/Special:Random> - retrieved 28.10.2011.

⁶ <http://en.wikipedia.org/wiki/Category:Contents> - retrieved 28.10.2011.

After we built the corpus, we labelled it manually with the relevant relations (*is-a* and *not-is-a*). In Table 2 we summarize the size and distribution between *is-a* and *not-is-a* links in the different corpora.

Table 2. Summarized statistics of the different corpora

Language	English	Spanish	German	Arabic	Russian
Number of <i>is-a</i> links	1297 (29.9 %)	786 (36.1 %)	808 (33.6 %)	1135 (41.4 %)	2545 (40.4 %)
Number of <i>not-is-a</i> links	3048 (70.1 %)	1388 (63.9 %)	1597 (66.4 %)	1604 (58.6 %)	3752 (59.6 %)
Number of labeled links	4345	2174	2405	2739	6297

4.2 Evaluation Results

In this section, we present the results of our evaluation. We used the Weka Machine Learning Toolkit [17] and chose J48 decision trees (Weka implementation of C 4.5 [14]) as a classifier. Decision trees are a commonly used classifier as they are fast to train and in classifying instances, their rules are simple to understand and they can be combined with other decision techniques in order to improve results. All classification results were subjected to ten-fold cross validation. Table 3 gives an overview of our results. It shows correctly and incorrectly classified instances. On average, 83.1 % of the links are labelled correctly and 16.9 % are labelled incorrectly.

Table 3. Summarized results of our approach by languages

Language	English	Spanish	German	Arabic	Russian
Correctly classified inst.	3583 (82.6 %)	1838 (84.6 %)	1963 (81.6 %)	2283 (83.4 %)	5067 (80.5 %)
Incorrectly classified inst.	753 (17.4 %)	336 (15.4 %)	442 (18.4 %)	456 (16.6 %)	1230 (19.5 %)
Total number of instances	4345	2174	2405	2739	6297

Table 4 summarizes the most common metrics of evaluation of categorization algorithms: Precision, Recall and F-Measure. For Precision, we obtained average results of 74.5 % and Recall was 77.2 % for *is-a* relations and for *not-is-a* relations Precision was 87 % and Recall 86.3 %. The English confusion matrix is additionally shown in Table 5.

Table 5 shows that the major source of misclassification is incorrectly classified *is-a* links. The reason is that a high number of *is-a* instances could not be recognized as *is-a* by single features, but by the combination of multiple feature values. Thus, these combinations can recognize more instances than simple heuristics, but they introduce some misclassified instances as they do not have a precision of 100 %. One possibility to improve the results presented here is to use cross-language links to integrate the results from different languages. Further, we evaluated the effect of each type of features measured in Accuracy. We can see in Table 6 that single feature classes in most of the cases do not perform better than 70 %.

Table 4. Detailed Accuracy by class and language

	Precision	Recall	F-Measure	Class
English	70.0 %	73.0 %	71.2 %	is-a
	88.3 %	86.7 %	87.5 %	not-is-a
Spanish	76.3 %	83.1 %	79.5 %	is-a
	89.9 %	85.4 %	87.6 %	not-is-a
German	71.9 %	74.4 %	73.1 %	is-a
	86.8 %	85.3 %	86.0 %	not-is-a
Arabic	79.7 %	80.4 %	80.0 %	is-a
	86.0 %	85.5 %	85.7 %	not-is-a
Russian	73.2 %	81.5 %	77.1 %	is-a
	86.4 %	79.8 %	83.0 %	not-is-a

Table 5. Confusion matrix English

a	b	← classified as
944	349	a = is-a
404	2639	b = not-is-a

Table 6. Effect of feature classes by languages

	Preprocessing features	Syntactic features	Structural features	Article features
English	70.2 %	69.8 %	75.8 %	71.4
Spanish	63.8 %	69.9 %	73.6 %	70.5 %
German	66.4 %	68.1 %	74.1 %	67.2
Arabic	64.2 %	59.8 %	77.8 %	73.5
Russian	66.2 %	64.6 %	69.1 %	66.8

Finally, we rank all features by information gain [9], measuring how well a given feature separates the training instances according to their target classification. This is shown in Table 7. Features recognizing *not-is-a* links are ranked higher as the number of *not-is-a* links is higher than *is-a* links, i.e. they are used to categorize more links. In general, we can see that syntactic and structural features performed best. We observe, that the structural features are better for detecting *not-is-a* links and the syntactic features for *is-a* links.

Furthermore, we observed that the features `distanceC1ToCommonAncestor` and `distanceC2ToCommonAncestor` were not very distinctive. This can be explained by the method used to build our corpus. In our corpus, we collected only direct links between categories and articles. However, we believe that these features could be helpful in other scenarios. For instance, to train a classifier which does not only recognize direct links, but also indirect links, i.e. transitive links. Such a classifier could be used to recognize *is-a* relations independently of our taxonomy scenario. This is going to be part of future work.

4.3 Evaluation of Our Approach Using Existing Knowledge Bases

It is crucial to evaluate the obtained hierarchical relations in comparison with similar approaches. In this section, we compare the Accuracy of our approach against WordNet [8] and WikiNet [10] for English. WordNet is one of the most popular knowledge bases in English and WikiNet is a knowledge base which was acquired using interwiki links without additional external corpora.

Table 7. Ranking of the used features by languages

	English features	Spanish features	German features	Arabic features	Russian features
1.	c2InclFeature	c1c2IncomingLinks	c1c2IncomingLinks	c2InclFeature	PositionOfHead
2.	refinementLink	c2InclFeature	refinementLink	c1Article	c1c2IncomingLinks
3.	c2Article	PositionOfHead	c1Article	commonArticleFeature	c2InclFeature
4.	c1c2IncomingLinks	c2Article	c2Article	firstSentence	refinementLink
5.	c1c2OutgoingLinks	refinementLink	PositionOfHead	refinementLink	firstSentence

First, we select those pairs of categories that overlap with WordNet and WikiNet. For each category pair, both categories have to be mapped to WordNet synsets and to WikiNet concepts. Our evaluation consists of a set of 15,483 instances which belong to the Wikipedia category graph, WordNet and WikiNet.

These pairs are evaluated by querying WordNet whether the concept denoted by the Wikipedia subcategory (c_1) is an instance or a subclass of the concept denoted by its category (c_2). The WordNet pairs c_1 and c_2 are looked up in direct relation as well as in indirect relation (i.e. c_1 *is-a* ... *is-a* c_2). We then take the result of the query as the actual (*is-a* or *not-is-a*) semantic relation for the category pair and use it to evaluate the results of our approach. The same procedure is done to evaluate the quality of WikiNet on this dataset. This way we are able to compute standard measures of Precision, Recall and F-Measure of our approach and WikiNet and compare the values, i.e. the information contained in WordNet is used a gold Standard.

Table 8 gives an overview of the results. It shows correctly classified instances by our approach and by WikiNet. 85.95 % of the labelled links were labelled by our approach correctly and 78.23 % by WikiNet. Table 9 shows detailed results

Table 8. Results of our approach and WikiNet compared with WordNet

	Our approach	WikiNet
Correctly classified	13307 (85.95%)	12113 (78.23%)
Incorrectly classified	2176 (14.05%)	3370 (21.77%)
Total number of inst.	15483	15483

of both approaches in our evaluation corpus. For F-Measure, we obtained results of 80.48 % for *is-a* relations and 89.02 % for *not-is-a* relations. These results were significantly better than the results provided by WikiNet.

Table 9. Detailed results of our approach and WikiNet compared with WordNet

	Precision	Recall	F-Measure	Class
Our approach	86.12 %	73.54 %	80.48 %	<i>is-a</i>
	85.86 %	92.42 %	89.02 %	<i>not-is-a</i>
WikiNet	69.14 %	78.29 %	73.37 %	<i>is-a</i>
	85.20 %	78.29 %	81.60 %	<i>not-is-a</i>

These results suggest that our approach performs better than approaches based on wikilinks for the English Wikipedia version. However, only 15 % of the whole instances could be evaluated using WordNet. All in all, we could perform an evaluation for 15,483 instances, but 85,938 instances remain unevaluated. There are two reasons for this: first, Wikipedia has a much larger coverage than WordNet and second, many categories in Wikipedia are semi-phrases (e.g. "People in fiction") that cannot be mapped to proper WordNet synsets.

In order to further evaluate the quality of our approach, we performed additional experiments, for example, applying our features not only to links between categories, but also to links between articles and categories. These results are however not presented in this paper, as they simply confirm the results already presented here.

5 Conclusion

Taxonomies are very useful for many NLP applications. However, automatic derivation of taxonomies relies in the most of cases on language-dependent methods or it is based on existing manually created knowledge bases like WordNet or GermaNet. In this work, we used the preliminary results of previous studies to develop a set of features and to train a binary classifier to automatically recognize taxonomic relations between pairs of Wikipedia categories extracted from the category graph.

We describe a robust language-independent Wikipedia-based approach which does not depend on further external sources of knowledge. Eventually, we evaluate the proposed features by measuring the accuracy of the classification of instances for each language. We compare the results with WordNet as ground truth and WikiNet as an approach using no additional external corpora other than Wikipedia. In future work, we plan the evaluation of our approach against approaches relying on additional external corpora other than Wikipedia like YAGO [15]. However, we expect that such approaches perform better than generic approaches as they are optimized with language-specific methods to work in one specific language, e.g. English.

Generally, our approach enables us to automatically derive a taxonomy from Wikipedia for different languages using syntactical and structural features and reducing the dependency on third parties. Further, the research presented here provides a foundation on which further applications and research (e.g. in the field of Wikipedia Mining or attaching semantics to web resources) can be based. This approach can also be used to automatically large-scale evaluation of knowledge bases in languages where manually created knowledge bases do not already exist.

Acknowledgments. This work was supported by funds from the German Federal Ministry of Education and Research under the mark 01 PF 08015 A and from the European Social Fund of the European Union (ESF). The responsibility for the contents of this publication lies with the authors.

References

1. Chernov, S., Iofciu, T., Nejd, W., Zhou, X.: Extracting Semantics Relationships between Wikipedia Categories. In: Völkel, M., Schaffert, S. (eds.) Proceedings of the First Workshop on Semantic Wikis – From Wiki To Semantics, Workshop on Semantic Wikis, ESWC 2006 (June 2006)
2. de Melo, G., Weikum, G.: Towards a universal wordnet by learning from combined evidence. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, New York, NY, USA, pp. 513–522 (2009)
3. de Melo, G., Weikum, G.: MENTA: Inducing Multilingual Taxonomies from Wikipedia. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 1099–1108 (2010)
4. Domínguez García, R., Rensing, C., Steinmetz, R.: Automatic acquisition of taxonomies in different languages from multiple wikipedia versions. In: To Be Published in Proceedings of the 10th International Conference on Web-Based Learning (ICWL 2011) (2011)
5. Domínguez García, R., Scholl, P., Steinmetz, R.: Supporting resource-based learning on the web using automatically extracted large-scale taxonomies from multiple wikipedia versions. In: Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies, p. 35. ACM (2011)
6. Harman, D., Liberman, M.: Tipster complete. In: Corpus number LDC93T3A, Linguistic Data Consortium, Philadelphia (1993)
7. Kassner, L., Nastase, V., Strube, M.: Acquiring a Taxonomy from the German Wikipedia. In: Proceedings of the International Conference on Language Resources and Evaluation. European Language Resources Association (2008)
8. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38, 39–41 (1995)
9. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
10. Nastase, V., Strube, M., Boerschinger, B., Zirn, C., Elghafari, A.: WikiNet: A Very Large Scale Multi-Lingual Concept Network. In: Proceedings of the International Conference on Language Resources and Evaluation (2010)
11. 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 (2010)
12. Nguyen, D.P.T., Matsuo, Y., Ishizuka, M.: Subtree Mining for Relation Extraction from Wikipedia. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pp. 125–128 (2007)
13. Ponzetto, S.P., Strube, M.: Deriving a Large-Scale Taxonomy from Wikipedia. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, pp. 1440–1445. AAAI Press (2007)
14. Quinlan, J.R.: C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning), 1st edn. Morgan Kaufmann (January 1993)
15. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th International World Wide Web Conference (WWW 2007). ACM Press, New York (2007)
16. Sumida, A., Torisawa, K.: Hacking Wikipedia for Hyponymy Relation Acquisition. In: Proceedings of the International Joint Conference on Natural Language Processing (2008)
17. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques, 2nd edn. Elsevier, Morgan Kaufman, Amsterdam (2005)

Ontology-Driven Construction of Domain Corpus with Frame Semantics Annotations

He Tan¹, Rajaram Kaliyaperumal², and Nirupama Benis²

¹ Institutionen för datavetenskap, Linköpings universitet, Sweden

² Institutionen för medicinsk teknik, Linköpings universitet, Sweden

Abstract. Semantic Role Labeling plays a key role in many text mining applications. The development of SRL systems for the biomedical domain is frustrated by the lack of large domain specific corpora that are labeled with semantic roles. In this paper we proposed a method for building corpus that are labeled with semantic roles for the domain of biomedicine. The method is based on the theory of *frame semantics*, and uses domain knowledge provided by ontologies. By using the method, we have built a corpus for transport events strictly following the domain knowledge provided by GO biological process ontology. We compared one of our frames to a BioFrameNet frame. We also examined the gaps between the semantic classification of the target words in this domain-specific corpus and in FrameNet and PropBank/VerbNet data. The successful corpus construction demonstrates that ontologies, as a formal representation of domain knowledge, can instruct us and ease all the tasks in building this kind of corpus. Furthermore, ontological domain knowledge leads to well-defined semantics exposed on the corpus, which will be very valuable in text mining applications.

1 Introduction

The sentence-level semantic analysis of text is concerned with the characterization of events, such as determining "who" did "what" to "whom", "where", "when" and "how" [9]. It plays a key role in text mining (TM) applications such as Information Extraction, Question Answering and Document Summarization. The predicate of a clause expresses "what" took place, and other sentence constituents express the participants in the event. Semantic Role Labeling (SRL) is a process that, for each predicate in a sentence, indicates what semantic relations hold among the predicate and its associated sentence constituents. The relations are described by using a list of pre-defined possible semantic roles for that predicate (or class of predicates). The associated constituents in a clause are identified and their semantic role labels are assigned, as in: [*Transporter* CBG] **delivers** [*Entity*cortisol] [*Destination* to target cells].

1.1 Corpus Annotated with Semantic Roles for Biomedicine Domain

Recently, large corpora have been manually annotated with semantic roles in FrameNet [7] and PropBank [21]. PropBank annotates the semantic roles for

all verbs in the Wall Street Journal (WSJ) news corpus. The FrameNet project collects and analyzes the attestations of target words (both verbs and nouns) from the British National Corpus. With the advent of resources, SRL has become a well-defined task with a substantial body of work and comparative evaluation. As with other technologies in natural language processing (NLP), researchers have experienced the difficulties of adapting SRL systems to a new domain, different than the domain used to develop and train the system.

Biomedical text considerably differs from the PropBank and FrameNet data, both in the style of the written text and the predicates involved. Predicates in the data are typically verbs, biomedical text often prefers nominalizations, gerunds and relational nouns [3,11]. Predicates like *endocytosis* and *translocate*, though common in biomedical text, are absent from both the FrameNet and PropBank data [28,2,25]. Predicates like *block*, *generate* and *transform*, have been used in biomedical documents with different semantic senses and require different number of semantic roles compared to FrameNet [25] and PropBank data [28].

The development of SRL systems for the biomedical domain has been frustrated by the lack of large domain-specific corpora that are labeled with semantic roles. The projects, PASBio [28], BioProp [27] and BioFrameNet [4], have made efforts on building PropBank-like and FrameNet-like corpora for processing biomedical text. Up until recently, these corpora are relatively small. PASBio annotated the semantic roles for 31 predicate (distributed 29 verbs). PASBio used a model for a hypothetical signal transduction pathway of an idealized cell, to motivate verb choices. BioProp annotated the arguments of 30 frequent biomedical verbs found in the GENIA corpus [12]. BioFrameNet built semantic frames in which 32 verbs and also nouns are annotated with the semantic roles. BioFrameNet project considers a collection of GRIF (Gene References in Function) texts that are annotated by the protein transport classes in the Hunter Lab¹ knowledge base (HLKB).

More importantly, these methods have been rather informal. The difficulties of building this kind of corpus include how to discover and define semantic frames together with associated semantic roles within the domain and their interrelations, how to collect and group domain-specific predicates to each semantic frame, and how to collect example sentences from publication databases, such as the PubMed/MEDLINE database containing over 20 million articles?

Corpus construction is time-consuming and expensive. The PropBank project provides argument structures for all the verbs in the Penn Treebank [15]. The VerbNet project [13] made efforts to classify individual verbs in PropBank into VerbNet classes (referring to Levin verb classes [14]). The FrameNet project collects and analyzes the corpus (the British National Corpus) attestations of target words with semantic overlapping. The attestations are divided into semantic groups, and then these small groups are combined into frames. These methods are not applicable to the case of biomedicine domain, since no large corpus of biomedical texts exists. In this paper, we propose that building corpus

¹ Website for Hunter's Bioinformatics research lab: <http://compbio.uchsc.edu/>

that are labeled with semantic roles for the biomedicine domain can be instructed by domain knowledge provided by ontologies. Our successful corpus construction demonstrates that ontologies, as a structured and semantic representation of domain-specific knowledge, instruct and ease all the above tasks.

1.2 Ontologies and Lexical Resources

Interfacing ontologies and lexical resources has been initiated in several works [10, 8, 19]. The article [8] aligns WordNet’s top level to DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering). The authors suggest that such alignment could restructure the lexicon on the basis of ontological-driven principles. The article [19] populates SUMO (Suggested Upper Merged Ontology [18]) with lexical information by mapping WordNet synsets to its concepts. The authors claim that such mapping enables an ontology to be used automatically by NLP applications. More recently, the FrameNet project links FrameNet’s semantic types to SUMO concepts [23]. The linking is to constrain the filler types of frame elements for specific domains. It is the first step of their work aiming at improving FrameNet capability for deductive reasoning with natural language.

BioFrameNet is a domain-specific FrameNet extension. Its *intracellular protein transport* frames and frame elements are mapped to HLKB protein transport classes and slots. BioFrameNet considers a collection of GRIF (Gene References in Function) texts that are annotated by the protein transport classes in HLKB. PASBio and BioProp are the projects that aim to produce definitions of Predicate Argument Structure (PAS) frames in the domain of biomedicine. They do not offer any direct linking of the predicates or their arguments to domain or general ontologies.

Up to recently, biomedical text mining systems have mainly used ontologies as terminologies to recognize biomedical terms, by mapping terms occurring in text to concepts in ontologies, or used ontologies to guide and constrain analysis of NLP results, by populating ontologies. In the latter case, ontologies are more actively used as a structured and semantic representation of domain knowledge. In this paper we proposed a method for building domain-specific corpus annotated with semantic roles based on ontological domain knowledge. We believe that ontological, as a structured and semantic representation of domain-specific knowledge, can instruct and ease all the tasks in corpus construction. Furthermore, ontological domain knowledge leads to well-defined semantics exposed on the corpus, which will be very valuable in text mining applications.

2 Method

The FrameNet project is the application of the theory of *Frames Semantics* [6] in computational lexicography. Frame semantics begins with the assumption that in order to understand the meanings of the words in a language, we must first have knowledge of the background and motivation for their existence in the language and for their use in discourse. The knowledge is provided by the conceptual structures, or *semantic frames*. In FrameNet, a semantic frame describes

an event, a situation or a object, together with the participants (called frame elements (FE)) involved in it. The semantic type (ST) indicates the basic typing of fillers of FE. A word evokes the frame, when its sense is based on the frame. The main relations between frames include *inheritance*, *subframe*, *causative_of*, *inchoative_of* and *using*.

Ontology is a formal representation of knowledge of a domain of interest. They reflect the structure of the domain knowledge and constrain the potential interpretations of terms. An ontology includes concepts that represent sets or classes of entities within a domain. It defines different types of relations among concepts, as well as the rules for combining these concepts and relations. Ontological terms typically are associated with textual definitions which is to give precise meaning of terms within context of a particular ontology in many cases. Intuitively, ontological concepts, relations, rules and their associated textual definitions can be used as the frame-semantic descriptions imposed on a corpus.

A large number of ontologies have been developed in biomedical domain. Many of them contain concepts that comprehensively describe a certain domain of interest, such as Gene Ontology (GO) [1]. GO biological process ontology, containing 20,368 concepts, provides the structured knowledge of biological processes that are recognized series of events or molecular functions. For example, the concept GO:0015031 *protein transport* defines the scenario, "the directed movement of proteins into, out of or within a cell, or between cells, by means of some agent such as a transporter or pore". It is a subclass of GO:0006810:transport and GO:0045184:establishment of protein localization. The class has 177 descendant classes in *is-a* hierarchies. A Protein Transport frame can be effectively described by using these classes and the relations between them.

In many cases ontological terms can be seen as phrases that exhibit underlying compositional structures [17][20]. Fig. 1 presents the compositional structures of 9 direct subclasses describing various types of protein transport. They indicate that *translocation*, *import*, *recycling*, *secretion* and *transport* are the possible predicates, evoking the protein transport event. The more complex expressions, e.g. *translocation of peptides or proteins into other organism* involved in symbiotic interaction (GO:0051808), express participants involved in the event, i.e. the entity (*peptides or proteins*), destination (*into other organism*) and condition (*involved in symbiotic interaction*) of the event.

So far, we using these classes and relations between them, have partly defined the semantic frame *Protein Transport*, decided the participants involved in the event, and listed the domain-specific words evoking the frame. We also can identify the *inheritance* relation between this frame and another frame *Transport* which is defined based on the superclass GO:0006810 *transport*. The complete frame description, including a complete list of frame elements, evoking words, the basic typing of fillers of frame elements, and relations to other frames in the corpus, can be given after studying all the related classes and their relations. Lastly, collecting example sentences will be based on knowledge based search engine for biomedical text, like GoPubMed [5]. As such, domain knowledge provided by ontologies, such as GO biological process ontology and molecular function

ontology, and pathway ontologies, will instruct us in building large frame corpora for the domain. Here, we outline the aspects of ontology driven frame-semantic descriptions:

- The structure and semantics of domain knowledge in ontologies constrain the frame semantics analysis, i.e. decide the coverage of semantic frames and the relations between them;
- Ontological terms can comprehensively describe the characteristics of events or scenarios in the domain, so domain-specific semantic roles can be determined based on terms;
- Ontological terms provide domain-specific predicates, so the semantic senses of the predicates in the domain are determined;
- The collection and selection of example sentences can be based on knowledge-based search engine for biomedical text.

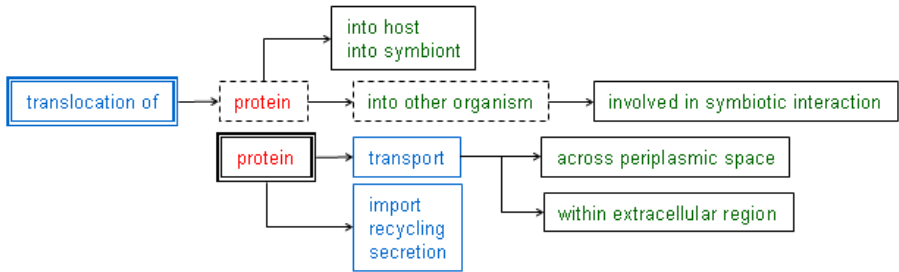


Fig. 1. A concise view of 9 GO terms describing *Protein Transport*. We use the modified finite state automaton (FSA) representation given in [20]. Any path that begins at a start state, represented by double solid borders, and ends in an end state, represented by a single solid border, corresponds to a term. The nodes with a dashed border are neither start states nor end states.

3 Results and Discussion

3.1 A Corpus Covering Transport Events

We built a corpus that currently covers transport events. Most cellular processes are accompanied by transport events. Transport events have been studied by scientists over the last 30 years. For understanding biomedical texts, transport events are among the most important things to know about.

The definition and description of the frames in the corpus strictly follows the domain knowledge provided by the piece of GO describing the events. The core structure of the frame is the same as that of FrameNet. The description of the scenario evoked by the frame is provided, along with a list of the FEs and their definitions. A list of lexical units (LUs) that evoke the frame is provided. In addition, example sentences that contain at least one of the LUs, are given

annotations using definitions of the frame. The annotations follow FrameNet’s guidelines for lexicographic annotation, described in [22].

Resources The piece of GO biological process ontology describing transport events (we call them ”sub-ontology”) currently is considered. GO:0006810 transport is the only root of the sub-ontology. GO:0006810 transport is a general concept whose depth in GO is 3. The sub-ontology very extensively describes transport events. It has 1061 descendant classes in *is-a* hierarchies. The structural dimension of the sub-ontology is given in Table 1. A total of 2235 class names and synonyms are collected for the study. In addition to that from GO concepts, synonyms are also gathered by querying the UMLS Metathesaurus [24].

Table 1. The structural dimension of the sub-ontologies. GO is a directed acyclic graph. The leaf nodes are the nodes having no ingoing *is-a* arc in a graph. Depth relates to the length of *is-a* paths in a graph. Breadth relates to the number of sibling nodes having the same distance from (one of) root node in a graph.

sub-ontology	#node	avg. depth	max. depth	avg. breadth	max. breadth	#leaf
transport	1061	7.7	16	1.9	38	323
protein transport	177	7.3	12	2.5	38	101

Frame. The Transport frame characterizes ”*transport and autonomous movement of substances or cellular components into, out of or within a cell, or between cells, or within a multicellular organism by means of some agent such as a transporter or pore*”, following the definition of GO:0006810 transport. All living cells contain proteins that carry out specialized functions within the cell. Understanding protein transport has been a major focus of cell biology for several decades. Thus, we also include a more specific frame Protein Transport in the corpus, which characterizes transport events of proteins [26]. The description of the frame follows the definition of GO:0015031 protein transport. GO:0015031 protein transport is one of 36 direct subclasses of GO:0006810 transport. The class has 177 descendant classes, together with a total of 581 class names and synonyms. The sub-ontology having the only root GO:0015031 protein transport, is the largest ”sub-ontology” of the transport sub-ontology.

Frame Elements. By studying all the names and synonyms (we call them ”term”), we defined all possible FEs for the frame (see Table 2). For instance, in the term GO:003295 B cell receptor transport within lipid bilayer, lipid bilayer is the location within which protein transport happens. The term GO:0072322 protein transport across periplasmic space describes the path along which protein transport occurs. The term GO:0043953 protein transport by the Tat complex specifies a molecule that carries protein during the movement. GO:0030970 retrograde protein transport, ER to cytosol indicates the direction (**retrograde**) of the movement. An attribute (**SRP-independent**) of the event is described in the term GO:0006620 SRP-independent protein-membrane targeting ER.

Table 3 gives the percentage (number) of the GO terms that indicate the FEs. The percentages are similar in Transport and Protein Transport cases. Although the FE Transport_Place is not indicated in the sub-ontology for protein transport, domain experts identified it as a FE for Protein Transport frame. This conforms to the definition of the *inheritance* relation in [22]. The first 4 FEs are considered as the core FEs by domain experts. They are the participants in the events most indicated by the GO terms as well.

Table 2. FE definitions

FES	definition
Transport_Entity (TE) <i>Core</i>	Protein or protein complex which is undergoing the motion event into, out of or within a cell, or between cells, or within a multicellular organism.
Transport_Origin (TO) <i>Core</i>	The organelle, cell, tissue or gland from which the Transport_Entity moves to a different location.
Transport_Destination (TDS) <i>Core</i>	The organelle, cell, tissue or gland to which the Transport_Entity moves from a different location.
Transport_Condition (TC) <i>Core</i>	The event, substance, organelle or chemical environment which positively or negatively directly influences or is influenced by, the motion event. The substance or organelle does not necessarily move with the Transport_Entity
Transport_Location (TL)	The organelle, cell, tissue or gland where the motion event takes place when the origin and the destination are the same or when origin or destination is not specified.
Transport_Path (TP)	The substance or organelle which helps the entity to move from the Transport_Origin to the Transport_Destination, sometimes by connecting the two locations, without itself undergoing translocation
Transport_Transporter (TT)	The substance, organelle or cell crucial to the motion event, that moves along with the Transport_Entity, taking it from the Transport_Origin to the Transport_Destination.
Transport_Direction (TDR)	The direction in which the motion event is taking place with respect to the Transport_Place, Transport_Origin, Transport_Destination or Transport_Location.
Transport_Attribute (TA)	This describes the motion event in more detail by giving information on how (particular movement, speed etc.) the motion event occurs. It could also give information on characteristic or typical features of the motion event.
Transport_Place (TPL)	The organelle, cell, tissue, gland or organism where the Transport_Origin, Transport_Destination or Transport_Location is situated

Predicates. Table 4 gives the heads of the GO terms (noun phrases), and the number of the GO terms with the head. If it is identified by domain experts that the verb/gerund derived from a head are also used to describe the biological process expressed by the head, the verb/gerund is included as a LU in the corpus.

Table 3. The percentage of the GO terms that indicate the EFs. The number in the bracket indicates the number of GO terms.

sub-ontologies	TE	TO	TDS	TC	TL	TP	TT	TDR	TA	TPL
Protein Transport (581 terms)	99.5% (578)	8.6% (50)	37.4% (159)	16.4% (95)	7.1% (41)	4.6% (27)	1.0% (6)	0.3% (2)	0.2% (1)	0% (0)
Transport (2235 terms)	92.6% (2070)	12.2% (272)	19.3% (432)	9.9% (221)	5.7% (127)	7.3% (164)	1.9% (43)	1.5% (34)	1.8% (40)	0.36% (8)

GO terms, such as *related* and *broad* synonyms², are not always considered in the task. For example, *fat body metabolism*, a *broad* synonym of GO:0015032 *storage protein import into fat body*, is not considered. *Transport* is one of the series of biological processes of *metabolism*.

We collected 129 predicates from transport sub-ontology. Among them, 26 predicates can be found in protein transport sub-ontology. Most of the predicates collected in transport sub-ontologies can be used to describe protein transport event. For some predicates, such as *defecation* and *hydration*, *Transport_Entity* can not be protein. This conforms to the definition of the *inheritance* relation in [22]. Being evoked by a particular set of lexical units represent meta-information of a frame rather than a property of the frame that is strictly semantic.

Semantic Types. We directly identified the domain-specific semantic types (STs) that indicate the typing of fillers of FEs, by using the semantic types from the domain upper-level ontology, UMLS Semantic Network [16]. This kind of STs is mainly used to aid frame parsing and automatic FE recognition [22]. In the UMLS project, the role of the semantic network is to provide a consistent categorization of all concepts represented in the UMLS Metathesaurus.

In our corpus, the semantic type [T072] *Physical Object* from UMLS Semantic Network indicates the typing of fillers of the FEs, *Transport_Entity*, *Transport_Path*, *Transport_Transporter*. [T017] *Anatomical structure* indicates the typing of fillers of the FEs, *Transport_Origin*, *Transport_Destination*, *Transport_Location*, *Transport_Place*. [T017] *Anatomical structure* is a child of [T072] *Physical Object* in UMLS Semantic Network. No semantic types from UMLS Semantic Network can be used to precisely constraint the categories of the other FEs, *Transport_Condition*, *Transport_Attribute*, *Transport_Direction*.

Example Sentences. In our corpus, minimally for each LU 10 annotated sentences are gathered from the PubMed by using GoPubMed, if applicable. For each sentence annotated, we mark the target LU, and collect and record syntactic and semantic information about the relevant frame’s FEs. For each FE, three types of annotation are gathered. The first layer is the identity of the specific FE. In cases when the FE is explicitly realized, phrase type (PT, for exam-

² Synonyms in GO are alternative words or phrases closely related in meaning to the term name. The relationships between the name and synonym include *exact*, *broad*, *narrow* and *related*.

ple NP) and grammatical function (GF) of the realization are annotated. The GFs describe the ways in which the constituents satisfy abstract grammatical requirements of the target word. In cases when the FE is omitted, the type of its null instantiation is recorded. These three layers for all of the annotated sentences, along with complete frame and FE descriptions are used in summarizing valence patterns for each annotated LU. Part of the corpus is now available on <http://www.ida.liu.se/~hetan/bio-onto-frame-corpus>.

The example sentences in the corpus, are retrieved from the MEDLINE database by using the GoPubMed, a knowledge-based search engine for biomedical text. The sentences to be annotated, are always the most relevant and from the latest publications. For LUs derived from one head, we acquired sentences by using the GO terms with a head. The query starts from using the most general GO terms. In the case that the number of query results is huge, more specific terms are used instead. Minimally, 10 sentences are gathered for each LU, if applicable. In cases when only specific GO terms are available and the number of query results is too small, we generalize the query term. For example, the lexical units, `release.n` and `release.v`, are derived and only derived from GO:0002001 renin secretion into blood stream's synonym renin release into blood stream. No query result returns for the GO term. The general term "protein release" is used as the query term instead.

Table 4. The heads of the GO terms (noun phrases). The number in the bracket indicates the number of GO terms with the head.

Transport sub-ontology	absorption (9), budding (3), channel (3), chemiosmosis (1), clearance (4), conductance (4), congestion (1), defecation (1), degranulation (15), delivery (2), discharge (1), distribution (4), diuresis (1), efferocytosis (1), efflux (6), egress (2), elimination (3), endocytosis (16), entry (3), establishment of...localisation (5), establishment of...localization (14), exchange (5), excretion (3), exit (2), exocytosis (25), export (122), flow (1), hydration (1), import (168), influx (1), internalization (9), invagination (1), lactation (1), loading (1), localization (6), micturition (1), migration (7), mobilization (2), movement (4), natriuresis (1), phagocytosis (2), pinocytosis (2), positioning (4), progression (1), reabsorption (3), recycling (14), reexport (2), release (15), removal (1), retrieval (1), reuptake (14), salivation (1), secretion (338), sequestration (1), shuttle (6), sorting (3), streaming (1), targeting (71), trafficking (3), transcytosis (10), transfer (1), translocation (87), transpiration (1), transport (1011), uptake (53), urination (1), voiding (1)
Protein Transport sub-ontology	delivery (1), egress (2), establishment of ... localization (19), exit (2), export (20), import (88), recycling (2), release (1), secretion (226), sorting (4), targeting (68), trafficking (1), translocation (76), transport (100), uptake (5)

3.2 Compared to BioFrameNet

We compared our Protein Transport frame to the frame Protein_transport in BioFrameNet³. BioFrameNet covers the phenomenon of intracellular protein transport. It considered a collection of GRIF texts annotated by 5 HLKB protein transport classes. The HLKB classes are arranged in *is-a* hierarchy. The top level class *protein transport* follows the definition of GO:0015031 protein transport which, however, is a superclass of GO:0006886 intracellular protein transport in GO. 4 FEs are taken from the slot defined for the HLKB classes: *Transported_entity*, *Transport_origin*, *Transport_destination* and *Transport_locations*. 32 predicates are extracted from the collection of GRIF texts.

Several LUs in BioFrameNet are missed in our corpus data. For the LUs *enter.v*, *redistribution.n*, *return.v*, and *traffic.n*, their nominals are absent from GO biological process ontology terms. The second group of missing LUs includes those appear in GO, but in the terms that are not included in the sub-ontology of transport. The LU *recruitment.n* appears in GO:0046799 recruitment of helicase-primase complex to DNA lesions and GO:0046799 recruitment of 3'-end processing factors to RNA polymerase II holoenzyme complex, which describe the movement of protein complex to another macro molecule. *shift.n* only occurs in GO:0003049 regulation of systemic arterial blood pressure by capillary fluid shift, although *capillary fluid shift* describes a kind of transport event. *relocation.n* and *relocate.v* appear in GO:0009902 chloroplast relocation which is considered as a kind of organelle organization.

The number of example sentences for each lexical unit in BioFrameNet relies on the existing collection of GRIFs in HLKB. The number of annotated sentences for each LU ranges from 1 to over 200. In our corpus, minimally for each LU 10 annotated sentences are gathered, if applicable. We noticed that there are differences between the valence patterns of two corpus. One of the reasons is that example sentences in BioFrameNet are GRIF texts which mainly describe about protein itself. Although protein transport is described, different topics may be covered in the sentences in our corpus.

3.3 Predicates in FrameNet and PropBank/VerbNet

We examined the gaps between the semantic classification of the LUs (or verbs) in our corpus, and in FrameNet and PropBank/VerbNet data. 40 of 129 LUs in our corpus appear in FrameNet data. 4 LUs have different semantic senses as in FrameNet when they are used in describing transport events. We identified the FEs and their STs based on the domain knowledge. The number of FEs and their definitions are very different from FrameNet data.

38 of 62 verbs in our corpus are included in PropBank data. 29 of them have the same semantic senses as in PropBank when they are used to describing transport events. 23 of these verbs have been classified into VerbNet classes.

³ We used the data that is publicly available on <http://dolbey.us/BioFN/BioFN.zip> (28-Mar-2009).

4 Conclusions

In this paper we proposed a method for building corpus that are labeled with semantic roles for the domain of biomedicine. The method is based on the theory of *frame semantics*, and relies on domain knowledge provided by ontologies. By using the method, we have built a corpus for transport events strictly following the domain knowledge provided by GO biological process ontology. We compared one of our frames to a BioFrameNet frame. We also examined the gaps between the semantic classification of the target words in this domain-specific corpus and in FrameNet and PropBank/VerbNet data. The successful corpus construction demonstrates that ontologies, as a formal representation of domain knowledge, can instruct us and ease all the tasks in building this kind of corpus. Furthermore, ontological domain knowledge leads to well-defined semantics exposed on the corpus, which will be very valuable in text mining applications.

In the future, we aim to extend the corpus to cover other biological events. GO ontologies will be the main resource to provide domain knowledge, but also other ontologies, such as pathway ontologies, will be considered as important domain knowledge resources. We intend to develop a supporting environment which has the components that support parsing and visualizing lexical properties of ontological terms, retrieving of example sentences from PubMed, defining frame semantics description and annotation task.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29 (2000)
2. Bethard, S., Lu, Z., Martin, J.H., Hunter, L.: Semantic role labeling for protein transport predicates. *BMC Bioinformatics* 9, 277 (2008)
3. Cohen, K.B., Palmer, M., Hunter, L.: Nominalization and alternations in biomedical language. *PLoS ONE* 3(9) (2008)
4. Dolbey, A., Ellsworth, M., Scheffczyk, J.: Bioframenet: A domain-specific framenet extension with links to biomedical ontologies. In: *Proceedings of KR-MED*, pp. 87–94 (2006)
5. Doms, A., Schroeder, M.: Gopubmed: exploring pubmed with the gene ontology. *Nucleic Acids Research* 33, W783–W786 (2005)
6. Fillmore, C.J.: Frames and the semantics of understanding. *Quaderni di Semantica* 6(2) (1985)
7. Fillmore, C.J., Wooters, C., Baker, C.F.: Building a large lexical databank which provides deep semantics. In: *The Pacific Asian Conference on Language, Information and Computation* (2001)
8. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A.: Sweetening wordnet with dolce. *AI Magazine* 3(24), 13–24 (2003)
9. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. *Computational Linguistics* 28(3), 245–288 (2002)

10. Guarino, N.: Some ontological principles for designing upper level lexical resources. In: Proceedings of First International Conference on Language Resources and Evaluation, pp. 527–534 (1998)
11. Kilicoglu, H., Fiszman, M., Roseblat, G., Marimpietri, S., Rindfleisch, T.C.: Arguments of nominals in semantic interpretation of biomedical text. In: Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (2010)
12. Kim, J.D., Ohta, T., Teteisi, Y., Tsujii, J.: Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics* 19(suppl. 1), 180–182 (2003)
13. Kipper, K., Dang, H.T., Palmer, M.: Class-based construction of a verb lexicon. In: AAAI 2000 Seventeenth National Conference on Artificial Intelligence (2000)
14. Levin, B.: English Verb Class and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993)
15. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The penn treebank: annotating predicate argument structure. In: Proceedings of the Workshop on Human Language Technology (1994)
16. McCray, A.T.: An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics* 4, 80–84 (2003)
17. McCray, A.T., Browne, A.C., Bodenreider, O.: The lexical properties of the gene ontology. In: Proceedings of AMIA Symposium, pp. 504–508 (2002)
18. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, pp. 2–9 (2001)
19. 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 (2003)
20. Ogren, P.V., Cohen, K.B., Hunter, L.: Implications of compositionality in the gene ontology for its curation and usage. In: Pacific Symposium on Biocomputing, vol. 10, pp. 174–185 (2005)
21. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics* 31, 71–105 (2005)
22. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended theory and practice. Tech. rep., ICSI (2005), <http://framenet.icsi.berkeley.edu/book/book.pdf>
23. Scheffczyk, J., Pease, A., Ellsworth, M.: Linking framenet to the sumo ontology. In: International Conference on Formal Ontology in Information Systems (2006)
24. Schuyler, P.L., Hole, W.T., Tuttle, M.S., Sherertz, D.D.: The umls metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association* 81(2), 217–222 (1992)
25. Tan, H.: A study on the relation between linguistics-oriented and domain-specific semantics. In: Proceedings of the 3rd International Workshop on Semantic Web Applications and Tools for the Life Sciences (2010)
26. Tan, H., Kaliyaperumal, R., Benis, N.: Building frame-based corpus on the basis of ontological domain knowledge. In: Proceedings of the 2011 Workshop on Biomedical Natural Language Processing, pp. 74–82 (2011)
27. Tsai, R.T.H., Chou, W.C., Su, Y.S., Lin, Y.C., Sung, C.L., Dai, H.J., Yeh, I.T.H., Ku, W., Sung, T.Y., Hsu, W.L.: Biosmile: adapting semantic role labeling for biomedical verbs: an exponential model coupled with automatically generated template features. In: Proceedings of the 2005 Workshop on Biomedical Natural Language Processing (2006)
28. Wattarujekrit, T., Shah, P.K., Collier, N.: Pasbio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics* 5, 155 (2004)

Building a Hierarchical Annotated Corpus of Urdu: The URDU.KON-TB Treebank

Qaiser Abbas

University of Konstanz, Department of Linguistics,
Box D 185, 78457 Konstanz, Germany

qaiser.abbas@uni-konstanz.de

<http://ling.uni-konstanz.de/pages/compling/>

Abstract. This work aims at the development of a representative treebank for the South Asian language Urdu. Urdu is a comparatively under resourced language and the development of a reliable treebank for Urdu will have significant impact on the state-of-the-art for Urdu language processing. In URDU.KON-TB treebank described here, a POS tagset, a syntactic tagset and a functional tagset have been proposed. The construction of the treebank is based on an existing corpus of 19 million words for the Urdu language. Part of speech (POS) tagging and annotation of a selected set of sentences from different sub-domains of this corpus is in process manually and the work performed till to date is presented here. The hierarchical annotation scheme we adopted has a combination of a phrase structure (PS) and a hybrid dependency structure (HDS).

Keywords: Urdu, Treebank, POS, Phrase, Hybrid.

1 History and Introduction

The primary aim of this work is to build a treebank URDU.KON-TB. A treebank or parsed corpus is a text corpus of sentences, annotated with a syntactic structure (a tree structure), hence the name treebank. Similarly, corpus annotation is simply the process of the addition of interpretative linguistic information to a corpus, e.g. addition of tags/labels identifying the class of words in a text. This is so-called part of speech (POS) tagging [1]. A sample of a POS and syntactic annotation scheme is given in Figure 1 for the sentence as follows:

حامد نے شیر کو مارا.

Roman: Hamid ne sher ko mara.

English: Hamid killed the lion.

In this tree KP, PN, P, NN, VP and VB represents case phrase, proper noun, particle, noun, verb phrase and verb respectively. As Urdu is a case marking language, so a KP for case phrase is used and not CP. Similarly, particle P is used to tag case marker(CM) like 'ne' and 'ko' in above tree and not CM. These

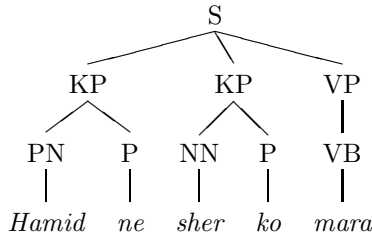


Fig. 1. Example tree for “*Hamid ne sher ko mara*”

issues of ambiguous tagging are discussed in [20]. Anyhow a phrase structure annotation scheme was adopted which has the advantages of simplicity, is easily convertible into bracketed sentences, ‘light’ on resources and the tree structure is relatively easy to read without specialized software tools. The leaf nodes represent words and the nodes connected directly to leaves represent POS tags of the respective words. Other nodes up above the POS tags represent the syntactic annotation for a sentence *S*.

It is pertinent to note that the tree above is only for common understanding of human being. Computers can not process this tree annotation until the link between each node is converted into some computer readable form. So, bracketing of trees is needed here which will be described in design section. The tree does not include functional annotation information e.g. grammatical, semantic and thematic which is included in the current work of URDU.KON-TB treebank with meaningful POS and syntactic annotation.

POS tagging schemes are generally useful for differentiating words which have same spelling but different meanings, for example the word “present” may denote a noun *gift*, a verb to give someone a *present* or an adjective *not absent*. So, having a reliable and linguistically informed annotation scheme is very important. Moreover, there are many annotation schemes for corpora which include semantic (meanings of words), discourse (adding a information about anaphoric links), stylistic (adding information about speech and thought presentation), lexical (adding the identity of the lemma/base/stem of each word form in a text), etc [2]. Annotation can help automatic processing and analysis in many different ways. For example POS tagged corpora can generate frequency list or frequency dictionaries with grammatical classification e.g. the verb “leaves” and the noun “leaves” should be treated differently based on their frequency.

The newly developed 19 million word corpus by [3] available for the Urdu language is a huge corpus, but a balanced approach for selection of set of sentences from each domain of the corpus has been adopted. A tagset based on linguistics distinctions is developed. A portion of the Urdu corpus is first annotated with POS tags and then syntactic analysis is added on top of the POS annotation. The URDU.KON-TB treebank is enriched with sufficient semantic or other linguistic information. Our approach to treebank creation as per recommendation is completely manual at the moment and will be semi-automatically in future for speeding up the procedure on the whole 19 million

corpus. A detail of POS, syntactic, grammatical, semantic and thematic tagging is discussed in the design section, along with a sample bracketed sentence. A combination of a phrase structure (PS) and a hybrid dependency structure (HDS) has been adopted which is also useful for conversion into C-structure and F-structure. This outcome also allows linguistics community to use the resources of these languages in very effective way.

A simple but large difference between phrase and dependency annotation structure is that in phrase structure annotation, the nodes represents phrases/constituents only e.g. KP, VP, etc as shown in Figure 11, while in dependency structure the nodes represents head words or head word plus its syntactic tag as shown in Figure 6(b). Some treebanks alongwith their annotation structure are as follows: the BulTreeBank¹ for the Bulgarian language follows HPSG (Head-driven Phrase Structure Grammar), the Penn Treebank² and ICE-GB³ (International Corpus of the English-Great Britain) for English adopted the phrase structure scheme, the Prague Dependency Treebank⁴ for the Czech language by Czech republic and the Quranic Arabic Dependency Treebank⁵ for the Arabic language by University of Leeds, UK had adopted the dependency structure during treebank annotation. Almost 64 treebank exist for different languages in the world. At present, the languages for which more than three treebanks are available included Arabic, English, German, Italian, Latin and Japanese. A list of treebanks can be seen in 4. The most popular treebank in the history was the Penn treebank for the English language in 1993. A short description of some treebanks is as follows.

The Penn Treebank for English: An annotated corpus containing 4.5 million words of the English language. In the first phase of this project, POS information 5 was encoded along with annotated syntactic structure (tree structure) in parallel with half of the corpus. A large number of work heavily relied on the Penn Treebank e.g. stochastic parsing 6,7,8, skeletal parsing 9,10, training POS taggers 11, disambiguating spoken sentences 12, linguistic theory and psychological modelling 13, grammar development etc. The Penn Treebank had limitations like no clear argument/adjunct relationship, trapping problem, inconsistencies in the annotation scheme, limited annotation and need of predicate-argument structure. Other Treebanks for English are the Susanne Corpus 14, the Lancaster Parsed Corpus 15, and International Corpus of English 16. The German treebank TIGER 17 is based on the NEGRA treebank and the popular treebank for German is Tuba-D/Z. These treebanks (TIGER & Tuba-D/Z) contain 22000 and more than 50000 sentences respectively collected from German newspapers and also annotated with phrase and dependency structure

¹ <http://www.bultreebank.org/> The project was funded by the Volkswagen Stiftung, Germany under the Programme "Cooperation with Natural and Engineering Scientists in Central and Eastern Europe".

² <http://www.cis.upenn.edu/treebank/>

³ <http://www.ucl.ac.uk/english-usage/projects/ice-gb/index.htm>

⁴ <http://ufal.mff.cuni.cz/pdt/>

⁵ <http://corpus.quran.com/>

حامد نے شیر کو افریقہ کے جنگل میں بندوق سے مارا ۔

Roman: Hamid ne sher ko Africa ke jungle
mein bandooq se mara.

English: Hamid killed the lion in the jungle
of Africa with the gun.

(S
 (KP (PN حامد) (P نے))
 (KP (NN شیر) (P کو))
 (PREP
 (NP
 (GP
 (PN افریقہ)
 (P کے))
 (NN جنگل))
 (PRE میں))
 (SEP (NN بندوق) (SE سے))
 (VP (VB مارا))
 (SM .))

Fig. 2. A sample bracketed sentence from NU-FAST Treebank

respectively. Each treebank used Stuttgart Tübingen POS Tagset along with 49 and 36 grammatical function labels respectively [18]. TIGER has a flat annotation scheme with no unary branching while the other one allows for this and contains a deeper hierarchical structure.

The selection of annotation scheme is totally dependent on the constituent ordering of the language. Phrase structure is good for fixed constituent order languages like English, Bulgarian, Chinese, etc., while dependency structure is good for free constituent order languages like Urdu, Hindi, German, etc. [19]. The current work builds on a previous study at constructing a NU-FAST treebank [20]. However, the design of that treebank proved to be too simple and flat. It neither contained detailed syntactic, morphological, semantic and thematic information, nor any information about displaced constituents/phrases, empty arguments or traces. In addition, it was based on a POS-Tagset that is currently revised by the author due to issues like the word 'ne' in Figure 1, which should be tagged as case marker CM instead of particle P [21]. A bracketed sentence from the NU-FAST Treebank is given in Figure 2.

Another Hindi-Urdu treebanking effort is under way in a collaborative project⁶ between five different universities at Colorado. However, the Urdu treebank being developed is comparatively small and is being done as part of a larger effort at establishing a treebank for Hindi. Although Urdu and Hindi share many structural features, there are some interesting differences and as the main effort of the Hyderabad-Colorado cooperation is focused on Hindi, many of the issues with

⁶ <http://verbs.colorado.edu/hindiurdu/>

respect to Urdu are remaining unresolved. Our work take up these issues and proposing solutions for them with passage of time.

2 Design

The first objective achieved of this URDU.KON-TB treebank, is the analysis of the corpus. This 19 million word corpus is available at Centre for Language Engineering (CLE)⁷. This corpus is in a very balanced and normalized form already. The corpus has six different domains which are from C1 to C6. The corpus does not yet include ethical and religious domain C7(shown in italics), which is ready after sufficient inspection and investigation but not yet merged. The domains and sub-domains of the corpus with size distribution and number of distinct words are given in Table 1. For treebank work, Samples of 200 sentences from each domain are selected, which comes up with 1400 sentences. The work of manual annotation with PS & HDS has been completed for domains C1 to C3 and information extracted till to date is presented in this paper.

The second objective of this work achieved is to study and investigate encoding & annotation scheme which is a combination of PS & HDS annotation and find useful for the Urdu language. Since, Urdu is a free word order plus case marked language and head word can not be found in sequence like in English most of the time. Hence, HDS annotation solve the problem here alongwith the PS. As for as the encoding scheme related to POS tagging is concerned, an existing POS tagger [22] is being used to tag the words in a sentence computationally just to speed up the process and then these tags are manually corrected & updated during its annotation process. The existing tagger only has very basic functionality with a limited tagset. It was therefore decided by our integrated team (KN + UET)⁸ that a new POS tagset should be developed. This linguistically motivated POS tagset is almost in existence after the completion of C1 to C3 domain's sentences. A sample of the newly developed POS tagset is given in Table 2 and dot '.' is used to add multiple subcategories in a main category e.g. V.LIGHT.PERF, which is a verb V as main category having light LIGHT and perfective PERF concepts as subcategories, concluding hierarchical structure.

The tagset in Table 2 represents a complete POS tagset extracted from the manually annotated sentences of domains C1 to C3 of the corpus. For example, adjectives are being dealt with in four ways. A general category of adjectives as ADJ, a manner category of adjectives as ADJ.MNR, a spatial category of adjectives as ADJ.SPT and a temporal category of adjectives as ADJ.TMP. The Relevant examples are provided in Figure 3.

The addition of this .SPT and .TMP after the syntactic tag ADJ for adjective represents spatial and temporal adjectives respectively, however this '.'

⁷ CLE, University of Engineering and Technology (UET), Lahore, Pakistan (<http://www.cle.org.pk>).

⁸ KN + UET is a team of University of Konstanz, DE and University of Engineering & Technology, PK.

Table 1. Existing Corpus at CLE

Domains	Sub Domains
C1. Sports/Games	C1.1.Sports (special events)
C2. News	C2.1. Local and international affairs C2.2. Editorials and opinions
C3. Finance	C3.1. Business, domestic and foreign market
C4. Culture/Entertainment	C4.1. Music, theatre, exhibitions, review articles on literature C4.2. Travel / tourism
C5. Consumer Information	C5.1. Health C5.2. Popular science C5.3. Consumer technology
C6. Personal communications	C6.1. Emails, online discussions, editorials,e-zines
C7. Ethics and Religious	C7.1. History, Online discussion, Preaching literature, e-magazines

Domains	Size	Distinct words
C1. Sports/Games	1,666,304	23,118
C2. News	8,957,259	67,365
C3. Finance	1,162,019	17,024
C4. Culture/Entertainment	3,845,117	59,214
C5. Consumer Information	1,980,723	34,151
C6. Personal communications	1,685,424	30,469
C7. Ethics and Religious	2,756,695	28,170
Total	22,053,541	132,511

dot notation can also be used for morphological purposes e.g. for the continuous/progressive verb form, a V.PROG tag is being used. Similarly, adverbs which are mostly used as a qualifier of verbs can also be used independently. Adverbs are categorized into six forms presented in Table 2. Some examples of adverbs are given in Figure 4. The examples quoted above and below are to give an idea how POS encoding/tagging has been assigned to given words in a sentence of a corpus.

In URDU.KON-TB treebank, an intermediate approach of PS and HDS has been adopted. In this annotation scheme, the PS approach is implemented on an outer level (physical) and the HDS approach is implemented at an inner level (logical). For example, in Figure 5, bold formatted noun phrase NP is extracted from the hidden concept of noun lying among case phrase KP, noun N, coordination conjunction C.CORD and coordination phrase CP at the same level inside the NP. This is basically logical concept of HDS adopted, while the physical annotation without head words as in Figure 5 is PS. Due to this combination, it will also be very easy for linguists to obtain additionally the XLE⁹

⁹ XLE is a rule based parser introduced by Xerox and Palo Alto Research Center (PARC) in 1993.

Table 2. The URDU.KON-TB Part of Speech Tagset

ADJ (Adjective)	QW (Question Word)
ADJ.MNR (Manner ...)	SM (Sentence Marker)
ADJ.TMP (Temporal...)	U (Unit)
ADJ.SPT (Spatial ...)	V.IMPERF (Imperfective ...)
ADV (Adverb)	V.INF (Infinitive Verb)
ADV.DEG (Degree ...)	V.INF.OBL (Oblique ...)
ADV.MNR (Manner ...)	V.LIGHT (Light Verb)
ADV.NEG (Negative ...)	V.LIGHT.IMPERF (...)
ADV.SPT (Spatial...)	V.LIGHT.INF (Infinitive ...)
ADV.TMP (Temporal...)	V.LIGHT.PERF (Perfective ...)
C.CAUS (Causative Conjunction)	V.LIGHT.ROOT (Root...)
C.CORD (Coordination ...)	V.LIGHTV.PERF (... Light-Verb ...)
C.SBORD (Subordinate ...)	V.MOD (Modal Verb)
CM (Case Marker)	V.PASS.INF (Infinitive Passive ...)
DATE.CAL (Calendar Date)	V.PASS.LIGHTV.IMPERF (...)
DATE.Y (Year Date)	V.PERF (Perfective Verb)
DATE.Y.CAL (Calendar ...)	V.ROOT (Root Verb)
IZF (Izafe)	V.ROOT.LIGHT (Light ...)
KER (Ker)	V.ROOT.LIGHTV (...)
N (Noun)	V.ROOT.PERF (Perfective ...)
N.ADJ (Adjectival ...)	V.TB.PERF (Perfective To-Be Verb)
N.CURR (Currency ...)	VALA (Vala)
N.PROP (Proper ...)	VAUX.COP (Copula Verb-Auxiliary)
N.PROP.DATE.M (Month...)	VAUX.COP.IMPERF (...)
N.PROP.SPT (Spatial ...)	VAUX.COP.PRES (Present ...)
N.SPT (Spatial...)	VAUX.COP.ROOT (Root ...)
N.TMP (Temporal...)	VAUX.IMPERF (Imperfective ...)
P (Pronoun)	VAUX.LIGHT (Light Verb-Auxiliary)
P.DEM (Demonstrative ...)	VAUX.LIGHTV.IMPERF (...)
P.INDF (Indefinite ...)	VAUX.LIGHTV.PERF (...)
P.PERS (Personal ...)	VAUX.MOD (Modal Verb-Auxiliary)
P.REF (Reflexive ...)	VAUX.MOD.IMPERF (...)
P.REF.POSS (Possessive...)	VAUX.MOD.PERF (...)
P.REL (Relative ...)	VAUX.PASS.IMPERF (...)
P.REL.DEM (...)	VAUX.PASS.PERF (... Passive ...)
P.REL.PERS (Personal ...)	VAUX.PASS.ROOT (Root ...)
POSTP (Post Position)	VAUX.PERF (... Verb-Auxiliary)
POSTP.SPT (Spatial...)	VAUX.PROG (Progressive ...)
POSTP.TMP (Temporal...)	VAUX.PROG.PERF (...)
PREP (Pre Position)	VAUX.REP.HABT (Habitual ...)
PT (Particle)	VAUX.REP.HABT.IMPERF (...)
Q (Quantifier)	VAUX.REP.HABT.LIGHT (...)
Q.CARD (Cardinal ...)	VAUX.TB.IMPERF (... To-Be ...)
Q.FRAC (Factional ...)	VAUX.TENS (Tense Verb-Auxiliary)
Q.ORD (Ordinal ...)	VAUX.TENS.COP (Copula ...)
	VAUX.TENS.LIGHT (...)

1. اچھا لڑکا (*Good boy*)
Acha larka
ADJ N
Here the word *Acha* 'good' is used as an adjective
2. ملتانى كھسہ (*Multani shoe*)
Multani khussah
ADJ.SPT N
Multani 'city name' is as a spatial adjective ADJ.SPT
3. گذشتہ سال (*Previous year*)
Guzashta sal
ADJ.TMP N
Guzashta 'previous' is as a temporal adjective ADJ.TMP
4. جابرانہ حکومت (*Cruel government*)
Jaberana hakoomat
ADJ.MNR N
Jaberana 'cruel' is manner adjective ADJ.MNR

Fig. 3. Examples of Adjective

1. تقریباً ساری دنیا میں (*Almost in the whole world*)
Taqreeban sari dunia mein
ADV Q N.SPT CM
Taqreeban 'Almost' is used as an adverb ADV
2. عمارت مکمل نہ ہو سکی۔ (*The building could not be completed.*)
Emarat mukamal na ho saki
N ADJ ADV.NEG V.LIGHT.ROOT V.MOD SM
na 'not' here is used as a negative adverb ADV.NEG
3. بہت اچھی لڑکی (*Very good girl*)
Bohat achi larki
ADV.DEG ADJ N
bohat 'very' is used as a degree adverb ADV.DEG

Fig. 4. Examples of Adverbs

(Xerox Linguistic Environment) parser like C-structure/phrase structure and F-structure/dependency structure from this treebank. . A phrase/constituent structure annotation example is already depicted in Figure 1 and can also be seen in Figure 2. Bracketed notation in Figure 2 is equivalent to the C-structure of XLE parser. However, the dependency structure is distinct from phrase structure annotation as discussed earlier. Dependency structure is not dependent on a specific word order and hence well suited to free word order Urdu language. Mostly, dependency structure is presented in arrows pointing to head word from its dependent words or vice versa as shown in Figure 6(a). However, it can also be

اخبارات مسلمانوں کی لاشوں اور چیخ و پکار کی تصویروں
 سے بھرے ہوتے ہیں -
 Roman: Akh-barat musalmano ki lashon aur cheekh o
 pukar ki tasweeron se bhare hote hein.
 English: The newspapers are used to full of pictures of
 Muslim's corpses and cries.

(S

(NP.NOM-SUB

(N اخبارات))

(KP.INST-OBL

(NP

(KP.POSS

(NP

(KP.POSS

(N مسلمانوں) (CM کی))

(N لاشوں) (C.CORD اور))

(CP

(N پکار) (و) (C.CORD) (N چیخ))

(CM کی))

(N تصویروں))

(CM سے))

(VCMAIN

(V.PERF بھرے) (VAUX.TB.IMPERF ہوتے)

(VAUX.TENS ہیں))

(SM .))

Fig. 5. Physical and logical concept

presented in tree form but then the main verb should be used as root (predicate) of the sentence as in Figure 6(b).

A physical concept of phrase structure and logical concept of dependency structure is merged in the current development of URDU.KON-TB treebank. Moreover, normally dependency structure is limited to the word level e.g. concluding head word from given words, but we enhanced and implemented this idea further at constituent/phrase level which is named as hybrid dependency structure. It means concluding head constituent from given constituents i.e. concluding relationship among head constituent and dependent constituents (hybrid dependency). This whole scheme is named as combination of phrase structure and hybrid dependency structure. The need of such type of schemes is highly advocated in literature such as [19,23], etc. To build such schemes, it is very important that you must have some POS, syntactic and functional tagset. The syntactic tagset is shown in Table 3.

The bracketing form of Figure 6(c) can also be represented in the following bracketing form with some addition of functional tags which we have built in our

Table 3. The URDU.KON-TB Syntactic Tagset.

Tags	Description
ADJP	Adjective Phrase
ADVP	Adverb Phrase
CL.KER	Ker Clause
CP	Conjunction Phrase
DATEP	Date Phrase
KP	Case Phrase
KP.ACC	Accusative KP
KP.DAT	Dative KP
KP.ERG	Ergative KP
KP.INST	Instrumental KP
KP.POSS	Possessive KP
NP	Noun Phrase
NP.ACC	Accusative NP
NP.DAT	Dative NP
NP.ERG	Ergative NP
NP.NOM	Nominative NP
NP.OBL	Oblique NP
PP	Pre/Post Position Phrase
QP	Quantifier Phrase
S	Sentence
SBAR	Subordinate Clause
SBARQ	Question Subordinate Clause
SQ	Yes/No Question Sentence and Subconstituent of SBARQ
UP	Unit Phrase
VALAP	Vala Phrase
VALAP.NOM	Nominative VALAP
VCMAIN	Verb Complex Main
VCP	Verb Complex Predicate
VIP	Verb Infinitive Phrase

URDU.KON-TB treebank e.g. PRD, SUB, OBJ, etc.. This bracket form is very close to F-structure of XLE parser. Moreover, the functional tagset developed for the URDU.KON-TB treebank is given in Table 4.

```
[ PRED mara
  [ SUBJ Hamid
    [ CM ne] ]
  [ OBJ sher
    [ CM ko] ] ]
```

A sample of a bracketed sentence using POS, syntactical and functional tagging can be seen in Figure 7 and compare this tagging and encoded information to the existing NU-FAST treebank output given in Figure 2 earlier. This new output in Figure 7 is the achievement of our third objective of the project which

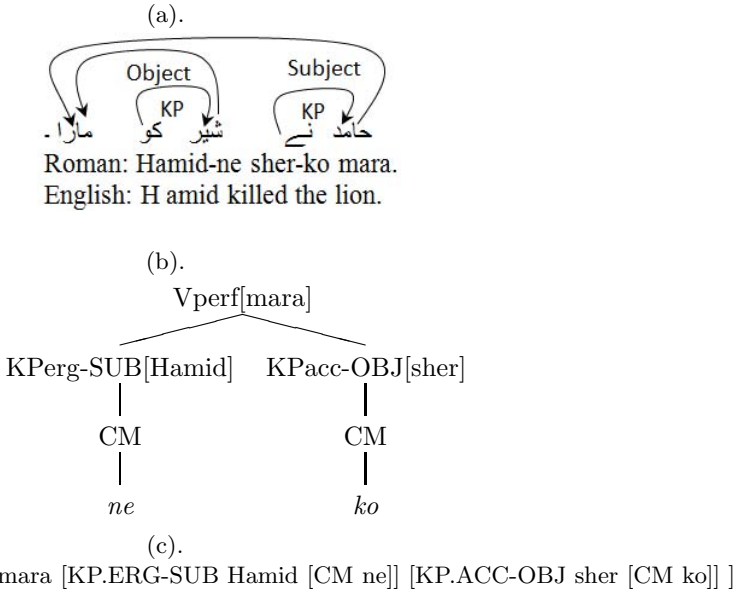


Fig. 6. (a). Dependency arrow (b). Dependency tree (c). Dependency bracket form

Table 4. The URDU.KON-TB Functional Tagset

Grammatical	Semantic/Thematic
-PRD (Predlink)	-DEG (Degree)
-OBJ (Direct Object)	-SPT (Spatial)
-SUB (Subject)	-TMP (Temporal)
-OBJ2 (Indirect Object)	-MNR (Manner)
-OBL (Oblique)	-CMP (Comparative)
	-INST (Instrumental)
Miscellaneous	
* (for Empty Categories)	
-L (for linking displaced constituents/categories)	
-1/-2/-3.... (for labelling of displaced constituents/categories)	

enables us to produce a linguistically enriched treebank. The symbol '//' (double slashes) is used to explain the terms in form of comments. In URUD.KON-TB treebank, the structures are identified and approach of understanding & labelling words is purely realistic and humanistic.

حامد نے شیر کو افریقہ کے جنگل میں بندوق سے مارا .
 Hamid Ne Sher Ko Africa Ke Jangle Mein Bandoq Se Mara .
 Hamid killed the lion in the jungle of Africa with the gun.

(S //rootS (SYNTACTICAL)
 (KP.ERG-SUB //Ergative Case Phrase (SYNTACTICAL), and Subject (GRAMATICAL)
 (N.PROP حامد) (CM نے) //proper noun (POS) //case marker(POS)
 (KP.ACC-OBJ //Accusative Case phrase (SYNTACTICAL), and Object (GRAMATICAL)
 (N شیر) (CM کو) //noun (POS) //case marker (POS)
 (KP-SPT //Case Phrase (SYNTACTICAL) and spatial (THEMATIC)
 (NP //Noun Phrase (SYNTACTICAL)
 (KP.POSS //possessive case phrase (SYNTACTICAL)
 (N.PROP افریقہ) (CM کے) // proper noun (POS) //case marker (POS)
 (N.SPT جنگل) //spatial noun (POS)
 (CM میں) //case marker (POS)
 (KP.INST //Instrumental Case Phrase(SYNTACTICAL)
 (N بندوق) (CM سے) // noun (POS) //case marker (POS)
 (VCMAN //Verb Complex Main or Verb Phrase (SYNTACTICAL)
 (V.PERF مارا) //perfective verb (MORPHOLOGICAL & SYNTACTICAL)
 (SM .) //sentence marker (SYNTACTICAL)

Fig. 7. A bracketing sentence from URDU.KON-TB tree bank with linguistically encoded information

3 Conclusion

A limited sized standard URDU.KON-TB treebank for the Urdu language is the main output yet after the first phase of this work. However, additional resources developed till to date contained a standard POS tagset, a syntactic tagset, a functional tagset and new tagged corpus. These resources will be enhanced further as the work progress. These resources can all be used for natural language processing (NLP) such as probabilistic parsing, training of POS taggers, disambiguation of spoken sentences, grammar development [24], language identification [25], sources for linguistic inquiry and psychological modelling, pattern matching, and in many applications of NLP and machine learning domains [26] & [27].

Acknowledgments. The author would like to express his gratitude to Prof. Dr. Miriam Butt, University of Konstanz for her encouragement, guidance and support. I am also indebted to the members of our integrated team in the various stages of this being continued research project.

References

1. Leech, G.: Adding linguistic annotation. In: Wynne, M. (ed.) Developing Linguistic Corpora: A Guide to Good Practice, ch. 3, pp. 17–29. Oxbow Books, Oxford (2005)

2. Garside, R., Leech, G.N., McEnery, T.: *Corpus annotation: linguistic information from computer text corpora*. Longman, London (1997)
3. Ijaz, M.: *Urdu 5000 Most Frequently Used Words: Technical Report*, Center for Research in Urdu Language Processing (CRULP), Lahore, Pakistan (2007)
4. Wallis, S.: *Searching treebanks and other structured corpora*. In: Lüdeling, A., Kytö, M. (eds.) *Corpus Linguistics: An International Handbook*. Handbücher zur Sprache und Kommunikationswissenschaft, ch. 34. Mouton de Gruyter, Berlin (2008)
5. Santorini, B.: *Part-of-speech tagging guidelines for the Penn treebank project: Technical report MS-CIS-90-47*, Department of Computer and Information Science, University of Pennsylvania (1990)
6. Brill, E.: *Discovering the lexical features of a language*. In: 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA (1991)
7. Brill, E., Magerman, D., Marcus, M.P., Santorini, B.: *Deducing linguistic structure from the statistics of large corpora*. In: *DARPA Speech and Natural Language Workshop* (1990)
8. Magerman, D., Marcus, M.P.: *Parsing a natural language using mutual information statistics*. In: *AAAI* (1990)
9. Pereira, F., Schabes, F.: *Inside-outside re-estimation from partially bracketed corpora*. In: 30th Annual Meeting of the Association for Computational Linguistics (1992)
10. Weischedel, R., Ayuso, D., Bobrow, R., Boisen, S., Ingria, R., Palmucci, J.: *Partial parsing: a report of work in progress*. In: 4th *DARPA Speech and Natural Language Workshop* (1991)
11. Meteer, M., Schwartz, R., Weischedel, R.: *Studies in part of speech labelling*. In: 4th *DARPA Speech and Natural Language Workshop* (1991)
12. Veilleux, M.N., Ostendorf, M.: *Probabilistic parse scoring based on prosodic features*. In: 5th *DARPA Speech and Natural Language Workshop* (1992)
13. Niv, M.: *Syntactic disambiguation*. *The Penn Review of Linguistics* 14, 120–126 (1991)
14. Sampson, G.: *English for the computer: The SUSANNE corpus and analytic scheme*. Clarendon Press, Oxford (1995)
15. Leech, G.: *The Lancaster Parsed Corpus*. *ICAME Journal* 16(124) (1992)
16. Greenbaum, S.: *Comparing English worldwide: The International Corpus of English*. Clarendon Press, Oxford (1996)
17. Dipper, S., Brants, T., Lezius, W., Plaehn, O., Smith, G.: *The TIGER Treebank*. In: *Third Workshop on Linguistically Interpreted Corpora LINC 2001*, Leuven, Belgium (2001)
18. Schiller, A., Teufel, S., Stoeckert, C.: *Vorläufige Guidelines fuer das Tagging deutscher Textcorpora mit STTS(Deutsche): Technical Report*, IMS-CL, University Stuttgart (1995)
19. Skut, W., Krenn, B., Brants, T., Uszkoreit, H.: *An Annotation Scheme for Free Word Order Languages*. In: *Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C (1997)
20. Abbas, Q., Karamat, N., Niazi, S.: *Development of Tree-bank based probabilistic grammar for Urdu Language*. *International Journal of Electrical & Computer Science* 09(09), 231–235 (2009) ISSN: 2077-1231
21. Butt, M., King, T.H.: *The Status of Case*. In: Dayal, V., Mahajan, A. (eds.) *Clause Structure in South Asian Languages*, pp. 153–198. Springer, Berlin (2005)

22. Sajjad, H., Schmid, H.: Tagging Urdu Text with Parts of Speech: A Tagger Comparison. In: 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2009 (2009)
23. Clark, A., Fox, C., Lappin, S.: The Handbook of Computational Linguistics and Natural Language Processing. Blackwell Handbooks in Linguistics, vol. 52, pp. 239–244. John Wiley and Sons (2010) ISBN: 1405155817, 9781405155816
24. Abbas, Q., Khan, A.H.: Lexical functional grammar for Urdu modal verbs. In: 5th IEEE (ICET) 2009 International Conference on Engineering and Technology, pp. 07–12 (2009)
25. Abbas, Q., Ahmed, M.S., Niazi, S.: Language Identifier for Languages of Pakistan Including Arabic and Persian. International Journal of Computational Linguistics (IJCL) 01(03), 27–35 (2010) ISSN: 2180-1266
26. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English. Computational Linguistics (CL) 19(2), 313–330 (1993)
27. Bies, A., Ferguson, M., Katz, K., Macintyre, R.: Bracketing guidelines for Treebank II style penn treebank project: Technical Report, University of Pennsylvania (1995)

A Morphological Analyzer Using Hash Tables in Main Memory (MAHT) and a Lexical Knowledge Base

Francisco J. Carreras-Riudavets, Juan C. Rodríguez-del-Pino,
Zenón Hernández-Figueroa, and Gustavo Rodríguez-Rodríguez

Departamento de Informática y Sistemas,
Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas, Spain
{fcarreras, jcrodriguez, zhernandez, grodriguez}@dis.ulpgc.es
<http://tip.dis.ulpgc.es>

Abstract. This paper presents a morphological analyzer for the Spanish language (MAHT). This system is mainly based on the storage of words and its morphological information, leading to a lexical knowledge base that has almost five million words. The lexical knowledge base practically covers the whole morphological casuistry of the Spanish language. However, the analyzer solves the processing of prefixes and of enclitic pronouns by easy rules, since the words that can include these elements are much and some of them are neologisms. MAHT reaches a processing average speed over 275,000 words per second. This one is possible because it uses hash tables in main memory. MAHT has been designed to isolate the data from the algorithms that analyze words, even with their irregular forms. This design is very important for an irregular and highly inflectional language, like Spanish, to simplify the insertion of new words and the maintenance of program code.

Keywords: Morphological analysis, lemmatization, lexical knowledge base, computational linguistics, natural language processing.

1 Introduction

The automated morphological analysis is a fundamental task [16], [34] to solve important issues of natural language processing and computational linguistics, such as: PoS-tagging, word sense disambiguation, text summarization [21], information retrieval [7], [16], information extraction [5], etc.

The purpose of morphological analysis is to identify for any word its canonical form or lemma —lemmatization—, its grammatical category (name, adjective, verb...) and its inflection (gender, number, diminutive...) [16], [27], [32], [33]. The result of the morphological analysis frequently offers a multiple answer: if a word can correspond with more than one canonical form and category, or with different inflections of the same canonical form, the morphological analysis must provide all the possibilities. This multiplicity generates an ambiguity when we want to apply the analyzer, for example, to a PoS tagger. A part-of-speech tagger can assign the correct interpretation to the word-form, taking context into account [16].

A task related to morphological analysis is the synthesis or morphological generation [7], [16], [22], [35], consisting of *given a canonical form and some characteristics of desired inflection, a word is obtained*.

Many authors have recognized the difficulty that involves the automatic processing of highly inflectional and irregular languages such as Spanish, Polish, German, French and Finnish,... [10], [28], [32], [35]. This means that the development of automatic tools of morphological processing is complicated and moreover necessary. For example, in the Spanish language, most of the frequently used words belong to the following groups:

- Over 3,000 irregular verbs: *ir, ser, estar, tener, poder, poner...*
- Over 8,000 canonical forms change gender or number affecting the written accent of the word: *alemán/alemana, compás/compases...*
- Over 2,000 canonical forms change gender or number in an irregular way: *príncipe/princesa, cualquier/cualesquiera...*
- Over 1,400 canonical forms have irregular suffixes: *nariz/narigón, azúcar/azuquita...*
- Over 100 adjectives have the irregular superlative: *pobre/paupérrimo, sabio/sapientísimo...*

2 Related Works

Many works on automated morphological analysis base their solution on Koskenniemi's two-level morphology model [19], which was initially developed in order to process Finnish. It is a system of rules that, when they are executed in parallel, establishes a one by one transition between the surface level symbols and those of the lexical level. Jointly with the application of the rules, the model explores a lexicon that serves as a lexical filter. In principle, the Koskenniemi model (1983) is independent of the language. However, some authors indicate that, in any case, it requires developing the rules adapted for each language [33]. Other authors point out some difficulties to apply the model to languages like Spanish or the Slavic languages, which present a high number of alterations in the stem [35].

Another method to confront morphological analysis, which can be very effective in applications for natural language processing, is using a lexical knowledge base [28] and not using rules. For example, the works of Sgarbas [33] use a directed acyclic word graph [26] to represent words, stems and grammatical information, so that it simplifies the analysis and synthesis process by searches in the graph, increasing the speed. The authors apply this method, which is independent of the language, to the morphological analysis of the Greek modern language. According to them, it is much faster —10,000 words per second— than a two-level morphological analyzer. The two-level morphological analyzer with the same number of stems only processed 20 words per second. The authors point out that the data structure size is a disadvantage.

The optimization of storage space has also interested other authors who work using a lexical knowledge base, due to the space restrictions that existed years ago. This has led them to develop methods based on segmentation algorithms. These algorithms divide the word in smaller meaningful components, which have similarities to linguistic morphemes —smallest meaningful unit in the grammar of a language—, and to store these components separately.

Baldzis [8] has created a stems and suffixes knowledge base, which includes word formation rules by the Modern Greek language. With this database, his system recognizes approximately 1,000,000 words, although it does not specify the recognition speed. Papakitsos [24] has developed another one for the same Greek language in which he does not specify the universe of words that it is capable of recognizing, but state the precision —98.2%— over a corpus of 1,879,000 Greek forms.

Sedláček [32] uses a trie structure [18] to store the stems of the Czech language, in order to take advantage of the sharing of prefixes among them. The author implements the trie as a minimal finite state automaton [15] with the purpose of reducing the space requirements. In this way, he finds that with a database of approximately 2 MBytes composed by stems, suffixes and morphological patterns, it may be possible to recognize and generate 5,678,122 words of the Czech language —the supported lexicon by the original Koskenniemi model (1983) also used tries, which is a very good structure to store strings of characters that share common prefixes.

STILUS is a lexical platform that has been developed for the Spanish language [36]. STILUS uses a run-time lexicon that is a collection of allomorphs for both, stems and endings, as well as a small rules component. This model permits word formation based solely on morpheme concatenation, driven by a feature-based unification grammar. STILUS recognizes over 1,700,000 words corresponding to 88,227 canonical forms, although it does not specify the recognition speed.

The Automatic Inflector and Lemmatizer of Spanish Words (AILESW) [31] also apply the separation between stems and suffixes. AILESW uses the trie structure to store the suffixes, while the stems are stored on a compressed hash table [18] in external memory, which reduces its processing average speed to 1,672 words per second. The usefulness of AILESW is approved by more than 4,900,000 inflectional or derived words that treat the system, which includes all the entries of the different dictionaries and books: *Diccionario de la Real Academia Española* [30], *Diccionario General de la Lengua Española* [9], *Diccionario de Uso de la Lengua Española* [23], *Gran Diccionario de la Lengua Española* [20], *Diccionario de Uso del Español Actual* [14], *Gran Diccionario de Sinónimos y Antónimos* [17], *Diccionario Ideológico de la Lengua Española* [13], *Nuevo Diccionario de Voces de Uso Actual* [3] and *Todos los Verbos Castellanos Conjugados* [2]. AILESW contemplates about 15,000 words that correspond to surnames and proper nouns of person, animal, thing or place. It also includes about 9,000 adjectives derived from verbal participles, which the mentioned dictionaries do not include.

3 Lexical Knowledge Base

A solution using mainly a lexical knowledge base instead of applying rules, minimizes the difficulty that involves the processing of a highly inflectional and irregular language. The morphological analyzer (MAHT), that we present, uses a lexical knowledge base with 4,980,387 Spanish words. This database includes the words generated by Santana [31], and it has been updated with the currents published modifications by the Royal Spanish Academy: the verbal conjugations of speech in Argentina, according to the new accentuation rules and verbal monosyllables [1].

Almost five million of words have been mainly generated in an automatic way when inflecting the 196,597 canonical forms of the universe of consulted sources [2-3], [9], [13-14], [17], [20], [23], [30], and some irregular forms have been manually introduced in the knowledge base. The inflections generated for the verbs are: the simple conjugation, the inflection of the participle as a verbal adjective —gender and number— and the diminutive of the gerund. The inflections generated for the non-verbal forms are:

- The gender and number for the substantives, adjectives, pronouns and articles.
- The heteronymy by change of sex in the substantives.
- The superlative for adjectives and adverbs.
- The adverbialization for the superlative.
- The diminutive, augmentative and pejorative for substantives, adjectives and adverbs
- Variant graphs in all grammatical categories.
- The invariant forms such as prepositions, conjunctions, exclamations and some neologisms originated from words of other languages and several expressions or phrases.

The wide variety of recognized words by MAHT improves to other known morphological analyzers for Spanish [31], [36]. The lexical knowledge base includes, in addition, the necessary morphological information for the recognition and generation of each word —grammatical category and inflection with respect to its canonical form. We have stored and organized the lexical information on relational database —Oracle—, which makes easier the maintenance of the information and the insertion of new words. This last operation is independent of the data already stored and of the recognition and generation algorithms. This fact improves the system integrity. The words included in the database, according to the test results, which we present further on, include a wide range of the Spanish language.

To carry out the suitable morphological processing, of the information included in the database, we have constructed some specific structures, described in the following section, to reach the maximum performance with the resources of the current processor and memory. This is necessary, since Oracle, as the majority of the commercial database systems for general purpose, utilizes B-Trees [8] as the basic structure for its indexes [4]. This structure does not provide an optimal answer speed for an integrated morphological analyzer inside system for natural language processing. Concretely, the tests carried out under the same conditions as MAHT, showed that the recognition speed was lower than 3,000 words per second, it is below the speed reached with the hash tables proposed in this article, as we show in the Section 7.

4 Data Structure

A hash table is a good data structure for handling large lexicons of words [37]. The structure of MAHT's operative data is made up of two hash tables. The program loads these tables in RAM memory from individual files generated from the exported information from Oracle database: the first one includes the word, its key of inflection and a key corresponding to its canonical form, and the second one, includes the canonical form, its primary key and its grammatical category.

The performance of a hash table depends on the used hash function. As the data set is known before the creation of the structure, we have tested different hash functions with the purpose of achieving an optimum performance. We have tested hash functions based on multiplicative and bit-wise methods separately. The results of the experiment proved that the access speed to hash table practically does not change using a method or other in against what Ramakrishna says [29] —it must be due to hardware evolution. With the following hash function we have achieved an average of 1.4 accesses to the lists of collisions:

$$H(s) = \left(\left(\sum_{n=1}^l \left(s[n] * \left(b^{n-1} \bmod m \right) \right) * p1 + p2 \right) \bmod p3 \right) \bmod ltable \quad (1)$$

This means that, on this average, MAHT executes less than two comparisons to locate any word. Where s is the word to handle, $s[n]$ is the value of codification of the character that occupies the umpteenth position in the word, b represents the base of displacement to apply to each character, m is the limit of used numbers to avoid an overflow and $ltable$ is the size of the hash table. We have used three prime numbers, $p1=252551843$, $p2=84850729$ and $p3=2020251127$, to select a universal hash function [12].

Since the space on the disc is not a problem nowadays, we have not considered compacting the structure. Moreover, this improvement of the space management would not involve an increase of the speed of recognition. We have not carried out the method move-to-front hashing heuristic [37], neither the method cache-conscious resolution in string hash tables [6], since they are irrelevant for speed on an efficient hash function applied to a large hash table with very few collisions.

The 4,980,387 words stored in the database and transferred to the hash tables previously mentioned, do not include neologism with enclitic pronouns or prefixes. This addition would increase enormously the size of the database. We have decided to handle them by means of rules incorporated into the program. The rules implemented are defined in Pérez [25]. Then, for recognition, first the program searches for the word in the structure, later it examines the presence of enclitic pronouns, and repeats the search for the words obtained in this examination without pronouns; if these searches are unsuccessful, the program examines the existence of prefixes and repeats the previous searches with the words obtained by removing them; therefore, the handling of prefixes is applied only to neologisms. The loss of performance when applying these rules is compensated by the gain in recognition power.

For the handling of enclitic pronouns we have updated the rules of recognition of postponed enclitic pronouns, with respect to the ones implemented in GEISA [25],

adjusting them to the new Spanish orthography of verbal monosyllables. The new Spanish orthography [1] says that a verbal monosyllable, with a postponed enclitic pronoun, has graphic accent in accordance with the Spanish accentuation rules. Before, the verbal monosyllable kept the graphic accent after adding the enclitic pronoun —*pidió + le* as *pidiole* (~*pidióle*), *cayó + se* as *cayose* (~*cayóse*).

The rules to recognize the enclitic pronouns as well as the prefixes and prefixal elements, use individual trie, with the purpose to carry out the corresponding cuts to the non-recognized words.

5 The Enclitic Pronouns

The enclitic pronouns (*la, las, le, les, lo, los, me, nos, os, se* and *te*) belong to a word when they are added to the end of a verbal form: *mírala, llámame...* In the Spanish language, we can combine these pronouns among themselves, applying the rule of order, which establishes that *se* is in the front always, and then follow second persons, and then first persons and third persons are always last. MAHT uses a trie structure to store the enclitic pronouns in main memory. It allows looking very fast for the existence of enclitic pronouns in the words. The combination number gives rise to 101 useful possibilities in Spanish [1], for example, MAHT recognizes *comiéndosemelo* as *comiendo* (v. *comer*) + *se + me + lo*, *tráenoslo* as *trae* (v. *traer*) + *nos + lo*, *llévatela* as *lleva* (v. *llevar*) + *te + la...*

Nowadays, the use of postponed enclitics in Spanish is only frequent in the infinitive, the gerund, the imperative forms and the exhortative affirmative subjunctive [1]. If we only consider the verbal forms which can usually add postponed clitics in Spanish —about 30— and the useful combinations of enclitic pronouns —101—, the capacity of recognition of MAHT achieves to more than 42 million words —about 14,000 verbs multiplied by 101 pronoun combinations and multiplied by about 30 verbal forms—.

6 Prefixes and Prefixal Elements

The prefixation is one of the most common processes of Spanish word formation by means of addition by the left side of a prefix or a prefixal element to a word. MAHT recognizes *precálculo* as *pre-* + *cálculo*, *extrabaratado* as *extra-* + *barato*, *pseudocultura* as *pseudo-* + *cultura*, *microláser* as *micro-* + *láser...* The prefixes are added to the words regardless of their inflection and usually they do not produce changes of grammatical category; normally, they clarify, correct, modify and, finally, they guide the meaning of the word. Nevertheless, the prefixal elements do change the meaning of the word to which they are joined, in an objective or subjective way.

MAHT recognizes neologisms that have incorporated prefixes and also those that have added some of the prefixal elements which contribute a pronominal or adverbial sense to the base or whose semantic value is generic and of applicability to almost any grammar category [11]. MAHT handles the following prefixal elements: *acro-*, *aero-*, *agro-*, *ambi-*, *andro-*, *anto-*, *astro-*, *audio-*, *auto-*, *bar-*, *bati-*, *bien-*, *bio-*, *cachi-*,

cardi-, ciclo-, circa-, crio-, crono-, cuasi-, diali-, eco-, etno-, euro-, fil-, filo-, foto-, geo-, hepta-, hetero-, hidro-, homo-, infra-, intra-, kilo-, macro-, maxi-, mega-, meso-, metro-, micro-, mini-, mono-, moto-, multi-, nano-, narco-, neo-, neuro-, oligo-, omni-, pen-, penta-, per-, peri-, piro-, pluri-, plus-, poli-, por-, preter-, pseudo-, psico-, radi-, radio-, res-, servo-, seudo-, socio-, sono-, tatar-, tecno-, tetra-, topo-, tri-, turbo-, uni-, video-, xeno-. Just as in the enclitic pronouns, MAHT uses a trie structure to store the prefixes in main memory. It allows looking very fast for the existence of prefixes in the words.

If we suppose that only to the half of the words —2,000,000— it is logical to add about 60% of prefixes and prefixal elements —100—, MAHT can recognize about 200 million of words with prefixes and prefixal elements. In addition, if we consider that Spanish prefixes are combinable among themselves —*co+sub+director*—, the number of words with prefixes or prefixal elements handled by MAHT can raise 4,000 million. We suppose the combination of words only with the 10% of prefixes and prefixal elements to calculate this number. MAHT also recognizes the verbal forms with prefixes and enclitic pronouns added simultaneously —*precomiéndoselas* as *pre + comiendo* (*v. comer*) + *se + las*—.

7 Results

Before executing the performance tests of the proposed solution, we have checked that the recognition of the 4,980,381 stored words in the system work correctly. For that purpose, we carried out two tests that they finished successfully:

1. To recognize all the words and to check that the result matches with the stored morphological information in the relational database.
2. To recognize all the words and to check that the results matches with those of the Automatic Inflector and Lemmatizer of Spanish Word (AILESW) [31], which has been tested throughout more than ten years in Internet.

As other authors have expressed [5], [7], [16], [21], the need of morphological analysis is frequently linked to tasks that involve an analysis of text, beyond the study of the words of independent form. For this reason, we consider appropriate to measure the performance of MAHT analyzing two textual corpora and a small set of high quality Spanish texts:

1. A textual corpus with 291,554,063 words distributed in 9,255 Spanish texts —65% of literary texts: story, novel, theatre, poetry, and others, and 35% of non-literary texts: literature, right, politic, history, cinema, philosophy and others.
2. A corpus of Spanish journalistic texts —cultural and opinion articles— with 42,412,699 words distributed in 75,912 articles.
3. Five literary texts of the five Spanish authors who have obtained the Nobel Prize for literature from the year 1970: *Veinte poemas de amor y una canción desesperada* by Pablo Neruda, some poems by Vicente Aleixandre, *El coronel no tiene quien le escriba* by Gabriel García Márquez, *La familia de Pascual Duarte* by Camilo José Cela and *Todos santos, día de muertos* by Octavio Paz Lozano.

The analysis of texts is more a task of a Part-of-Speech Tagger (POS), since the results of an analyzer must be complemented to resolve the ambiguities and the categorization of non-recognized words—a morphological analyzer only identifies what it knows—. Ambiguities are due to that some words can correspond morphologically to more than one stem. Since our database only includes the most frequent proper nouns, plus the consolidated abbreviations and acronyms, we have developed three simple contextual rules that applied to the not recognized words; allow identifying the majority of these words. These rules analyze the location and the use of capital and lower-case letters in agreement with the academic rules [1]. MAHT doesn't solve the ambiguity that any word can have, since it is not a Part-of-Speech.

Table 1 shows the results obtained on the mentioned corpus. MAHT recognizes an average speed of 275,083 words per second run on an Intel Core 2 CPU 6600 / 2.4GHz, with 4 Gbytes of RAM memory under the Windows XP Professional operating system. The performance of morphological analysis is always superior to 99%. We have catalogued the non-recognized words in: neologisms, proper nouns—mainly foreign—, words of another language, misspelling and others. Due to that MAHT recognizes exclusively Spanish words and that less than 0.2% of the non-recognized words group would be Spanish—neologisms and some proper nouns—, MAHT's performance of morphological analysis of Spanish words is greater than 99.8%.

Table 1. Results of analyzed corpus with MAHT

	Analyzed corpus		
	Textual corpus	Journalistic corpus	Nobel Prizes
Files	9,255	75,912	5
Full recognized files (100%)	7.455%	51.507%	0.0%
Words per second	105,925	86,236	108,173
Words	291,554,063	42,412,699	67,608
Recognized words			
Recognized	99.623%	99.790%	99.972%
Recognized with enclitic pronouns	1.259%	1.196%	0.435%
Recognized with prefixes	0.054%	0.105%	0.016%
Non-recognized words (estimated percentages)			
Foreign proper nouns	0.161%	0.063%	0.001%
Non-spanish words	0.106%	0.056%	0.006%
Orthographic errors	0.073%	0.046%	0.000%
Neologisms	0.033%	0.043%	0.021%
Others	0.004%	0.002%	0.000%

In table 1, the distribution percentages of non-recognized words are estimated, due to the evident difficulty to catalog them. We have considered neologism any well-written Spanish word that has not been recognized by MAHT. In this group, we have

catalogued besides the neologisms by derivation or prefixation —*amantísimo* (32¹), *aerocicleta* (0)—, specialized words —*mantra* (140)— and science-fiction words —*élfica* (1), *nesghais* (0)—. The group *Others* includes words of difficult cataloging and expressions of the type: *aaahhh!*, *verdaaaaaad* —literary versions of *ah!* and *verdad* used by the authors with the purpose to emphasize.

The non-recognized words are mainly, foreign proper nouns, words of another language and orthographic errors. Nevertheless, in the texts of authors of great prestige, the neologisms are the most frequent, although with a residual percentage —0.021%— with respect to the total number of words. Table 2 shows the more frequent non-recognized words for each one of the processed corpus. We have sought in corpus of current Spanish (CREA²) the neologisms of the analyzed Nobel prizes texts —we remark between parenthesis the appearance frequency— and we observe that these neologisms are words of almost unitary and exclusive use by those authors.

Table 2. The most frequent non-recognized words in the analyzed corpus with MAHT

Analyzed corpus								
Textual corpus			Journalistic corpus			Nobel Prizes		
Word	Type	Frec.	Word	Type	Frec.	Word	Type	Frec.
<i>the</i>	Other lang.	4835	<i>ó</i>	Archaism	654	<i>murder</i>	Other lang.	2(0)
<i>fué</i>	Archaism	4548	<i>the</i>	Other lang.	430	<i>week</i>	Other lang.	1(0)
<i>madame</i>	Other lang.	3603	<i>and</i>	Other lang.	380	<i>tronal</i>	Neologism	1(2)
<i>and</i>	Other lang.	3506	<i>von</i>	Other lang.	296	<i>sermoncete</i>	Neologism	1(0)
<i>Ayla</i>	Noun	3417	<i>ésto</i>	Error	260	<i>seroformo</i>	Neologism	1(1)
<i>usté</i>	Archaism	2441	<i>angel</i>	Error	253	<i>instantero</i>	Neologism	1(0)
<i>habemos</i>	Archaism	2395	<i>dió</i>	Archaism	245	<i>cavadora</i>	Neologism	1(1)
<i>John</i>	Noun	2382	<i>du</i>	Other lang.	188	<i>giralunas</i>	Neologism	1(0)
<i>destos</i>	Archaism	2270	<i>Cruyff</i>	Noun	158	<i>porlan</i>	Neologism	1(0)
<i>du</i>	Other lang.	2152	<i>John</i>	Noun	150	<i>Estirao</i>	Noun	1(1)

Of the three analyzers (AILESW [31], Sedláček [32] and Sgarbas [33]), Sedláček obtains better results, which has been evaluated with a corpus of 100,000,000 words of which, according to its authors, only stops recognizing a small percentage mainly formed by words of another language, and is capable of analyzing 20,000 words per second using a Pentium III processor of 800MHz. However, these 20,000 words per second, still considering the difference of benefits of the used processor, are less than the 275,083 that MAHT can recognize, including the recognition of words with prefixes and enclitics by means of rules. Nevertheless, the papers of Sedláček [32] do not mention the enclitic and only include three prefixes (*nej+ne*, *nej*, *ne*).

¹ Word frequency in the reference corpus of current Spanish (CREA²). It proves the negligible use of these words in Spanish texts.

² CREA: Reference corpus of current Spanish made by the Spanish Royal Academy. CREA is composed by 170 million of words coming from a large variety of written and oral texts produced in all countries of Hispanic speech from 1975 to current time (<http://corpus.rae.es/creanet.html>).

8 Conclusions

This article has presented a Spanish morphological analyzer (MAHT), which reaches a high performance in speed, based on mainly in the storage of words and all the morphological information associated to each of them. This approach turns out to be especially useful in an irregular and highly inflectional language, like Spanish.

MAHT recognizes at an average speed of 275,083 words per second, because the information is stored into hash tables in the main memory. MAHT has a very high performance of morphological recognizing for Spanish language —over 99.8%—. It uses a lexical knowledge database with all morphological information of 4,980,387 Spanish words. The proposed solution frees the data with respect to the analysis algorithms when handling the irregular forms, which remarkably simplifies the insertion of new words inside the database and the maintenance of the program code.

For the processing of prefixes and enclitic pronouns we have implemented easy rules, to avoid storing the large number of potential words with enclitic and prefixes. The small loss of inherent performance to the implementation of these rules is highly compensated for the wide range of recognized words.

The tests demonstrate that with almost five million words in the database, MAHT practically covers the whole morphological casuistry of the Spanish language.

The system can improve with the addition of new words to the database: some conjugated verbal forms with enclitic pronouns, some frequent derivations and some neologisms with prefixes. These additions would increase the recognition speed of texts, since the application of the rules of pronouns and prefixes decreases.

Acknowledgements. The research that appears in this paper is framed partially within the project "Caracterización Objetiva de la Dificultad General de los Originales" (CÓDIGO) (Project ID: FFI2010-15724, National Plan I+D+i).

References

1. Academia Española de la Lengua: Ortografía de la Lengua Española. Espasa Calpe, Madrid (1999)
2. Alsina, R.: Todos los Verbos Castellanos Conjugados, 17th edn. Teide, Barcelona (1990)
3. Alvar Ezquerro, M.: Diccionario de voces de uso actual. Arco/Libros, Madrid (1994)
4. Antoshkov, G., Ziauddin, M.: Query processing and optimization in Oracle Rdb. *The International Journal on Very Large Data Bases* 54, 229–237 (1996)
5. Appelt, D.E., Israel, D.J.: Introduction to information extraction technology. In: *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI 1999*, Tutorial, Stockholm (1999)
6. Askitis, N., Zobel, J.: Cache-Conscious Collision Resolution in String Hash Tables. In: Consens, M., Navarro, G. (eds.) *SPIRE 2005*. LNCS, vol. 3772, pp. 91–102. Springer, Heidelberg (2005)
7. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison-Wesley Longman Publishing Company, Boston (1999)

8. Baldzis, S., Kolalas, S., Eumeridou, E.: The Computational Modern Greek Morphological Lexicon —An Efficient and Comprehensive System for Morphological Analysis and Synthesis. *Literary and Linguistic Computing* 202, 153–187 (2005)
9. Bibliograf (ed.): *Diccionario General de la Lengua Española Vox*, Electronic edn. Bibliograf, Barcelona (1997)
10. Byrne, W., Hajič, J., Ircing, P., Krbeč, P., Psutka, J.: Morpheme Based Language Models for Speech Recognition of Czech. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *TSD 2000. LNCS (LNAI)*, vol. 1902, pp. 211–216. Springer, Heidelberg (2000)
11. Carreras, F.J.: *Sistema Computacional de Gestión Morfológica del Español SCOGEME*. PhD Thesis. Las Palmas de Gran Canaria: Universidad de Las Palmas de Gran Canaria, Spain (2002)
12. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. *Journal Computer and System Sciences* 18, 143–154 (1979)
13. Casares, J.: *Diccionario Ideológico de la Lengua Española*, 2nd edn. Gustavo Gili, Barcelona (1990)
14. Clave: *Diccionario de Uso del Español Actual*. Electronic edn. Clave S.M, Madrid (1997)
15. Daciuk, J., Watson, R.E., Watson, B.: Incremental construction of acyclic finite-state automata and transducers. In: *Proceedings of Finite State Methods in Natural Language Processing*. Bilkent University, Ankara (1998)
16. Erjavec, T., Džeroski, S.: Machine Learning of Morphosyntactic Structure: Lemmatizing Unknown Slovene Words. *Applied Artificial Intelligence* 181, 17–41 (2004)
17. Espasa Calpe (ed.): *Gran Diccionario de Sinónimos y Antónimos*, 4th edn. Espasa Calpe, Madrid (1991)
18. Horowitz, E., Sahni, S.: *Fundamentals of Data Structures*. Pitman Publishing Limited, London (1977)
19. Koskenniemi, K.: Two-level Model for Morphological Analysis'. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 8–12. Karlsruhe, West Germany (1983)
20. Larousse (ed.): *Gran Diccionario de la Lengua Española*. Larousse Planeta, Barcelona (1996)
21. Mani, I., Maybury, M.T. (eds.): *Advances in Automatic Text Summarization*. MIT Press (1999)
22. Minnen, G., Carroll, J., Pearce, D.: Applied morphological processing of English. *Natural Language Engineering* 73, 225–250 (2001)
23. Moliner, M.: *Diccionario de Uso del Español de María Moliner*, 2nd electronic edn. Gredos, Madrid (1996)
24. Papakitsos, E., Grigoriadou, M., Philokyprou, G.: Modelling a Morpheme based Lexicon for Modern Greek. *Literary and Linguistic Computing* 174, 475–490 (2002)
25. Pérez, J.R.: *Reconocimiento y generación integrada de la morfología del español: Una aplicación a la gestión de un diccionario de sinónimos y antónimos*. PhD thesis. Las Palmas de Gran Canaria: Universidad de Las Palmas de Gran Canaria (1996)
26. Polguère, A.: Towards a theoretically-motivated general public dictionary of semantic derivations and collocations for French. In: *Proceedings of EURALEX 2000*, Stuttgart, pp. 517–528 (2000)
27. Prószyky, G.: Industrial Applications of Unification Morphology. In: *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, pp. 213–214 (1994)

28. Prószyński, 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, Maryland, pp. 261–268 (1999)
29. Ramakrishna, M.V., Zobel, J.: Performance in practice of string hashing functions. In: Proceedings of the International Conference on Database Systems for Advanced Applications, pp. 215–223 (1997)
30. Real Academia Española (ed.): Diccionario de la Real Academia Española, Electronic edn. 21.1.0. Real Academia Española and Espasa Calpe, Madrid (1995)
31. Santana, O., Pérez, J., Carreras, F., Hernández, Z., Rodríguez, G.: The Spanish Morphology in Internet. In: Cueva Lovelle, J.M., Rodríguez, B.M.G., Gayo, J.E.L., del Puerto Paule Ruiz, M., Aguilar, L.J. (eds.) ICWE 2003. LNCS, vol. 2722, pp. 507–510. Springer, Heidelberg (2003)
32. Sedláček, R., Smrž, P.: Automatic Processing of Czech Inflectional and Derivative Morphology. FI MU Report Series. Faculty of Informatics, Masaryk University (2001)
33. Sgarbas, K.N., Fakotakis, N.D., Kokkinakis, G.K.: A Straightforward Approach to Morphological Analysis and Synthesis. In: Proceedings of COMLEX 2000, Workshop on Computational Lexicography and Multimedia Dictionaries, Kato Achaia, Greece, pp. 31–34 (2000)
34. Sproat, R.: Morphology and Computation. MIT Press, Cambridge (1992)
35. Velásquez, F., Gelbukh, A., Sidorov, G.: AGME: un sistema de análisis y generación de la morfología del español. In: Proceedings of Workshop Multilingual Information Access and Natural Language Processing of IBERAMIA 2002 (8th Iberoamerican Conference on Artificial Intelligence), pp. 1–6 (2002)
36. Villena, J., González, J.C., González, B.: STILUS: Sistema de revisión lingüística de textos en castellano. *Procesamiento del Lenguaje Natural* 29, 305–306 (2002)
37. Zobel, J., Heinz, S., Williams, H.: In memory hash tables for accumulating text vocabularies. *Information Processing Letters* 80(6), 271–277 (2001)

Optimal Stem Identification in Presence of Suffix List

N. Vasudevan and Pushpak Bhattacharyya

Computer Science and Engg Department
IIT Bombay, Mumbai
{vasudevan,pb}@cse.iitb.ac.in

Abstract Stemming is considered crucial in many NLP and IR applications. In the absence of any linguistic information, stemming is a challenging task. Stemming of words using suffixes of a language as linguistic information is in comparison an easier problem. In this work we considered stemming as a process of obtaining minimum number of lexicon from an unannotated corpus by using a suffix set. We proved that the exact lexicon reduction problem is NP-hard and came up with a polynomial time approximation. One probabilistic model that minimizes the stem distributional entropy is also proposed for stemming. Performances of these models are analyzed using an unannotated corpus and a suffix set of Malayalam, a morphologically rich language of India belonging to the Dravidian family.

1 Introduction

Stemming is a crucial component in most of the NLP applications. Since the stemming identifies the same stem for all inflectional variants of a lexeme, it will improve the performance of information retrieval systems. Inflectional suffix in a word carries its morphosyntactic information and paradigm information. For example, by stemming the word **boys**, we will get the suffix **s** from that word. This suffix carries the information about its plurality. Such information is essential for many NLP applications like machine translation. This indicates the importance of building good stemmer for languages.

Building a morphology analyser or a stemmer is always a challenging task for all languages. This is more challenging for inflectional, agglutinative and isolated languages. Lot of linguistic expertise is needed for building such tools. This is again difficult for those languages which have no linguistic tradition. The linguistic expertise needed for stemming may not be available for such languages. So they have to rely on word forms in a corpus and their processing. Therefore building a low cost automatic stemmer from a corpus for such languages has a great significance.

Building a stemmer using only an unannotated corpus is the most inexpensive feasible approach. Since there is no availability of direct linguistic information and supervision, such a system with good performance is very difficult to build.

Knowledge of inflectional suffixes in a language will reduce the level of difficulty. I.e. the performance of stemming using only an unannotated corpus can be improved by using a set of inflectional suffixes. Usually a language has a small closed set of inflectional suffixes. Identification of this suffix set of a language is a relatively easier job. So we consider a semi-supervised scenario where the suffix list is given. In this work we are trying to build a stemmer using an unannotated corpus with sufficiently large number of distinct words and a set of inflectional suffixes. Sample sets of inflectional suffixes in English and Malayalam are shown below (Example 1, 2)

Example 1. English: { s, es, ed, ing, ... }

Example 2. Malayalam: { കൾ {kal} (Plural), കളെ {kale} (Plural + Accusative), കളുടെ {kalude} (Plural + Genitive), മാർ {mar} (Plural), മാറെ {mare} (Plural + Accusative), മാരുടെ {marude} (Plural + Genitive), ുക {uka} (Present), കുക {kkuka} (Present), ി {i} (Past), ും {um} (Future), ... }

Inputs of this stemming problem are an unannotated corpus (*Corpus*) and a suffix set (*Suffixset*), and the output is the stem of each word. A small input and its output is shown in Example 3.

Example 3. Input: *Corpus* = { mosses, moss, boys, boy }, *Suffixset* = { es, s, ϕ }, where ϕ is the null suffix.

Output: { moss (mosses), moss (moss), boy (boys), boy (boy) }

We make an assumption that *Suffixset* includes all orthographic variations of all valid suffixes in the language. കൾ {kal}, കൾ {kka} and ങ്ങൾ {ngngal} are the variants of Malayalam¹ plural marker കൾ {kal}. So a Malayalam *Suffixset* contains all these variants. Some languages have only one possible suffix in a word, while others have more. In such languages a word has the structure *stem.suffix₁.suffix₂...suffix_x*. For example, a Malayalam word കുട്ടികളുടെ {kuttikalude} (child+Plural+Genitive) contains the stem കുട്ടി {kutti} (child), a plural marker കൾ {kal} and a genitive case marker ുടെ {ude}. Concatenated sequence of suffixes, *suffix₁.suffix₂...suffix_x* is considered as a suffix in such words. So *Suffixset* of such a language contains some concatenations of suffixes also. In the previous example of Malayalam word കുട്ടികളുടെ {kuttikalude}, കളുടെ {kalude} is actually a concatenation of suffixes കൾ {kal} and ുടെ {ude}. But we consider this as a single suffix. We take another assumption that the *Corpus* contains sufficiently large number of words.

To define the stemming problem with suffix set, let us introduce a term *possible_stem_set*. *possible_stem_set* of a word *w* be the set of all prefixes of *w* such that *w* can be expressed as the concatenation of that prefix and a suffix from *Suffixset*.

$$possible_stem_set(w) = \{st : \exists x \in Suffixset \text{ such that } w = st.x \}$$

Consider the input shown in Example 3. *possible_stem_set* of each word in the *Corpus* is shown in the Table 1. Now formally we can say, stemming is the

¹ A morphologically rich language of India belonging to the Dravidian family.

Table 1. *possible_stem_set* and their Correct Stem

word	<i>possible_stem_set</i>	correct stem
<i>mosses</i>	{ <i>mosses, mosse, moss</i> }	<i>moss</i>
<i>moss</i>	{ <i>moss, mos</i> }	<i>moss</i>
<i>boys</i>	{ <i>boy, boys</i> }	<i>boy</i>
<i>boy</i>	{ <i>boy</i> }	<i>boy</i>

process of identifying $stem(w)$ optimally for all word w , where $stem(w)$ is an element of $possible_stem_set(w)$.

The stemming process is the selection of correct entry from $possible_stem_set$ of each word in *Corpus*. In the previous example, the stemmer needs to select the stems **moss** from $possible_stem_set(\mathbf{mosses})$, **moss** from $possible_stem_set(\mathbf{moss})$, **boy** from $possible_stem_set(\mathbf{boys})$ and **boy** from $possible_stem_set(\mathbf{boy})$.

The roadmap of the paper is as follows. Related work of the stemming problem is briefly summarized in section 2. We propose two models for stemming problem in this work. In section 3, we propose a deterministic model that reduces the number of distinct stems. In section 4, we propose a probabilistic model that learns the distribution by reducing the entropy. A case study of these models in one of the morphologically rich language Malayalam is included in section 5. Performances of these models are measured in this section by using a wordlist and a suffix set.

2 Related Work

Morphology learning is one of the widely attempted problem in the literature. A recent survey paper by Harald Hammarstrom [1] gives an overall view on unsupervised morphology learning. There are lots of probabilistic approaches for morphology learning. Linguistica model [2], maximum a posteriori model [3], stochastic transducer based model [4] and generative probabilistic model [5] are the relevant probabilistic models for stemming that we found in the current literature. A Markov Random Field by Dreyer [6] is also a useful work related to unsupervised morphology.

Graph based model [7], lazy learning based model [8], clustering based same stem identification model [9,10] ParaMor system for paradigm learning [11] are also relevant works in the same area. Full morpheme segmentation and automatic induction of orthographic rules by Sajib Dasgupta [12,13] is also a relevant work.

We never found any model which use the information from suffix list. To the best of our knowledge, this is the first attempt for stemming in presence of suffix list. We found that Linguistica model is the closely related approach to ours. Frequency of stem and suffix candidates plays a crucial role in Linguistica model. Linguistica model is an optimal stem identification model by reducing the description length. In this work, we minimizes the number of distinct stems and entropy of stem distribution.

Table 2. Examples of *valid_mapping*

<i>valid_mapping</i> (fword \rightarrow stem)	range(f)	range(f)
{(mosses \rightarrow mosses), (moss \rightarrow moss), (boys \rightarrow boys), (boy \rightarrow boy)}	{mosses, moss, boys, boy}	4
{(mosses \rightarrow mosses), (moss \rightarrow moss), (boys \rightarrow boy), (boy \rightarrow boy)}	{mosses, moss, boy}	3
{(mosses \rightarrow mosses), (moss \rightarrow mos), (boys \rightarrow boys), (boy \rightarrow boy)}	{mosses, mos, boys, boy}	4
{(mosses \rightarrow mosses), (moss \rightarrow mos), (boys \rightarrow boy), (boy \rightarrow boy)}	{mosses, mos, boy}	3
{(mosses \rightarrow mosse), (moss \rightarrow moss), (boys \rightarrow boys), (boy \rightarrow boy)}	{mosse, moss, boys, boy}	4
{(mosses \rightarrow mosse), (moss \rightarrow moss), (boys \rightarrow boy), (boy \rightarrow boy)}	{mosse, moss, boy}	3
{(mosses \rightarrow mosse), (moss \rightarrow mos), (boys \rightarrow boys), (boy \rightarrow boy)}	{mosse, mos, boys, boy}	4
{(mosses \rightarrow mosse), (moss \rightarrow mos), (boys \rightarrow boy), (boy \rightarrow boy)}	{mosse, mos, boy}	3
{(mosses \rightarrow moss), (moss \rightarrow moss), (boys \rightarrow boys), (boy \rightarrow boy)}	{moss, boys, boy}	3
{(mosses \rightarrow moss), (moss \rightarrow moss), (boys \rightarrow boy), (boy \rightarrow boy)}	{moss, boy}	2
{(mosses \rightarrow moss), (moss \rightarrow mos), (boys \rightarrow boys), (boy \rightarrow boy)}	{moss, mos, boys, boy}	4
{(mosses \rightarrow moss), (moss \rightarrow mos), (boys \rightarrow boy), (boy \rightarrow boy)}	{moss, mos, boy}	3

3 Minimum Stem Range (MSR) model

Given the suffix set, stemming can be viewed as a process of reduction of lexicon entry. Minimum Stem Range (MSR) model is a direct and intuitive translation of this aspect of stemming problem to a well-defined computational model. MSR model finds out a mapping from each word to one of the string in its *possible_stem_set*. If suffix set is complete then actual stem of each word should be in *possible_stem_set* of that word. So, if *possible_stem_set* of a word contains exactly one entry then any mapping identifies the actual stem of that word. In the Table 1, *possible_stem_set* of **boy** contains only one element **boy**. So any mapping to *possible_stem_set* choose the correct stem **boy** from *possible_stem_set*(**boy**). Otherwise the actual stem needs to be identified by using the information from other words.

3.1 Model

MSR model is a computational model for the stemming problem. This model finds a mapping from each word in input corpus to its stem (a starting substring of the word). A mapping function f from each word in input corpus to its stem (a starting substring of the word) is called as a *valid_mapping* if and only if each word in input corpus is gets mapped to one of the stems in its *possible_stem_set*. All *valid_mappings* from Example 3 are shown in Table 2 (*possible_stem_sets* of this sample corpus is shown in Table 1). A mapping (**mosse**, **mos**, **boys**, **boy**) means, first word **mosses** is gets mapped to **mosse**, second word **moss** is gets mapped to **mos**, **boys** is gets mapped to **boys** and **boy** is gets mapped to **boy**.

MSR model will find out a *valid_mapping* with minimum range. Lets define the MSR model formally.

Input: *Corpus* (a sufficiently large list of plain words) and *Suffixset* (set of all suffixes)

Output: A *valid_mapping*, f^* with minimum cardinality of range,

$$f^* = \underset{f \in \text{valid_mapping}}{\operatorname{argmin}} \left\{ |\text{range}(f)| \right\}$$

Out of these 12 *valid_mappings* shown in Table 2, (*moss, moss, boy, boy*) have the minimum range. So the MSR model will identify this mapping.

Theorem 1. *If a language does not have any morphological ambiguity and MSR model identifies a valid stem for at least one word from all group of words with same stem then MSR model will identify the correct stem from all words.*

Proof. Lets assume there is no morphological ambiguity. So there will be exactly one split that generates a valid stem and a valid suffix. If one of the valid stem is in *possible_stem_set* of a word then that will be the correct stem of that word. Otherwise there will be more than one way to split that word and that will be a morphological ambiguity. So if MSR problem identifies a valid stem from a word then that will be the correct stem of that word.

Suppose there are m groups of words with the same stem. There exist a *valid_mapping* with range of size exactly m by correctly identifying all correct stems. So the range of output of MSR model will be less than or equal to m .

Let f be the output of MSR model that identifies correct stem from at least one word from all m groups. So, all valid stems from all words in input corpus will be in the range of f . Since there exist exactly one valid stem in each *possible_stem_set*, any incorrect *valid_mapping* produces an invalid stem. So if there is any incorrect mapping, then the range of f will be greater than m . If there is any incorrect mapping in f then f will not be the output of MSR model. Therefore the MSR model will identify the correct stem from all words.

Theorem 2. *Problem of computing MSR model (MSR problem) is NP-Hard.*

Proof. To prove the MSR problem is NP-hard, we want to find a polynomial time reduction from an existing NP-hard problem to MSR problem. Let us take minimum vertex cover problem, a known NP-hard problem for the reduction. Input of minimum vertex cover problem is an undirected graph $G = (V, E)$ and output is a set of vertices (vertex cover) VC with minimum cardinality, where $VC \subseteq V$ and for every edge e_{ij} in E , either $v_i \in VC$ or $v_j \in VC$. Given an instance of vertex cover problem ($G = (V, E)$), we construct an instance to MSR problem (*Corpus, Suffixset*) as follows.

Suppose $G = (V, E)$, $V = \{v_i\}$, $E = \{e_{ij}\}$ $1 \leq i, j \leq n$ be the input of vertex cover problem. For every edge e_{ij} , without losing generality we can say $i \leq j$. For each edge e_{ij} add a word $c_1.c_2 \dots c_j.m_{ij}$ in to the *Corpus*. Here $c_1, c_2, \dots c_j$ and m_{ij} can be any distinct characters. For each edge e_{ij} add $c_{i+1} \dots c_j.m_{ij}$ and m_{ij} in to the *Suffixset* (if $i = j$ then add only m_{ij} in to the *Suffixset*).

Consider a small graph shown below. Corresponds to four edges ($e_{12}, e_{13}, e_{23}, e_{34}$) add four words to the *Corpus*. Similarly corresponds to four edges add eight suffixes to the *Suffixset*. These words and suffixes are shown in the second and third columns of Table 3.

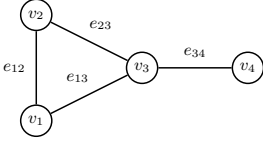


Fig. 1. Example Graph to Elucidate Reduction Procedure

Table 3. Reduction Example of the Graph

	word list	suffix list	possible_stem_set	f	VC
e_{12}	$c_1 c_2 m_{12}$	$c_2 m_{12}, m_{12}$	$\{c_1, c_1 c_2\}$	c_1	v_1
e_{13}	$c_1 c_2 c_3 m_{13}$	$c_2 c_3 m_{13}, m_{13}$	$\{c_1, c_1 c_2 c_3\}$	c_1	v_1
e_{23}	$c_1 c_2 c_3 m_{23}$	$c_3 m_{23}, m_{23}$	$\{c_1 c_2, c_1 c_2 c_3\}$	$c_1 c_2 c_3$	v_3
e_{34}	$c_1 c_2 c_3 c_4 m_{34}$	$c_4 m_{34}, m_{34}$	$\{c_1 c_2 c_3, c_1 c_2 c_3 c_4\}$	$c_1 c_2 c_3$	v_3

Let f be the output of MSR problem and R is the range of f . Since all c_i and m_{ij} are distinct, $possible_stem_set(c_1 \dots c_j . m_{ij})$ will be $\{c_1 \dots c_j, c_1 \dots c_i\}$. See the $possible_stem_set$ of each word in the previous example shown in Table 3. So each string in R will be in the form of $c_1 \dots c_k$ for some k . For each such string, add the vertex v_k in to VC . Here the cardinality of R and VC will be same. In the example the corresponding vertex cover (VC) of f function is a minimal vertex cover ($\{v_1, v_3\}$).

To prove the correctness of reduction we want to show the solution of MSR problem will be same as the solution of minimum vertex cover problem.

Let say f^* is the solution of MSR problem and VC^* is the corresponding output instance of minimum vertex cover problem. For each word $c_1 \dots c_j . m_{ij}$ either $c_1 \dots c_j$ or $c_1 \dots c_i$ will be in the range of any $valid_mapping$. Therefore VC^* will be a valid vertex cover and that can be the solution of minimum vertex cover problem. Suppose VC^* is not the solution of minimum vertex cover problem, then there exist another valid vertex cover VC' , such that $|VC'| < |VC^*|$. Then for VC' we can define a $valid_mapping$, f' ,

$$f'(c_1 c_2 \dots c_j m_{ij}) = \begin{cases} c_1 c_2 \dots c_i & \text{if } v_i \text{ is in } VC' \\ c_1 c_2 \dots c_j & \text{otherwise} \end{cases}$$

Here we can see $|range(f')| \leq |VC'|$

$$\Rightarrow |range(f')| < |VC^*|$$

$$\Rightarrow |range(f')| < |range(f^*)|$$

By the definition of f^* this is a contradiction. Therefore VC^* always will be the solution of minimum vertex cover problem.

Let VC^* is the solution of minimum vertex cover problem. Now we can define a corresponding $valid_mapping$ f^* for VC^* ,

$$f^*(c_1 c_2 \dots c_j m_{ij}) = \begin{cases} c_1 c_2 \dots c_i & \text{if } v_i \text{ is in } VC^* \\ c_1 c_2 \dots c_j & \text{otherwise} \end{cases}$$

Here $|VC^*| \geq |range(f^*)|$. If $|VC^*| > |range(f^*)|$ then there exist another vertex cover with less cardinality. But VC^* is the minimum vertex cover. So $|range(f^*)|$ will be same as $|VC^*|$.

Suppose f^* is not the solution of MSR problem, then there will be another *valid_mapping* f' such that $|range(f')| < |range(f^*)|$. Then there will be a valid vertex cover VC' corresponds to f' , s.t. $|range(f')| = |VC'|$

$$\begin{aligned} \Rightarrow |VC'| &< |range(f^*)| \\ \Rightarrow |VC'| &< |VC^*| \end{aligned}$$

But VC^* is the minimum vertex cover. So such a valid vertex cover does not exist. Therefore f^* will be a solution of MSR problem.

3.2 Approximation

Since the MSR problem is NP-hard, it is difficult to find stems from a large data. In order to solve the stemming problem computationally, an approximation of the above model is required. Similarity of this model to set cover problem can be utilized to find an approximation algorithm. MSR problem can be reduced to set cover problem. Input of set cover problem is a set U and a set of subsets (S) of U such that, $\bigcup_{s \in S} s = U$. Output of set cover problem is a subset of S , say C , such that $\bigcup_{s \in C} s = U$.

Let St be the set of all possible stems, i.e. $St = \bigcup_{w \in Corpus} possible_stem_set(w)$. Let $W(st)$ is the set of words in *Corpus* where st is an element of its *possible_stem_set*, i.e. $W(st) = \{w : st \in possible_stem_set(w)\}$. Take the sample input corpus **{mosses, moss, boys, boy}** and input suffix set **{es, s, ϕ }**. Set of all possible stems and its corresponding $W(st)$ are shown in the Table 4. Now consider the *Corpus* as U and the set $\{W(st)\}$ as S of set cover problem. Here $\bigcup_{st \in St} W(st) = Corpus$. The output of this set cover problem is a set of stems St' such that, for any word in *Corpus*, there exist at least one possible stem of that word in St' . By choosing any one possible stem of each word from St' , we can find a solution of MSR problem.

Table 4. Possible Stems and their $W()$ Set

Stem	mosses	mosse	moss	mos	boys	boy
$W(stem)$ (S)	{mosses}	{mosses, moss}	{moss}	{moss}	{boys}	{boys, boy}

Any approximation algorithm of set cover problem can be directly used for this stemming problem. Best-known approximation of set cover problem is the greedy algorithm. The greedy algorithm choose subsets one by one from S that have maximum number of uncovered elements in U , and cover those uncovered elements. The complexity of approximation algorithm is $O(NM)$ and approximation factor is $\log(N)$, where N is the number of words in corpus and M is the number of suffixes in suffix set.

4 Minimum Stem Entropy (MSE) Model

The MSR model and its approximation are deterministic approaches for the stemming problem. Greedy approximation of the above model may not find optimum mapping. Here we propose a new model, Minimum Stem Entropy (MSE) model, which is a probabilistic approach for the stemming problem.

4.1 Model

The probabilistic model assumes an uncertainty to choose the correct stem from *possible_stem_set* of each word. Input of MSE model is same as that of MSR model. Output of this model is a conditional probability distribution of stem given word ($Pr(st|w)$). $Pr(st|w)$ is the probability that string st (from possible stem set of w , $PSS(w)$) is the stem of word w . From the $Pr(st|w)$ we can select the maximum probable stem candidate as the correct stem of that word, i.e. stem of a word w , $Stem(w) = argmax_{st} \{Pr(st|w)\}$. Two basic conditions for $Pr(st|w)$ are,

$$\begin{aligned} Pr(st|w) &\geq 0 \text{ for all } st \text{ in possible stem set of } w \\ \sum_{st \in PSS(w)} \{Pr(st|w)\} &= 1 \end{aligned}$$

Many such probability distributions are possible. Most suitable distribution based on input corpus needs to be learned. The MSE model selects the distribution that minimizes the entropy of stem distribution. A stem distribution $Pr(st)$ is the probability of st is to be identified as correct stem of a randomly chosen word. We can write the stem distribution in terms of $Pr(st|w)$ as,

$$Pr(st) = \sum_{w \in corpus} \{Pr(w) \times Pr(st|w)\}$$

The MSE problem can be written as an optimization problem. I.e.

$$\begin{aligned} Pr(st|w) &= argmin_{Pr(st|w)} \left\{ Entropy(Pr(st)) \right\} \\ &= argmin_{Pr(st|w)} \left\{ Entropy \left(\sum_{w \in corpus} \{Pr(w) \times Pr(st|w)\} \right) \right\} \\ &= argmin_{Pr(st|w)} \left\{ - \sum_{st} \left\{ \sum_w \{Pr(st|w) \times Pr(w)\} \times \log \left(\sum_w \{Pr(st|w) \times Pr(w)\} \right) \right\} \right\} \end{aligned}$$

MSE learns the probability of possible stem of each word that minimize the entropy of complete stem distribution. A stem distribution with low entropy has a tendency for a small stem set. So this model has a similarity with MSR model. With minimum stem distributional entropy, probability of a string is to be a correct stem in a randomly chosen word is either low or high. I.e. a string can easily classify in to a valid stem class or invalid stem class. So the process tries to learn the information about stems. The learning process identify the stem distribution such that, maximum information about stem is available from the input corpus. The formal definition of this model is shown below.

Input: *Corpus* (a sufficiently large list of plain words) and *Suffixset* (set of all suffixes)

Output: $Pr(st|w)$ for all w in *Corpus* and st in *possible_stem_set(w)*

Consider a small corpus {**mosses**, **moss**} and a suffix set {**es**, **s**, ϕ }, where ϕ is the null suffix. The *possible_stem_set* of **mosses** is {**mosses**, **mosse**, **moss**} and *possible_stem_set* of **moss** is {**moss**, **mos**}. Here the word probability can be taken as uniform distribution. So,

$$Pr(\mathbf{mosses}) = \frac{1}{2} \times Pr(\mathbf{mosses}|\mathbf{mosses})$$

$$Pr(\mathbf{mosse}) = \frac{1}{2} \times Pr(\mathbf{mosse}|\mathbf{mosses})$$

$$Pr(\mathbf{moss}) = \frac{1}{2} \times (Pr(\mathbf{moss}|\mathbf{mosses}) + Pr(\mathbf{moss}|\mathbf{moss}))$$

$$Pr(\mathbf{mos}) = \frac{1}{2} \times Pr(\mathbf{mos}|\mathbf{moss})$$

In this case, for $Pr(\mathbf{moss}|\mathbf{mosses}) = 1$ and $Pr(\mathbf{moss}|\mathbf{moss}) = 1$, will get a zero stem distributional entropy. So the MSE will converge to this distribution, and the string **moss** will be selected as the stem of both **mosses** and **moss**.

4.2 Methodology

To learn the model from a corpus and a suffix set, an optimization problem needs to be solved. Since the objective function is not convex, an iterative hill climbing like approach is used. We used the Frank-Wolfe algorithm [14] to solve the optimization.

In each iteration of Frank-Wolfe algorithm, we solved a linear program. By converging to local optima, the algorithm will find out the best probability distribution given the input corpus and suffix set. Most probable stem from each word in corpus is then identified.

5 Case Study - Malayalam

Malayalam is a Dravidian language spoken by 32 million people primarily in Kerala, a state in southern India [15]. Malayalam is highly agglutinative and inflectionally rich with a free word order. This language has a strong postpositional inflection. A Malayalam noun can be inflected for case, number, person and gender, e.g. വേലക്കാരന്മാരുടെ {velakkaaranmaarude} (worker+Masculine+Plural+Genitive). Verb can be inflected by suffix for mood, aspect and tense, e.g. പറഞ്ഞു {paranju} (told), പറയാം {parayam} (may tell), പറഞ്ഞിരിക്കും {paranjirikkum} (will tell).

Approximated MSR model and MSE model are evaluated on Malayalam. We used a Malayalam unannotated corpus of size around 20000 words and a suffix set of around 200 Malayalam suffixes. We are extracted these words from the web. Malayalam is used by less than 0.1% of all the websites. Our assumption is that, the wordlist is sufficiently large so that it contains all morphological variants of words. In practical case this assumption may not hold. To reduce the gap between the ideal case and the real case, we make sure that, the training wordlist contains at least one more inflectional variant for each word. So we manually added such inflectional variants to wordlist if necessary.

We used the Morfessor [3] system to get a ballpark accuracy. We trained the Morfessor 1.0 model using their script downloaded from www.cis.hut.fi/projects/morpho/morfessor1.0.perl with default arguments. The Morfessor score is around 17% only. Since Morfessor is not using any information from suffix set, a direct comparison between the Morfessor score and our scores is meaningless. Since we are unable to find any other comparable approaches for this problem from the literature, we want to constructed baseline models to check the significance of these proposed models.

The problem is about selecting correct entry from possible stem set of each word. We considered different trivial strategies to select the stem from possible stem set. One basic information is that a stem is a prefix of the word by stripping some valid suffix. One trivial approach that uses only this basic information is a random selection from possible stem set. We considered this as one of the baselines. Length of the suffix (or stem) is another easily available information that can use for stemming. Based on this information we can build two simple strategies, smallest stem or largest stem. We want to choose one of these approaches. Since we are doing experimentation on data, we decided to consider both approaches for experimentation and decide based on the scores. So we considered these two approaches as second and third baselines. Performances of two new proposed models and these three baseline models are evaluated using around 1500 Malayalam Unicode words from wordlist. The accuracies are shown in the Table 5.

Table 5. Stemming Accuracies and Comparison with Baseline

Model	Baseline-1 (Min Stem length)	Baseline-2 (Max Stem length)	Baseline-3 (Random selection)	Approx. MSR	MSE
Accuracies (20K words)	84.58 %	20.09 %	44.20 %	91.32 %	93.91 %

The results shown in the above table indicates the correctness and significance of proposed models for Malayalam. It also shows that MSE model slightly outperforms the Approx-MSR model. From the scores it is clear that, choosing the stem with minimum length is more suitable for Malayalam. If a word in a morphologically rich language ends with a valid suffix, most likely that will be its suffix. In agglutinative languages, more than one suffix can attach to a stem. So a word may ends with more number of valid suffixes. In this case the actual stem is the prefix by stripping largest suffix from the word. Since Malayalam is a morphologically rich and agglutinative language, the high accuracy of smallest stem baseline compared to largest stem baseline is very intuitive. Also note that, this may not be true for morphologically poor languages.

To get more insight about the shortcomings of these experimentations and models, we did an error analysis of two newly proposed systems. Each and every wrong splits are analyzed. Errors in both models are common and it follows same distribution. The errors in the models are mainly three types. These errors are as follows.

1. Because of the incomplete suffix set, the models may unable to split any other inflections of some words. So the *possible_stem_set* of such words will be totally independent from *possible_stem_set* of other words. Then our stemming models choose one stem from *possible_stem_set* of such words randomly and that may leads to wrong stemming.
2. Some orthographically similar words with different stem affect the stemming process. For example, നിക്കോളെ {nikkole} and നിക്കോള {nikkola} are two different proper nouns, but it looks very similar. By removing one of the accusative case markers {e} from നിക്കോളെ {nikkole} we will get നിക്കോള {nikkola}, but നിക്കോള+Accusative {nikkola+Accusative} is നിക്കോളയെ {nikkolaye}. In this case our stemming models wrongly selects the word നിക്കോള {nikkola} as the stem of നിക്കോളെ {nikkole}.
3. There may exist multiple solutions that minimize the number of distinct stems or stem distributional entropy. In such cases, our models randomly choose one of the solutions. That solution may have some wrong stems.

Some wrong stems identified by our models and its error type (3 types mentioned above) are shown in the Table 6.

Table 6. Erroneous Stems Samples Found During Error Analysis

Word	Stem (Identified)	Stem (Correct)	Error Type
ചെനാറിൻ {chenaarin} (Chenar(Proper noun)+Gen)	ചെനാറിൻ {chenaarin}	ചെനാ {chena}	1
തൊൽക്കാപ്പിയരും {tholkkappiyarum} (Tholkkappi- yar(Proper noun)+Conj)	തൊൽക്കാപ്പിയരും {tholkkappiyaru}	തൊൽക്കാപ്പിയര {tholkkappiyara}	1
വേണ്ടി {vendi} (for that)	വേണ്ട {venda}	വേണ്ടി {vendi}	1
നിക്കോളെ {nikkole} (Nikkole(Proper noun))	നിക്കോള {nikkola}	നിക്കോളെ {nikkole}	2
മില്ലൻ {millan} (Millan(Proper noun))	മില്ല {milla}	മില്ലൻ {millana}	3
അമലുകളുടെ {amalukalude} (Amal(Proper noun)+Pl+Gen)	അമലുകളു {amalukala}	അമല {amala}	1

One of the reason for the remaining errors are the incomplete suffix set. If the suffix set is not complete then stems of some words in the input corpus will not be identified. Stem information from such words may useful in the process of stem identification of other words.

Consider the word വേണ്ടി {vendi} in the Table 6. Since ിയും {iyum} and ിയോ {iyo} are not in the suffix set, the system cannot split വേണ്ടിയോ {vendiyo} and വേണ്ടിയും {vendiyum} (other inflectional variants of വേണ്ടി {vendi}). In this case, effectively there is no other word in the corpus for the splitting of the word വേണ്ടി {vendi}. So a string from *possible_stem_set* of വേണ്ടി {vendi} selects randomly. Majority of remaining errors are because of this problem. So completion of the suffix set is considered for further improvement.

6 Conclusion and Future Work

To solve the stemming problem by using an unannotated corpus and a suffix set, two models are proposed. Problem of computing first model that directly

reduces the number of lexicon entries is NP-hard. A greedy approximation for this model is also proposed. Second model is a probabilistic model and it reduces the entropy of stem distribution. Approximated version of first model and second model are evaluated on Malayalam corpus. We got the best accuracy of 93% by using the MSE model. Improvement in suffix set is proposed for future work. Analysing the performances of these proposed models in other languages is also considered for future work.

References

1. Hammarström, H., Borin, L.: Unsupervised learning of morphology. *CL*, 309–350 (2011)
2. Goldsmith, J.A.: Unsupervised learning of the morphology of a natural language. *CL* (2), 153–198 (2001)
3. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. *TSLP* 4 (2007)
4. Clark, A.: Partially supervised learning of morphology with stochastic transducers. In: *NLPRS*, pp. 341–348 (2001)
5. Snover, M.G., Jarosz, G.E., Brent, M.R.: Unsupervised learning of morphology using a novel directed search algorithm: taking the first step. In: *Proc. of ACL-WMPL 2002*, pp. 11–20 (2002)
6. Dreyer, M., Eisner, J.: Graphical models over multiple strings. In: *Proc. of EMNLP 2009*, pp. 101–110 (2009)
7. Johnson, H., Martin, J.: Unsupervised learning of morphology for english and inuktitut. In: *Proc. of NAACL-HLT 2003*, pp. 43–45 (2003)
8. Bosch, A.v.d., Daelemans, W.: Memory-based morphological analysis. In: *Proc. of ACL 1999* (1999)
9. Hammarström, H.: A naive theory of affixation and an algorithm for extraction. In: *Proc. of HLT-NAACL 2006*, pp. 79–88 (June 2006)
10. Hammarström, H.: Poor Man’s Stemming: Unsupervised Recognition of Same-Stem Words. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) *AIRS 2006*. LNCS, vol. 4182, pp. 323–337. Springer, Heidelberg (2006)
11. Monson, C., Carbonell, J.G., Lavie, A., Levin, L.S.: ParaMor and Morpho Challenge 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) *CLEF 2008*. LNCS, vol. 5706, pp. 967–974. Springer, Heidelberg (2009)
12. Dasgupta, S., Ng, V.: High-performance, language-independent morphological segmentation. In: *HLT-NAACL*, pp. 155–163 (2007)
13. Dasgupta, S., Ng, V.: Unsupervised morphological parsing of bengali. *Language Resources and Evaluation*, 311–330 (2006)
14. Lawphongpanich, S.: Frank-wolfe algorithm. In: *Encyclopedia of Optimization*, pp. 1094–1097 (2009)
15. David, S.M.I.P.S.: A morphological processor for malayalam language. Technical report, South Asia Research (2007)

On the Adequacy of Three POS Taggers and a Dependency Parser

Ramadan Alfareed and Denis Béchet

LINA, University of Nantes, 2, rue de la Houssinière, 44000 Nantes, France
ramadan.alfared@etu.univ-nantes.fr, Denis.Bechet@univ-nantes.fr

Abstract. A POS-tagger can be used in front of a parser to reduce the number of combinations of possible dependency trees which, in the majority, give spurious analyses. In the paper we compare the results of the addition of three morphological taggers to the parser of the CDG Lab. The experimental results show that these models perform better than the model which do not use a morphological tagger at the cost of loosing some correct analyses. In fact, the adequacy of these solutions is mainly based on the compatibility between the lexical units defined by the taggers and the dependency grammar.

1 Introduction

Dependency parsing is widely used in natural language processing. Many different algorithms were suggested and evaluated for this task. They achieve both, a reasonable time complexity and a high accuracy and need a minimal necessary annotation based on POS. In the paper, we show how the use of POS tags may improve the rate of spurious ambiguity of parsing with a wide scope categorial dependency grammar (CDG) of French which uses *Lefff* [18] as its lexical base. Our parsing models have different characteristics because POS taggers and the French CDG (based on syntactic categories of *Lefff*) use different lexical units (LU) models. Therefore recall and precision performances vary. In CDG [12], all LU are grouped into lexical classes (CDG classes). All units of a class share the same syntactic types [1]. *Lefff* is a wide coverage lexicon of French representing a very large set of highly structured lexical information. [2] previously showed a correspondence between CDG classes and *Lefff* classification.

The rest of the paper is structured as follows. Section 2 describes dependency grammars, Section 3 describes the parsing problem and our models. Section 4 presents the experimental evaluation, and Section 5 contains a comparative error analysis of the our different models. Finally, Section 6 concludes the paper.

2 Dependency Grammars

Dependency-based representations have become increasingly popular in syntactic parsing, especially for languages that exhibit free or flexible word order, such

¹ A LU may belong to several CDG classes.

as Czech [7], Bulgarian [15], Turkish [14] and Russian [4]. Many practical implementations of dependency parsing are restricted to projective structures, where the projection of a head word has to form a continuous substring of the sentence. Dependency Grammars (DGs) are formal grammars assigning dependency trees (*DT*) to a sentence. A *DT* is a tree with words as nodes and dependencies, i.e. named syntactic binary relations between words, as arrows. In other words, if two words v_1 and v_2 are related by dependency d (denoted $v_1 \xrightarrow{d} v_2$) then v_1 is the governor and v_2 is the subordinate. Figure 1 illustrates the dependencies in the sentence “Au commencement était le Verbe.”

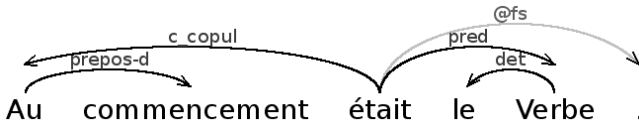


Fig. 1. French: in the beginning was the Verbe

The relation $était \xrightarrow{pred} Verbe$ represents the predicative dependency between the copula *était* and the subject *Verbe*. The head of this sentence is *était*.

2.1 Categoricals Dependency Grammars

Categorical Dependency Grammars introduced by [12] are lexicalized in the same sense as the conventional categorial grammars. Here we briefly give basic information on CDG. The CDG types are defined over a set C of elementary categories (types). A syntactic type may be repetitive or optional $C^* =_{df} \{X^* | X \in C\}$, $C^? =_{df} \{X^? | X \in C\}$. CDG use iteration to express all kinds of repetitive dependencies such as modifiers and coordination relations. The non-projective dependencies are expressed using *polarized valencies*. Namely, the governor G which has a right distant subordinate D through a discontinuous dependency d has positive dependency $\nearrow d$, whereas its subordinate D has the negative valency $\searrow d$. Together these dual valencies define the discontinuous dependency d . In CDG, the anchor types of the form $\#(\searrow d)$, $\#(\swarrow d)$ are used in the same way as local dependencies. More precisely, CDG define discontinuous dependencies using polarized valencies (left/right, positive/negative) and a simple valencies pairing principle First Available (*FA*). For every valency, the corresponding one is the closest dual valency in the indicated direction. In order to define polarized categories, we distinguish between four dependency polarities: left and right positive \nwarrow, \nearrow and left and right negative \swarrow, \searrow . For each polarity $v \in \{\nwarrow, \swarrow, \nearrow, \searrow\}$ there is a unique dual polarity $\check{v} : \nwarrow = \swarrow, \swarrow = \nwarrow, \nearrow = \searrow, \searrow = \nearrow$. $\nearrow C, \nwarrow C, \swarrow C$ and $\searrow C$ denote the corresponding sets of polarized distant dependency categories. The general form of a

CDG type is $[l_1 \setminus l_2 \setminus \dots \setminus H / \dots / r_2 / r_1]^P$ where the head type H defines the incoming dependency on the word, l_1 and r_1 are elementary (iterated or optional) categories which correspond to left or right outgoing dependencies or anchors, P is a potential, a string of polarized valencies which defines the long distance dependencies (incoming or outgoing), see [8], [3] and [9] for more details.

Figure 2 shows two discontinuous dependencies (non-projective – using dotted arrows) associated to two anchor relations (show under the sentence) in the sentence “elle la lui a donnée.”.

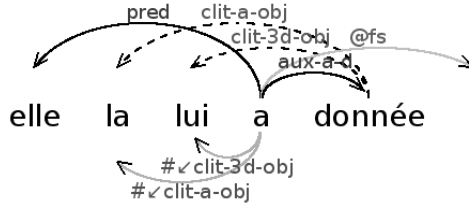


Fig. 2. Non-projective DS: “*she it[fem.] to him has given”

Categorial dependency grammars which define this dependency tree affect the types which anchor the clitics *la*, *lui* on the auxiliary *a*. The discontinuous dependencies are represented by dotted arrows.

$elle^{PN(Le\grave{x}=pers,C=n)}$	$\mapsto [pred]$
$la^{PN(Le\grave{x}=pn,F=clit,C=a)}$	$\mapsto [\#(\checkmark clit-a-obj)]^{\checkmark clit-a-obj}$
$lui^{PN(Le\grave{x}=pn,F=clit,P=3,C=d)}$	$\mapsto [\#(\checkmark clit-3d-obj)]^{\checkmark clit-3d-obj}$
$a^{Vaux(Le\grave{x}=avoir,F=fin)}$	$\mapsto [\#(\checkmark clit-3d-obj) \#(\checkmark clit-a-obj) \backslash pred \backslash S / @fs / aux-a-d]^{\checkmark clit-3d-obj \checkmark clit-a-obj}$
$donn\acute{e}e^{V2t(F=pz,C1=a,C2=d g l,T=past)}$	$\mapsto [aux-a-d]$
$.FullStop(Le\grave{x}=".")$	$\mapsto [@fs]$

The word *elle* is classified as a pronoun (PN), where *pers* and *n* correspond to person and noun. The word *la* is classified as a clitic at accusative case. The word *lui* is classified as a clitic for 3rd person with complement at dative case. The word *a* is classified as an auxiliary verb with a finite form “F=fin” while the word *donnée* is classified as a di-transitive verb where *pz* is “past participle” form and has two arguments (complement), the first complement is a direct complement (at accusative) and the second complement is a dative, a genitive or a locative. The NLP team of LINA has developed a large scale CDG of French and a general purpose offline CDG parser. In this French CDG, the types are assigned to CDG classes (see [13] for details). The CDG parser is currently used to develop dependency tree corpora. The linguist’s interface of this parser lets manually select for every LU one of its possible classes and one of

the possible head dependencies. Then the parser finds all analyses compatible with the selection. Our goal in this paper is to automatically pre-fetch the most probable CDG classes per LU depending on its POS and to measure the impact of this selection on the ambiguity of the parser as applied to the CDG of French.

3 POS-Based Parsing Models

Usually, the task of disambiguation of a dependency parser consists in deriving a single correct dependency tree τ for a given sentence S . The parsing problem consist in finding the mapping of an input sentence S , constituted of words $w_1 \cdots w_n$, to its dependency tree τ . More precisely, given a parsing model M and a sentence S , we derive the optimal dependency tree τ for S according to M . So the parsing problem is to construct the optimal dependencies for the input sentence, given the parsing model. Some parsers solve this problem by deriving a single analysis for each sentence. Our task is different: we should instead lower the ambiguity of the French CDG using POS tagging models and we evaluate the effect obtained by our method. Our POS-based parsing models first choose the most probable CDG classes through POS tags for the words in a sentence. Applying our method we should resolve a technical problem which arises from the nature of the lexical database of the CDG of French. In fact, this lexical database uses the (freely available) wide-coverage French lexicon *Lefff* [18]. It contains 110,477 lemmas (simple and compounds) and 536,375 inflected forms. The main part of the French CDG classes linked with *Lefff* is saved in a PostgreSQL Database. In this database, each LU of *Lefff* corresponds to one or several CDG classes. This correspondence is realized in the main table `lexicon`. Unfortunately, *Lefff* is not complete and contains errors. Therefore, in the lexical database there are several facilities for correction and complementation of *Lefff* definitions.

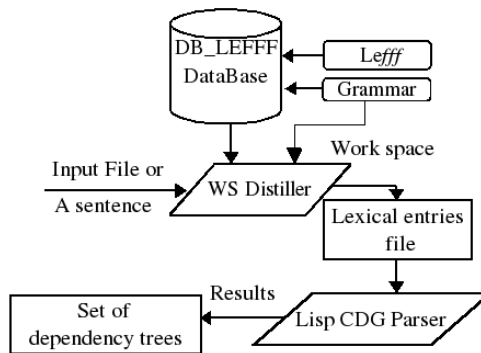


Fig. 3. General form of Base model

Before we describe our approach, we should explain that the CDG parser uses the following two strategies for lexicon (called below models):

- Base model gives access to the forms contained in the classes of the French CDG (about 1500 forms), and also gives access to the original definitions of *Lefff* related with the CDG classes in the database. Figure 3 illustres the workspace distiller for Base model.

- The three other models use *Lefff* and the French CDG implicitly. First, a tagger is applied to the input sentence (Tree-Tagger [19] in T.T Model, MELt-Tagger [10] in M.T model and Brill-Tagger [5] in B.T model). Then, depending on the computed (composite in general) LU and their POS, a compatible lexical definition for every pair (LU, POS) and the corresponding CDG class is found in the database. If and when they exist, they are integrated to the input file that is sent to the parser. Figure 4 presents this strategy.

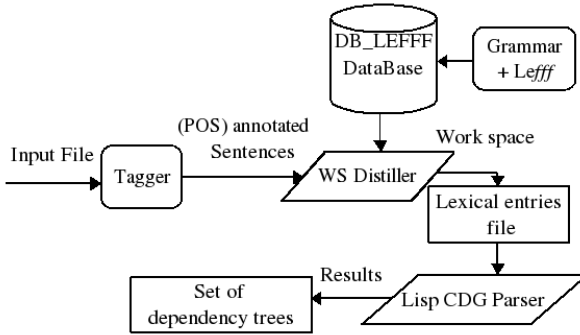


Fig. 4. General form of POS-based parsing models

Correspondence between POS tagging and *Lefff*: The correspondence between CDG classes and *Lefff* is established using the workspace distiller shown in Figure 4. We try to find the correspondence between the tags of POS-tagger and the syntactic categories of *Lefff*. This correspondence is approximate, because the lexical models of POS-tagger and of *Lefff* and the french CDG are different. Table 1 shows some examples of the correspondence.

Table 1. Examples of correspondence between POS-tagger and *Lefff*

<i>Lefff</i>	T.T	M.T	B.T
np	NAM	NPP	NAM, SBP
coo	KON	CC, ET	COO
det	DET:ART	DET	DTN
nc	NOM, NUM	NC	SBC, CAR

Some important information on POS-tagging e.g. `VER:futu` are very useful to determine both the *mood*² and the *tense* of a verb. In this case, we also compare them to the *mood* and *tense* of the lexicon database. For instance, `VER:futu` means that *mood* is indicative and *tense* is future.

The WS distillers of the different models take an input file which contains the sentences with its POS (annotated sentence), and the output is a file with (lexical entries) annotated CDG classes and word features. The algorithm chooses the most probable CDG classes for a LU by using POS tags and *Lefff category* of the database. This algorithm consists of the next three steps.

- First we search the word of the sentence and its POS tag and compare them to *form* and its *category* in the database, if they are equal, we take its CDG classes and morphological features such as *mood*, *tense*, *person*, *gender*, *number* and *lemma* and we save all these information in file (lexical entry).
- If there is no result from the first step then we only search the word compare it with *form* and take all CDG classes and morphological information that correspond to this *form*.
- If the word corresponds to nothing we classify this LU as an unknown term: The CDG class “`UT(Lex=V|N|Adj|Adv)`” is assigned to it.

4 Experimental Results

In our experiments we use a corpus of sentences divided into two subsets. The first subset, serving as a test set, consists of 1443 French sentences that have been analysed by Alexandre Dikovsky to build the French Gold Standard dependency corpus (DTB) [11]: a corpus with French sentences from various sources. These sentences have 14974 projective and non projective (discontinuous) dependencies. The second subset of the corpus has 184 French sentences from the French treebank [1].

For the experiment with the first subset, we first run the parser with the maximum number of viewed dependency trees set to 2000. We can not request all the possible dependency trees per sentence. With the French CDG, it generates hundreds of spurious structures per sentence. So for long and complex sentences, it is practically impossible to know how many DS are produced. Till the final step where the DS are extracted from the chart, the parsing algorithm is polynomial. Given that the number of these DS may be exponential with respect to the size of the chart, the final step is exponential in space in the worst case. In this step, the DS are generated from the chart in a certain order. The parser generates a HTML report page, which includes various useful statistics. It can also produce an XML structure representation of every DS including all necessary information.

For our POS-based parsing models, we compute the ambiguity reduction of

dependency trees using the formula $X^j = \sum_{i=1}^N A_i^j$ where A_i^j is the number of

² Italic names like *form*, *category*, *mood*, *tense*, *person*, *gender*, *number* and *lemma* are fields of the French database that are provided by *Lefff*.

Table 2. *Experimental results of (parsing accuracy and parsing times) compared to four parsing models*

	Base	T.T	M.T	B.T
# Sentences that have at least one analysis (1)	1089	1125	1005	949
# Sentences have 100% correct dependencies	1089	874	892	667
Recall	75.46%	60.65%	61.81%	46.22%
Precision	100%	77.68%	88.75%	70.28%
# of dependencies (from (1))	8255	9571	7730	7603
# correct dependencies	8255	8465	7380	6838
Recall correct dependencies	55.12%	56.53%	49.28%	45.66%
Precision	100%	88.44%	95.47%	89.93%
Labeled accuracy average (on all 1443 sentences)	100%	82.27%	85%	69%
Parsing time		08h 46mn 05s	05h 07mn 3s	07h 49mn 18s
		05h 07mn 3s	07h 49mn 18s	05h 07mn 59s

dependency trees that are found for model j , where j is Base model, T.T model, M.T model or B.T model and $i=1,\dots,N$. N represents the number of the sentences that have a 100% correct analysis in every model. For our experiments, $N=325$. The reduction of dependency trees of model j is $\frac{X^{Base}-X^j}{X^{Base}} \times 100$, where j is different from the Base model. We don't compute the number of dependency trees of the sentences that have more 2000 analyses.

For the second experiment with the first subset, we run the parser with the maximum number of viewed dependency trees set to 1 in order to obtain the maximal number of analyzed sentences, and also to know how many sentences have all dependencies correctly analyzed. We compute the total number of composition

trees³ using the formula $Y^j = \sum_{i=1}^M B_i^j$, where B_i^j is the number of composition

trees for sentences that are found using model j , where j is Base model, T.T model, M.T model or B.T model and $i=1,\dots,M$. M represents the number of sentences that have at least one analysis in every model. For our experiments $M=780$. The reduction of the composition trees for model j is $\frac{Y^{Base}-Y^j}{Y^{Base}} \times 100$, where j is different from Base model. Table 2 shows the experimental results for each parsing model and also gives comparative parsing times for each parsing model.

The evaluation of the parser uses classical measures. It uses the labeled attachment score AS_L for the mode on Figure 4, which is the proportion of tokens

³ For each dependency tree, there are several composition trees because each composition tree specifies also a set of word features, a class and a type. We use the number of composition trees rather than the number of dependency trees, because it's usually not possible to evaluate the total number of dependency trees.

that are assigned the correct head and the correct dependency label. There are several sentences which have accuracy over 90% of correct dependencies, but we count only the sentences that have 100% correct analysis. The result in Table 2 shows that our models achieve between 88% and 95% accuracy for correct dependency relation labeling.

We don't need to use unlabeled attachment score AS_U , because we don't compare the result of several parsers. AS_U is used by [6]. It compares two parsing architectures for the high accuracy on unknown words. Indeed BKY+FLABELER [17] achieves only a 82.56% tagging accuracy for the unknown words in the development set (5.96% of the tokens), whereas MElt+MST [16] achieves 90.01%.

Table 3. *Experimental results (composition trees and dependency trees) compared to four parsing models*

	Base	T.T	M.T	B.T
# Composition trees	16330×10^8	27×10^8	34×10^8	28×10^8
Reduction of # CT	% wrt j model	99.83%	99.79%	99.82%
Geometric mean # CT_j / # CT_{Base}	-	0,035	0,037	0,033
# DT	153938	42572	44056	46718
Ambiguity reduction (DT)	% wrt j model	72.34%	71.38%	69.65%
Geometric mean # DT_j / # DT_{Base}		0,24	0,23	0,26

The result in Table 3 shows that the numbers of composition trees of the three POS-based parsing models are inferior than Base model. Our models achieve high reduction of both composition trees and dependency trees (over 99% and 70% respectively).

Comparing between the three POS-based parsing models, we note that M.T model performs better than T.T and B.T models in terms of parsing accuracy. But T.T model is better than the other models in terms of ambiguity reduction and parsing time. Table 4 shows an example to explain the reduction for both dependency trees and composition trees of the four parsing models in the sentence : “il parle en courtes phrases.”. Figure 5 gives the DS of this sentence.

Table 4. *Reduction (dependency trees and composition trees) on the sentence “il parle en courtes phrases”*

	Base	T.T	M.T	B.T
Reduction of # DT	268	54	211	54
Reduction of # CT	28732	3336	8559	3336

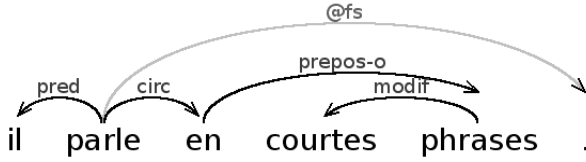


Fig. 5. "he speaks in short sentences"

$il \mapsto PN(lex = pers, C = n)$
 $parle \mapsto Vt(F = fin, C = g)$
 $en \mapsto PP(F = compl - obl, C = o)$
 $courtes \mapsto Adj(F = modifier)$
 $phrases \mapsto N(lex = common)$

Figure 6 and Figure 7 show of different DS for the sentence "he makes them learn that_{g=fem}".

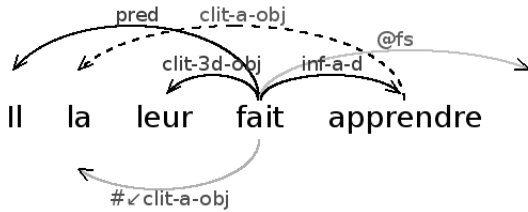


Fig. 6. "Correct DS"

$Il \mapsto PN(lex = pers, C = n)$
 $la \mapsto PN(lex = pn, F = clit, C = a)$
 $leur \mapsto PN(lex = pn, F = clit, P = 3, C = d)$
 $fait \mapsto Vlight(F = fin)$
 $apprendre \mapsto V2t(F = inf, C1 = a|p, C2 = d|g|l)$

For the second subset of 184 French sentences, we use only the Base model and T.T model. We only compute the number of composition trees using the same formula of the first subset. The results show that pre-fetching CDG classes reduces the ambiguity with respect to composition trees more than 99%.

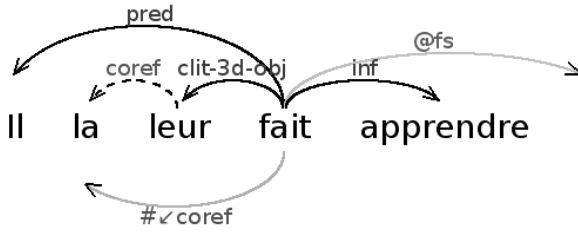


Fig. 7. "Incorrect DS"

Table 5. Effect of class pre-fetching (Paris 7 corpus)

	Base	T.T
#CT	1097325498316350	7048627222816
#CT	10973254×10^8	70486×10^8
Total reduction #CT	% wrt Base model	99,9%
Geometric mean of	-	0.002
# CT _{T.T} /# CT _{Base}		
# CT of the sentence Figure 8	17284	241
# DT of the sentence Figure 8	1295	68

Table 5 summarizes the experimental results for Base model and T.T model for the number of composition trees.

The results given in Table 5 show that pre-fetching classes reduces the ambiguity in terms of composition trees more than 99%.

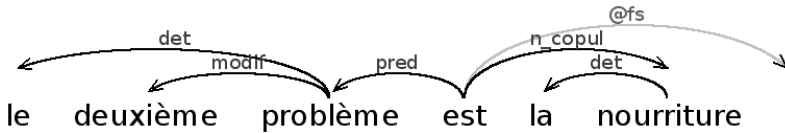


Fig. 8. "paris 7 corpus: The second problem is the food"

5 Discussion

This discussion provides a brief analysis of the errors made by the POS tagger for the first corpus, when we investigate the POS category of erroneous instances.

In T.T model, there are 318 sentences that have no dependency tree, 177 sentences among them are not analyzed (time exceeded), which means there was not enough time to parse them, (the maximum number of seconds per sentence is set to 60 second), as we indicate above for ambiguous CDG. There are 141 sentences that are analysed as incorrect sentences. A first reason for this fact is that, there

Table 6. *Errors make by the Parser for the parsing models*

	Base	T.T	M.T	B.T
memory exhausted	12	17	1	0
bad sentences	0	141	127	314
too complex sentences	342	160	310	180

is at least one of the next compound words in the sentences: *à peu près, Hé bien, dès lors, de loin, au dessous, la-bas, des EU, de l'*. In these cases, Tree-Tagger tags these compound words as separate words: *à* as *prep*, *peu* as *adv*, *près* as *adv*, etc. But the database has only complete entries for them. The second main reason is that Tree tagger makes errors in tagging some LU. Thus the distiller do not find a good CDG class for these LU. We have seen that the results of B.T model are worse than those for T.T and M.T models, because Brill-Tagger also makes many errors in tagging. For example, the sentence “Adam ne donne à Ève pas que les pommes.” (Adam do not give to Eve only the apples) is annotated as *Adam/SBC:sg ne/ADV donne/SBC:sg à/PREP Ève/SBC:sg pas/ADV que/SUB les/DTN:pl pommes/SBC:pl ./..* The verb *donne* is tagged as common noun *SBC* and not as a verb. There are 17 sentences which contain *donne* that are tagged as *SBC* and 28 sentences which contain the past participle “été” of the verbe “être” that are also tagged as *SBC*. Errors like these lead to 314 sentences that have been analyzed as incorrect sentences. The example in Figure 8 shows the reason why we have obtained several analyses for this sentence. We note that the word “la”, (*the*) is only tagged by the T.T, M.T and B.T models as “determiner”. Thus, there is only one CDG class corresponding to this LU: “*Det(Lex=art|pn)*”, while Base model leaves all the CDG classes for this word. More precisely, the word *la* has in the grammar three different CDG classes, because this LU has different syntactic categories in *Lefff* such as *det*, *nc* and *pro* as illustrated in Table 7. This lexical ambiguity in Base model leads

Table 7. *Some features and classes in the database for LU “La”*

Form	Cat	Class
la	cla	PN(Lex=pers,C=a)
la	cla	PN(Lex=pn,F=clit,C=a)
la	det	Det(Lex=art—pn)
la	nc	N(Lex=common)

to several analyses of this sentence. This example shows the importance of the assignment of proper POS tag to every word in a sentence which is to be parsed. When we have compared between the three taggers, we have seen that M.T model had better precision than M.T model and B.T model because the lexical categories of M.T model are similar to the lexical categories of *Lefff*. T.T model

is better than the other models in terms of ambiguity reduction, parsing time and the number of correct dependency label; Mainly because the parser succeeds more oftenly to find at least one analysis (not always 100% correct). In the one hand, the POS tagging reduces the search space for the parser, and also reduces ambiguity, improving parsing by limiting the search space. The sentences are also more often completely analyzed by the parser, because the search space is smaller as compared to Base model.

On the other hand, using POS tagging, we lost some analyses for the reason of POS tagging errors and compatibility between taggers and parsers. These sentences have been considered as incorrect sentences by the parser.

6 Conclusion

This paper evaluates the rate of improving dependency parsing through using three different POS-tag models. These models choose the most probable grammatical classes for a word in a sentence based on POS tags, unfortunately at the cost of losing some correct analyses. Our experimental results have demonstrated the utility of POS-based parsing models. These models achieved substantial reductions of the number of dependency trees and of composition trees per sentence. The results might be better if we could find a tagger that is completely compatible with *Lefff* and the French CDG. For instance, complex prepositions like “au dessous” are not correctly recognized by the POS taggers but are needed by the French CDG.

References

1. Abeillé, A., Barrier, N.: Enriching a french treebank. In: Proc. of LREC 2004, Lisbon, Portugal (2004)
2. Alfarad, R., Béchet, D., Dikovskiy, A.: “CDG LAB”: a toolbox for dependency grammars and dependency treebanks development. In: Gerdes, K., Hajicova, E., Wanner, L. (eds.) Proc. of the 1st Intern. Conf. on Dependency Linguistics (Depling 2011), Barcelona, Spain (September 2011)
3. Béchet, D., Dikovskiy, A., Foret, A.: Dependency Structure Grammars. In: Blache, P., Stabler, E.P., Busquets, J.V., Moot, R. (eds.) LACL 2005. LNCS (LNAI), vol. 3492, pp. 18–34. Springer, Heidelberg (2005)
4. Boguslavsky, I., Iomdin, L., Sizov, V., Tsinman, L., Petrochenkov, V.: Rule-based dependency parser refined by empirical and corpus statistics. In: Proc. of the 1st Intern. Conf. on Dependency Linguistics (Depling 2011), Barcelona, Spain (September 2011)
5. Brill, E.: Some advances in transformation-based part of speech tagging. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, pp. 722–727 (1994)
6. Candito, M., Crabbé, B., Denis, P.: Statistical french dependency parsing: Treebank conversion and first results. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA), Valletta (2010)

7. Collins, M., Hajic, J., Ramshaw, L., Tillmann, C.: A statistical parser for Czech. In: ACL 1999 (1999)
8. Dekhtyar, M., Dikovskiy, A.: Categorical dependency grammars. In: Moortgat, M. (ed.) Proceedings of Categorical Grammars 2004, pp. 76–91 (2004)
9. Dekhtyar, M., Dikovskiy, A.: Generalized categorical dependency grammars. In: Pillars of Computer Science 2008, pp. 230–255 (2008)
10. Denis, P., Sagot, B.: Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In: Pacific Asia Conference on Language, Information and Computation, Hong Kong, China (2009)
11. Dikovskiy, A.: Categorical dependency grammars: from theory to large scale grammars. In: Gerdes, K., Hajicova, E., Wanner, L. (eds.) Proc. of the 1st Intern. Conf. on Dependency Linguistics (Depling 2011), Barcelona, Spain (September 2011)
12. Dikovskiy, A.: Dependencies as categories. In: “Recent Advances in Dependency Grammars”, COLING 2004 Workshop, pp. 90–97 (2004)
13. Dikovskiy, A.: Towards wide coverage categorical dependency grammars. In: Proceedings of the ESSLLI 2009 Workshop Parsing with Categorical Grammars - Parsing with Categorical Grammars Workshop ESSLLI 2009 Book of Abstracts, pp. 230–255 (2009)
14. Eryiğit, G., Oflazer, K.: Statistical dependency parsing for turkish. In: EACL 2006, pp. 89–96 (2006)
15. Marinov, S., Nivre, J.: A data-driven dependency parser for bulgarian. In: Proceedings of TLT, pp. 89–100 (2005)
16. McDonald, R.: Discriminative Training and Spanning Tree Algorithms for Dependency Parsing. Ph.D. thesis, University of Pennsylvania (July 2006)
17. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: ACL 2006, pp. 433–440 (2006)
18. Sagot, B.: The lefff, a freely available and large-coverage morphological and syntactic lexicon for french. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA), Valletta (2010)
19. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester, UK (1994)

Will the Identification of Reduplicated Multiword Expression (RMWE) Improve the Performance of SVM Based Manipuri POS Tagging?

Kishorjit Nongmeikapam¹, Aribam Umananda Sharma¹,
Laishram Martina Devi¹, Napoleon Keisam¹,
Khangengbam Dilip Singh¹, and Sivaji Bandyopadhyay²

¹ Department. of Computer Science and Engg., Manipur Institute of Technology,
Manipur University, Imphal, India

² Department. of Computer Science and Engg., Jadavpur University,
Jadavpur, Kolkata, India

{kishorjit.nongmeikapa,marti.laishram1,
nepoleonk,dlkhng}@gmail.com,
umnnd@hotmail.com,
sivaji_ju_cse@yahoo.com

Abstract. Reduplicated Multiword Expressions (RMWEs) are abundant in Manipuri, the highly agglutinative India language. The Part of Speech (POS) tagging of Manipuri using Support Vector Machine (SVM) has been developed and evaluated. The POS tagger has been updated with identified RMWEs as another feature. The performance of the SVM based POS tagger before and after adding RMWE as a feature have been compared. The SVM based POS tagger has been evaluated with the F-Score of 77.67% which has increased to 79.61% with RMWE as an additional feature. Thus the performance the POS tagger has improved after adding RMWE as an additional feature.

Keywords: Part of Speech (POS), Multiword Expressions (MWE), Reduplicated Multiword Expressions (RMWE), Support Vector Machine (SVM), Feature.

1 Introduction

Part of Speech (POS) tagging is an important task as a preprocessing activity in various natural language processing (NLP) systems like Information Retrieval, Summarization, Machine Translation, Name Entity Recognition (NER), Multiword Expression (MWE) identification, etc. Part of Speech tagging is the task of labelling each word or token in a sentence with its appropriate syntactic category called part of speech. The Manipuri language or Manipuri or Meiteiron is a Scheduled Indian Language which is very highly agglutinative in nature. Manipuri uses two scripts: the first one is the Bengali script while the other one is the original Meitei Mayek (Script). The work here uses the Manipuri language in Bengali Script. It may be appropriate to mention that Manipuri documents in Bengali script are easily available.

Part of Speech taggers have been developed for several languages in the world. There are several works on POS taggers for English: a Simple Rule-based based POS tagger is reported in [1], transformation-based error-driven learning based POS tagger in [2], maximum entropy methods based POS tagger in [3] and Hidden Markov Model (HMM) based POS tagger in [4]. For Chinese, the works are found ranging from rule based, HMM to Genetic Algorithms [5]-[7]. For Indian languages like Bengali works are reported in [8]-[10] and for Hindi in [11]. Works of POS tagging using SVM methods can also be seen in [12]-[13].

Manipuri POS tagging is reported in [14]-[15]. The identification of Reduplicated Multiword Expression (RMWE) is reported in [16]-[17]. Web Based Manipuri Corpus for Multiword NER and RMWEs Identification using SVM is reported in [18].

The paper is organized in the following manner. Section 2 describes the highly agglutinative nature of Manipuri, Section 3 briefs about Manipuri Reduplicated MWEs, Section 4 details about Manipuri Stemming Algorithm, Section 5 includes discussion about the SVM model and feature selection, Section 6 gives the details about the experiments and the evaluation results, identification of RMWEs has been discussed in Section 7 while the conclusion is drawn in Section 8.

2 Manipuri Is Highly Agglutinative in Nature

The inflexion plays an important role in the word formation of Manipuri. In general the suffixes and prefixes almost determine the POS of a word. In English roots are almost free, i.e., there are separate grammatical categories for noun (box, bird etc), verb (run, try etc), adjective (tall, large etc) etc. This is not in the case in Manipuri since there are no separate roots for adjective and adverb in Manipuri.

The work of [19] explains that Manipuri roots are of two types, they are *free* and *bound* root. Free roots can stand alone without suffixes in a sentence and bound root takes other affixes excepting the one with free roots.

Manipuri is highly agglutinative in nature. Let us consider an example word: “পুশিনহনজরমগদবনিদকো” (“pusinhənjərəmgədəbənīdəkō”), which means “(I wish I) myself would have caused to bring in (the article)”. Here there are 10 (ten) suffixes being used in a verbal root: “pu” is the verbal root which means “to carry”, “sin” (in or inside), “hən” (causative), “j” (reflexive), “rəm” (perfective), “gə” (associative), “d” (particle), “b” (infinitive), “n” (copula), “d” (particle) and “kō” (endearment or wish).

Also in [19] its mention that altogether 72 (seventy two) affixes are listed in Manipuri out of which 11 (eleven) are prefixes and 61 (sixty one) are suffixes. Table 1 shows the prefixes of 10 (ten number) because the prefix ঋ (mə) is used as both formative and pronominal. Table 2 shows the suffixes in Manipuri with only are 55 (fifty five) suffixes in the table since some of the suffixes are used with different forms of usage such as গুম (gum) which is used as a particle as well as a proposal negative, দা (də) as particle as well as locative and না (nə) as nominative, adverbial, instrumental or reciprocal.

Table 1. Prefixes in Manipuri

Prefixes used in Manipuri
অ, ই, ই, থু, চা, ত, থ, ন, ম and শে

Table 2. Suffixes in Manipuri

Suffixes used in Manipuri
কন, কুম, কো, খরে, খই, খি, খোয়, গা, গনি, গী, গুম, ঙে, চা, চো, থ, খই, থেক, থোক, দা, দি, দুলা, দে, না, নতে, নি, নিং, নু, নে, গী, ফাং, বা, বু, মক, মন, মিন, মুক, লে, লা, লক, ল্ম, লি, লী, লু, লু, লে, লো, লোয়, শনু, শি, শিং, শিন, শু, হই and হন

3 Reduplicated Multiword Expressions in Manipuri

The Manipuri Reduplicated Multiword Expression (RMWE) identification has been reported in [16]. In [20] the process of reduplication is defined as: *reduplication is that repetition, the result of which constitutes a unit word*. These single unit words are the RMWE.

The reduplicated MWEs in Manipuri are classified mainly into four different types. These are: 1) Complete Reduplicated MWEs, 2) Partial Reduplicated MWEs, 3) Echo Reduplicated MWEs and 4) Mimic Reduplicated MWEs. Apart from these four types there are also cases of a) Double reduplicated MWEs and b) Semantic Reduplicated MWEs.

3.1 Complete Reduplicated MWEs

In the complete reduplication MWEs the single word or clause is repeated once forming a single unit regardless of phonological or morphological variations. Interestingly in Manipuri these complete reduplicated MWEs can occur as Noun, Adjective, Adverb, *Wh*- question type, Verbs, Command and Request. For example, মরিক মরিক (*mərik mərik*) which means *drop by drop*.

3.2 Partial Reduplicated MWEs

In case of partial reduplication the second word carries some part of the first word as an affix to the second word, either as a suffix or a prefix. For example, চংথোক চংসিন (*cə-thok cə-sin*) means *to go to and fro*, শামী লানমী (*sa-mi lan-mi*) means *army*.

3.3 Echo Reduplicated MWEs

The second word does not have a dictionary meaning and is basically an echo word of the first word. For example, থকসি থাসি (*thk-si kha-si*) means *good manner*. Here the first word has a dictionary meaning *good manner* but the second word does not have a dictionary meaning and is an echo of the first word.

3.4 Mimic Reduplicated MWEs

In the mimic reduplication the words are complete reduplication but the morphemes are onomatopoeic, usually emotional or natural sounds. For example, *করক করক* (*'khrək khrək'*) means *'cracking sound of earth in drought'*.

3.5 Double Reduplicated MWEs

In double Reduplicated MWE there consist of three words, where the prefix or suffix of the first two words is reduplicated but in the third word the prefix or suffix is absent. An example of double prefix reduplication is *ইমুন ইমুন মুনবা* (*'i-mun i-mun mun-ba'*) which means, *'completely ripe'*. It may be noted that the prefix is duplicated in the first two words while in the following example suffix reduplication take place, *ভাশোক ভাশোক ভাবা* (*'ηəw-srok ηəw-srok ηəw-ba'*) which means *'shining white'*.

3.6 Semantic Reduplicated MWEs

Both the reduplicated words have the same meaning as well as the MWE. Such type of MWEs is very special to the Manipuri language. For example, *পাম্বা কে* (*'pamba kəy'*) means *'tiger'* and each of the component words means *'tiger'*. Semantic reduplication exists in Manipuri in abundance as such words have been generated from similar words used by seven clans in Manipur during the evolution of the language.

4 Manipuri Word Stemming Algorithms

The stemming work of Manipuri has been reported in [21]. In this algorithm Manipuri words are stemmed by stripping the suffixes in an iterative manner. As mentioned in Section 2, a Manipuri word is rich with suffixes and prefixes. An iterative method of stripping is done by using the acceptable list of prefixes (11 numbers) and suffixes (61 numbers) as mentioned in the Table 1 and Table 2 respectively.

4.1 The Algorithm

This stemmer mainly consist of four algorithms the first one is to read the prefixes, the second one is to read the suffixes, the third one is to identify the stem word removing the prefixes and the last algorithm is to identify the stem word removing the suffixes.

Two file, *prefixes_list* and *suffixes_list* are created for prefixes and suffixes of Manipuri. In order to test the system another testing file, *test_file* is used.

The prefixes and suffixes are removed in an iterative approach as shown in the algorithm 3 and algorithm 4 until all the affixes are removed. The stem word is stored in *stemwrđ*.

Algorithm1: *read_prefixes()*

1. Repeat 2 to 4 until all prefixes (p_i) are read from *prefixes_list*
2. Read a prefix p_i
3. $p_array[i]=p_i$
4. $p_wrđ_count =i++$;
5. exit

Algorithm2: *read_suffixes()*

1. Repeat 2 to 4 until all suffixes (s_i) are read from *suffixes_list*
2. Read a suffix s_i
3. $s_array[i]=s_i$
4. $s_wrđ_count=i++$;
5. exit

Algorithm3: *Stem_removing_prefixes(p_array, p_wrd_count)*

1. Repeat 2 to 16 for every word (w_i) are read from the *test_file*
2. *String stemwrđ=" "*;
3. *for(int j=0;j<p_wrd_count;j++)*
4. {
5. *if(w_i .startsWith($p_array[j]$))*
6. {
7. *stemwrđ= w_i .substring(w_i .length()-((w_i .length()-($p_array[j].toString()).length()))$, w_i .length());*
8. *w_i =stemwrđ;*
9. *j=-1;*
10. }
11. *else*
12. {
13. *stemwrđ= w_i ;*
14. }
15. }
16. *write stemwrđ;*
17. *exit;*

Algorithm4: *Stem_removing_suffixes(s_array, s_wrd_count)*

1. Repeat 2 to 16 for every word (w_i) are read from the *test_file*
2. *String stemwrđ=" "*;
3. *for(int j=0;j<s_wrd_count;j++)*
4. {
5. *if(w_i .endsWith($s_array[j]$))*
6. {
7. *stemwrđ= w_i .substring(0, w_i .indexOf($s_array[j]$));*
8. *w_i =stemwrđ;*
9. *j=-1;*
10. }

```

11.  else
12.  {
13.    stemwrd=wi;
14.  }
15. }
16. write stemwrd;
17. exit

```

5 SVM Model and Feature Selection

The idea of Support vector machines (SVM) are discussed in [22, 23]. Support Vector Machines is the new technique for pattern classification which has been widely used in many application areas. The kernel parameters setting for SVM in training process has an impact on the classification accuracy. Feature selection is another factor that impacts classification accuracy. The POS tagger has been developed using the SVM mentioned in [21, 24], which performs classification by constructing an N dimensional hyperplane that optimally separates data into two categories. The training process has been carried out by YamCha¹ toolkit, an SVM based tool for detecting classes in documents and formulating the POS tagging task as a sequential labelling problem. Here, the *pairwise* multi-class decision method and *polynomial kernel function* have been used. For classification, TinySVM-0.07² classifier is used.

5.1 Feature Selection

A very careful selection of the feature is important in SVM. Various candidate features are listed. Those candidate features which are listed to run the system are as follows,

Dynamic POS: The POS of the previous words are considered as a feature. It is because the POS of the previous words are important for the POS of the succeeding word.

Surrounding words as feature: Preceding word(s) or the successive word(s) are important in POS tagging because these words play an important role in determining the POS of the present word.

Surrounding Stem words as feature: The Stemming algorithm mentioned in Section 4 is used. The preceding and the following stemmed words of a particular word can be used as features. It is because the preceding and the following words influence the present word POS tagging.

Number of acceptable standard suffixes as feature: As mention in Section 2, Manipuri being an agglutinative language the suffixes plays an important in determining the POS of a word. For every word the number of suffixes are identified during stemming and the number of suffixes is used as a feature.

¹ <http://chasen-org/~taku/software/yamcha/>

² <http://chasen-org/~taku/software/TinySVM/>

Number of acceptable standard prefixes as feature: Same is the case for the prefixes. It also plays an important role for Manipuri language. For every word the number of prefixes are identified during stemming and the number of prefixes is used as a feature.

Acceptable suffixes present as feature: The standard 61 suffixes of Manipuri which are identified is used as one feature. As mention with an example in Section 2, suffixes are appended one after another. The maximum number of appended suffixes in Manipuri is reported as ten. So taking into account of such cases, for every word ten columns separated by a space are created for every suffix present in the word. A “0” notation is being used in those columns when the word consists of no acceptable suffixes.

Acceptable prefixes present as feature: 11 prefixes have been manually identified in Manipuri and the list of prefixes is used as one feature. For every word if the prefix is present then a column is created mentioning the prefix, otherwise the “0” notation is used. Upto three prefixes are considered for observation.

Length of the word: Length of the word is set to 1 if it is greater than 3 otherwise, it is set to 0. Very short words are generally pronouns and rarely proper nouns.

Word frequency: A range of frequency for words in the training corpus is set: those words with frequency <100 occurrences are set the value 0, those words which occurs >=100 are set to 1. The word frequency is considered as one feature since occurrence of determiners, conjunctions and pronouns are abundant.

Digit features: Quantity measurement, date and monetary values are generally digits. Thus the digit feature is an important feature. A binary notation of ‘1’ is used if the word consist of a digit else ‘0’.

Symbol feature: Symbols like \$,%, - etc. are meaningful in textual use, so the feature is set to 1 if it is found in the token, otherwise, it is set to 0. This helps to recognize SYM (Symbols) and QFNUM (Quantifier number) tags.

5.2 Pre-processing and Feature Extraction

A Manipuri text document is used as an input file. The training and test files consist of multiple tokens or words. In addition, each token consists of multiple (but fixed number) columns where the informations in the columns are used as a features. A sequence of tokens and other information in a line becomes a **sentence**. Before undergoing training and testing in the SVM the input document is converted into a multiple token file with fixed information column representing the values of the various features as mentioned in section 5.1. In the training file the last column is manually tagged with all the identified POS tags³ whereas in the test file we can either use the same tagging for comparisons or only ‘O’ for all the tokens regardless of POS.

6 Experiment and Evaluation Results

Manual POS tagging is time consuming so three linguist experts from Linguistic Department, Manipur University has tagged 25,000 tokens. Considering it to be the Gold standard it is split into two files, one for training file and another for testing file.

³http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

In order to evaluate the experiment results, the system used the parameters of Recall, Precision and F-score. These parameters are defined as follows:

Recall:

$$R = \frac{\text{No of correct POS tag assigned by the system}}{\text{No of correct POS tag word in the text}}$$

Precision,

$$P = \frac{\text{No of correct POS tag assigned by the system}}{\text{No of POS tag assigned by the system}}$$

F-score,

$$F = \frac{(\beta^2 + 1) PR}{\beta^2 P + R}$$

Where β is one, precision and recall are given equal weight.

6.1 Experiment for Selection of Best Feature

A total of 25,000 words are divided into two files, one consisting of 20000 words as training file and the second file consisting of 5000 words as testing file. The sentences are separated into equal numbers of columns representing the different features separated by blank spaces.

The experiment is performed with different combinations of features. The features are manually selected in such a way that the result shows an improvement in the F-measure. Among the different experiments with different combinations Table 4 lists some of the best combinations. Table 3 explains the notations used in Table 4.

Table 3. Meaning of the notations

Notation	Meaning
W[-i,+j]	Words spanning from the i^{th} left position to the j^{th} right position
SW[-i, +j]	Stem words spanning from the i^{th} left to the j^{th} right positions
P[i]	The i is the number of acceptable prefixes considered
S[i]	The i is the number of acceptable suffixes considered
L	Word length
F	Word frequency
NS	Number of acceptable suffixes
NP	Number of acceptable prefixes
D	Digit feature (0 or 1)
SF	Symbol feature (0 or 1)
DP[-i]	POS spanning from the previous i^{th} word's POS

Table 4. System performance with various feature combinations

Feature	R(in %)	P(in %)	FS(in %)
DP[1],W[-2,+1], SW[-1,+1], P[1], S[4], L, F, NS, NP, D, SF	71.43	85.11	77.67
DP[1],W[-2,+2], SW[-2,+1], P[1], S[4], L, F, NS, NP, D, SF	69.64	82.98	75.73
DP[2], W[-2,+3], SW[-2,+2], P[1], S[4], L, F, NS, NP, D, SF	67.86	80.85	73.79
DP[1], W[-3,+1], SW[-3,+1], P[1], S[4], L, F, NS, NP, D, SF	66.07	80.43	72.55
DP[3], W[-3,+3], SW[-3,+2], P[1], S[5], L, F, NS, NP, D	62.50	76.09	68.63
W[-3,+4], SW[-2,+3], P[2], S[5], L, F, NS, SF	50.00	64.37	56.28
W[-4,+1], SW[-4,+1], P[2], S[6], L, NP, D, SF	31.25	74.47	44.03
DP[3], W[-4,+3], SW[-3,+3], P[3], S[9], L, F, D, SF	30.00	66.67	41.38
W[-4,+4], SW[-4,+4], P[3], S[10], NS, NP	28.57	25.00	26.67

6.2 Evaluation and the Best Feature Set

The best feature set so far reported in the previous SVM based Manipuri POS tagger [15] is as follows:

F = { W_{i-2} , W_{i-1} , W_i , W_{i+1} , |prefix| ≤ 3 , |suffix| ≤ 3 , Dynamic POS tag of the previous two word, Symbol feature, Digit feature, Length feature}

The list consists of surrounding words, prefixes, suffixes, dynamic POS, symbol feature, digit feature and length features. The model which has been adopted here has a different list and the best feature set is selected after experimenting with the possible combinations. The best result is the one which shows the best F-measure among the results. This happens with the following feature set:

F = { Dynamic POS tag of the previous word, W_{i-2} , W_{i-1} , W_i , W_{i+1} , SW_{i-1} , SW_i , SW_{i+1} , number of acceptable standard suffixes, number of acceptable standard prefixes, acceptable suffixes present in the word, acceptable prefixes present in the word, word length, word frequency, digit feature, symbol feature }

The best feature set in the model gives the Recall (**R**) of **71.43%**, Precision (**P**) of **85.11%** and F-measure (**F**) of **77.67%**. The earlier model in [19] reports that the SVM based system shows the F-measure of **74.38%** which is lower than this model.

7 Can Improvement Be Done Using RMWE?

With the intention of improving the efficiency of the POS tagging in the SVM system, the identification of Manipuri RMWEs has been carried and the identified Manipuri RMWEs have been included as a feature in the SVM based POS tagger.

7.1 The RMWE Identification Model

The algorithms and models for finding RMWEs in Manipuri text as suggested in [16] is used for the identification of RMWEs. The first model (Figure 1) is used to identify the complete, mimic, partial, double and echo RMWEs.

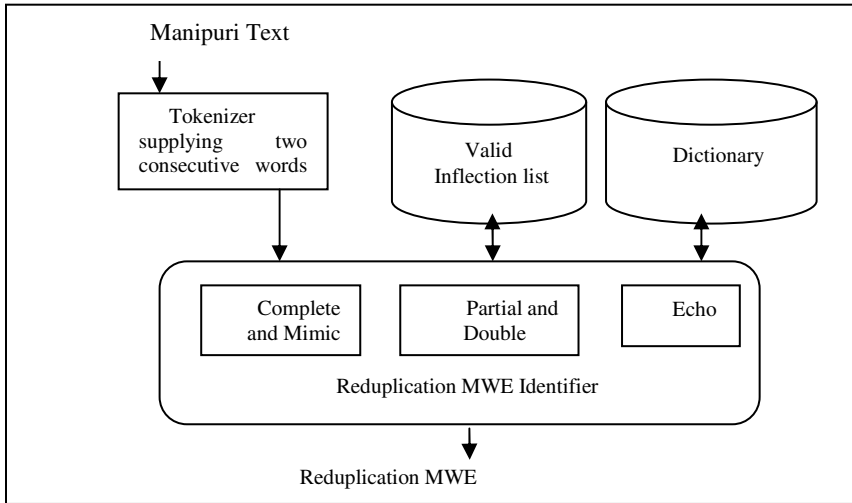


Fig. 1. The First model for identifying four types of reduplication MWEs and double reduplication

The functions performed by the different parts of the proposed model are:

- a) **Tokenizer**, separates the words based on blank space or special symbols to identify two consecutive words W_i and W_{i+1} .
- b) **Reduplication MWE Identifier**, it verifies the valid inflections present in the words and also checks the semantics of W_{i+1} in the dictionary for the Echo words.
- c) **Valid Inflection List**, list of commonly used valid inflection is listed. The inflection list is an important resource for MWE identification.
- d) **Dictionary**, it includes the lexicon and the associated semantics.

The second model (Figure 2) is used for identification of semantic reduplicated MWEs.

Some of the functions of the second model are same with the first model like **Tokenizer** and **Dictionary** parts but the **Semantic Comparator** works for checking the similarity in semantics of W_1 and W_2 .

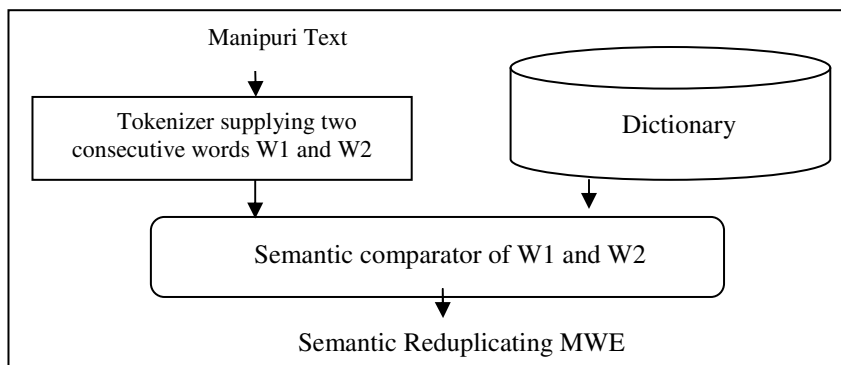


Fig. 2. The Second model for identifying semantic reduplicating MWE

7.2 RMWE as a Feature

After running the training and test files with the above model (Section 8.1) the identified RMWEs are marked with B-RMWE for the beginning and I-RMWE for the rest of the RMWE and O for the non RMWEs. The RMWE tags are placed as a new column in the multiple token file for both training and testing.

The training file is run again with the SVM toolkit which outputs a new model file. This model file is used to run the test file which adds up a new output column which is the POS tag by the machine after the learning process. This new tagging is used to compare with the previous output. The output shows an improvement of the following:

Table 5. Results RMWEs as a feature in SVM system

Model	Recall	Precision	F-Score
CRF	73.21	80.23	79.61

8 Conclusion

The main motive of this work is to test whether RMWE can improve the performance of SVM based POS tagging. The previous reported [15] Manipuri POS tagger has a reported F-measure of 74.38%. The SVM based POS tagger reported in the paper works with a different feature selection and has outperformed the previous reported model with the F-measure of 77.67%.

With the RMWE as a feature it even improves the F-measure to **79.61%**. We expect that the F-measure will improve when general MWEs are also considered as features of the POS tagger.

References

1. Brill, E.: A Simple Rule-based Part of Speech Tagger. In: The Proceedings of Third International Conference on Applied Natural Language Processing. ACL, Trento (1992)
2. Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics* 21(4), 543–545 (1995)
3. Ratnaparakhi, A.: A maximum entropy Parts-of-Speech Tagger. In: The Proceedings EMNLP, vol. 1, pp. 133–142. ACL (1996)
4. Kupiec, R.: Part-of-speech tagging using a Hidden Markov Model. *Computer Speech and Language* 6(3), 225–242 (1992)
5. Lin, Y.C., Chiang, T.H., Su, K.Y.: Discrimination oriented probabilistic tagging. In: The Proceedings of ROCLING V, pp. 87–96 (1992)
6. Chang, C.H., Chen, C.D.: HMM-based Part-of-Speech Tagging for Chinese Corpora. In: The Proceedings of the Workshop on Very Large Corpora, Columbus, Ohio, pp. 40–47 (1993)
7. Lua, K.T.: Part of Speech Tagging of Chinese Sentences Using Genetic Algorithm. In: The Proceedings of ICCS 1996, pp. 45–49. National University of Singapore (1996)
8. Ekbal, A., Mondal, S., Bandyopadhyay, S.: POS Tagging using HMM and Rule-based Chunking. In: The Proceedings of SPSAL 2007, IJCAI, India, pp. 25–28 (2007)
9. Ekbal, A., Haque, R., Bandyopadhyay, S.: Bengali Part of Speech Tagging using Conditional Random Field. In: The Proceedings 7th SNLP, Thailand (2007)
10. Ekbal, A., Haque, R., Bandyopadhyay, S.: Maximum Entropy based Bengali Part of Speech Tagging. In: Gelbukh, A. (ed.) *Advances in Natural Language Processing and Applications*, vol. (33), pp. 67–78 (2008)
11. Singh, S., Gupta, K., Shrivastava, M., Bhattacharya, P.: Morphological Richness offsets Resource Demand—Experiences in constructing a POS tagger for Hindi. In: The Proceedings of COLING-ACL, Sydney, Australia (2006)
12. Antony, P.J., Mohan, S.P., Soman, K.P.: SVM Based Part of Speech Tagger for Malayalam. In: The Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing (ITC), Kochi, Kerala, India, pp. 339–341 (2010)
13. Ekbal, A., Mondal, S., Bandyopadhyay, S.: Part of Speech Tagging in Bengali Using SVM. In: Proceedings of International Conference on Information Technology (ICIT), Bhubaneswar, India, pp. 106–111 (2008)
14. Doren Singh, T., Bandyopadhyay, S.: Morphology Driven Manipuri POS Tagger. In: The Proceeding of IJCNLP NLPLPL 2008, IIIT Hyderabad, pp. 91–97 (2008)
15. Doren Singh, T., Ekbal, A., Bandyopadhyay, S.: Manipuri POS tagging using CRF and SVM: A language independent approach. In: The Proceeding of 6th International Conference on Natural Language Processing (ICON 2008), Pune, India, pp. 240–245 (2008)
16. Kishorjit, N., Bandyopadhyay, S.: Identification of Reduplicated MWEs in Manipuri: A Rule based Approach. In: The Proceeding of 23rd International Conference on the Computer Processing of Oriental Languages (ICCPOL 2010), Redwood City, San Francisco, pp. 49–54 (2010)
17. Nongmeikapam, K., Laishram, D., Singh, N.B., Chanu, N.M., Bandyopadhyay, S.: Identification of Reduplicated Multiword Expressions Using CRF. In: Gelbukh, A. (ed.) *CICLing 2011, Part I. LNCS*, vol. 6608, pp. 41–51. Springer, Heidelberg (2011)

18. Doren Singh, T., Bandyopadhyay, S.: Web Based Manipuri Corpus for Multiword NER and Reduplicated MWEs Identification using SVM. In: The Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), the 23rd International Conference on Computational Linguistics (COLING), Beijing, pp. 35–42 (2010)
19. Nonigopal Singh, N.: A Meitei Grammar of Roots and Affixes. A Thesis. Manipur University, Imphal (1987) (unpublish)
20. Yashawanta, C.S.: Manipuri Grammar, pp. 190–204. Rajesh Publications, Delhi (2000)
21. Kishorjit, N., Bishworjit, S., Romina, M., Chanu, N.M., Bandyopadhyay, S.: A Light Weight Manipuri Stemmer. In: The Proceedings of National Conference on Indian Language Computing (NCILC), Chochin, India (2011)
22. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
23. Huang, C.-L., Wang, C.-J.: A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications* 31, 231–240 (2006), doi:10.1016/j.eswa.2005.09.024
24. Joachims, T.: Making Large Scale SVM Learning Practical. In: Scholkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods-Support Vector Learning* (1999)

On Formalization of Word Order Properties^{*}

Vladislav Kuboň, Markéta Lopatková, and Martin Plátek

Charles University in Prague, Faculty of Mathematics and Physics, Czech Republic
{vk,lopatkova}@ufal.mff.cuni.cz, martin.platek@mff.cuni.cz

Abstract. This paper contains an attempt to formalize the degree of word order freedom for natural languages. It exploits the mechanism of the analysis by reduction and defines a measure based on a number of shifts performed in the course of the analysis. This measure helps to understand the difference between the *word order complexity* (how difficult it is to parse sentences with more complex word order) and *word order freedom* in Czech (to which extent it is possible to change the word order without causing a change of individual word forms, their morphological characteristics and/or their surface dependency relations). We exemplify this distinction on a pilot study on Czech sentences with clitics.

1 Introduction

In this paper we are suggesting a formal treatment suitable for languages with higher degree of word order freedom. This phenomenon, although very important for the complexity (and success) of parsing algorithms, seems to be neglected by the formal theory. The languages with higher degree of word order freedom tend to achieve worse parsing results even when identical parsing methods are applied. The stochastic methods or methods of machine learning exploited in parsing do not answer the question whether the freedom of word order is really the crucial phenomenon which not only theoretically, but also practically constitutes the greatest parsing challenge.

We are not aiming at any particular parsing algorithm or system; instead, we would like to clarify some basic features and notions which may play a role in the investigations of the word order freedom. We are going to exploit the method of analysis by reduction and the formal data type derived from this method, so-called D-trees. A complete description of both the method and the data type can be found for example in [10].

The word order variations in a particular language can be divided into two major groups – those which affect word forms in a sentence (and their morphological or even syntactic categories) and those which don't. The first group may be illustrated for example by the differences between an active and passive sentence:

^{*} The paper reports on the research supported by the grant of GAČR No. P202/10/1333 and partially by the grant of GAČR No. P103/10/0783. The research was carried out under the project of MŠMT ČR No. MSM0021620838.

Peter bought a book for Maria yesterday.

A book was bought by Peter for Maria yesterday.

Because English is a language with very sparse number of word forms derived from a single lemma, the changes of the word order are accompanied by insertions or deletions of functional words or prepositions. This mechanism is common also in some languages with richer inflection and higher degree of word order freedom, as e.g. in Czech.

This type of word order variations does not constitute a good basis for the investigation of word order freedom, because too many factors are involved. The latter group is more interesting from our point of view: the constraint that the word forms and their morphological and syntactic properties should not be changed together with the change of a particular word order, helps to study and measure the word order freedom separately from other phenomena. Let us look at the set of examples for both languages mentioned above:

English allows only variations of the word order position of the temporal complementation, e.g.:

Peter bought a book for Maria yesterday.

Yesterday Peter bought a book for Maria.

Czech allows substantially more permutations, for example:

Petr koupil včera Marii knihu.

Knihu Marii koupil Petr včera.

Petr včera koupil Marii knihu.

Knihu koupil Petr včera Marii.

Petr Marii koupil včera knihu.

Knihu včera koupil Petr Marii.

Petr Marii včera koupil knihu.

Včera koupil Petr Marii knihu.

Marii Petr koupil včera knihu.

Včera koupil knihu Marii Petr.

Marii Petr včera koupil knihu.

Včera Petr Marii koupil knihu.

Knihu Petr koupil Marii včera.

Včera knihu Marii koupil Petr.

...

These sentences have the same syntactic structure (apart from the word order) – the same morphological (case, number, gender, tense, ...) and syntactic categories (Subject, Predicate, Direct or Indirect Object, ...) are assigned to individual words.¹ On the basis of these examples it seems that a simple measure of a degree of word order freedom could be related to a number of permutations preserving the above mentioned properties.

Although very natural, this measure would have two substantial drawbacks. The first one is a certain *gray zone* existing especially in languages with higher degree of word order freedom (and higher number of possible permutations) in which it is very difficult to judge individual permutations because they may be acceptable only in a very obscure reading. The second issue is related to the fact that the maximal number of permutations in a sentence with n words reaches $n!$ – a number too big for a manual enumeration of all variants.

In this paper we propose a different approach. It is based both on a sound theoretical and formal background as well as on syntactically annotated data. The

¹ These sentences differ in their communicative dynamism – what is an ‘old information’ referring to a previous context and what is a ‘new information’, i.e., the ‘core’ of the message.

theoretical background has already been described for example in [6,10], where a formal model of a stratificational dependency approach to natural language description is proposed and further enriched. The model is based on an elementary method of analysis by reduction (AR, see [6], here Sect. 2.2). The analysis by reduction has served as a motivation for a family of so called *restarting automata*, see [9].

In order to demonstrate how the proposed measure of word order freedom works, we are going to apply it to selected sentences from the Prague Dependency Treebank (PDT)² a large-scale treebank of Czech, based on the theory of the Functional Generative Description [11].

2 The Background

2.1 Functional Generative Description

The theoretical linguistic basis for our research is provided by the Functional Generative Description (FGD in the sequel), see esp. [11]. FGD is characterized by its stratificational and dependency-based approach to the language description.

The *stratificational approaches* split language description into layers, each layer providing complete description of a (disambiguated) sentence and having its own vocabulary and syntax. As we focus on surface word order phenomena in this project, we make use of three *surface layers* of FGD only³

***a*-layer** (analytical layer) capturing surface syntax in a form of a labeled dependency tree (non-projective in general); the most important information being an *analytical function*, i.e., surface syntactic function of a node (as e.g. Subject, Object, Attribute);

***m*-layer** (morphological layer) capturing morphology, i.e., a string of triples [word form, lemma, tag] for each word or punctuation mark in a sentence;

***w*-layer** (word layer) capturing individual words and punctuation marks in a form of a simple string.

Individual items of these three layers straightforwardly reflect individual words and punctuation marks in a sentence – there is an one-to-one correspondence between individual symbols of the *w*- and *m*-layer (we leave aside small exceptions here) and between individual symbols of *m*- and *a*-layer. We will refer to triples of items from all three layers corresponding to a single occurrence of a word form as to a *lexical bundle* in the sequel.

FGD as a *dependency-based approach* describes surface syntactic information in a form of dependency trees (Sect. 3.1; see also [8]). Individual words of a sentence are represented as nodes of the respective dependency tree, each node

² <http://ufal.mff.cuni.cz/pdt.html>

³ We disregard here the *tectogrammatical layer*, which captures deep syntax comprising language meaning – the core concepts of this layer being dependency, valency, and topic-focus articulation.

being a complex unit capturing the lexical, morphological and syntactic features; relations among words are represented by oriented edges. The dependency nature of these representations is very important particularly for languages with relatively high freedom of word order – the dependency trees are generally able to treat the word order in a natural and transparent way.

2.2 Basic Principles of the Analysis by Reduction

Analysis by reduction (AR) is based on a stepwise simplification of an analyzed sentence. It defines possible sequences of reductions (deletions) in the sentence – each step of AR is represented by deleting (at least) one word of the input sentence.⁴ Consequently, it is possible to derive formal dependency relations between individual sentence members based on the possible order(s) of reductions, see also [10]; very comprehensive overview of the problem can be found in [11], see also for further references there.

Using AR, we analyze an input sentence (*w*-layer) enriched with the meta-language information from the *m*- and *a*-layers. Symbols on different layers representing a single word of an input sentence (lexical bundles) are processed simultaneously. A sentence is simplified until so called *core structure* is reached (typically its predicate).

The principles of AR on the surface layers can be summed up in the following observations:

- The fact that a certain word (or a group of words) can be deleted implies that this word (or group of words) *depends in AR* on one of the words retained in the simplified sentence; the latter being called *governing word(s) in AR*. In other words, the governing word(s) has/ve the syntactic distribution identical to the entire combination of the governing and the dependent words.
- Two words (or groups of words) can be deleted in an arbitrary order if and only if they are *mutually independent in AR*.⁵
- In order to ensure correctness of the simplified sentence (see below), certain groups of words have to be deleted in a single step (e.g., a preposition and the corresponding noun; a finite verb plus its auxiliaries); such words are said to constitute a *reduction component*. Even in such cases, it is usual to determine governing-dependent pairs on the layer of surface syntax (*a*-layer). In such a case, it is necessary to define (rather technical) special rules for particular language phenomena.
- In specific cases, an operation *shift* consisting in shifting of a word form to another word order position is used to ensure correctness of the simplified sentence.

When simplifying an input sentence, it is necessary to apply certain elementary constraints assuring adequate analysis on the surface layers:

⁴ For the purposes of this article, we leave aside possible rewriting steps, which are necessary for an adequate analysis on the tectogrammatical layer of the FGD.

⁵ Here we focus on dependency relations and we disregard non-dependency relations as esp. coordination and apposition.

1. principle of *correctness*: a grammatically correct sentence must remain correct after its simplification;
2. principle of *shortening*: at least one word (i.e., its correlates on *w*-, *m*- and *a*-layer) must be deleted in each step of AR;
3. principle of *generalization*: a simplified sentence must preserve an overall meaning of the original sentence;
4. principle of *minimality*: each step of AR must be ‘minimal’: any potential reduction step concerning less symbols in the sentence would violate the principle of correctness on the *w*-layer.

The basic principles of AR can be illustrated on the following Czech sentence:

- (1) *Marii se Petr tu knihu rozhodl nekoupit.*
 to-Mary REFL Peter that/the book decided not-to-buy
 ‘To Mary, Peter decided not to buy the book.’

Let us look more closely at several possible reduction steps. For example, it is clear that the demonstrative pronoun *tu* ‘that/the’ has to be deleted prior to the noun *knihu* ‘book’ – otherwise, the simplified sentence would not be correct, e.g. **Marii se Petr tu rozhodl nekoupit.* ‘*To Mary, Peter decided not to buy the.’ It implies that the pronoun depends on the noun according to the AR principles. The dependency relation is represented as the edge [*tu, knihu*] in the dependency tree.

Similarly, the noun *knihu* ‘book’ must be reduced prior to the verb *nekoupit* ‘not-to-buy’ (as **Marii se Petr tu knihu rozhodl.* ‘*To Mary, Peter decided the book.’ is an incorrect simplification) and thus the noun depends on the verb.

On the other hand, *Marii* ‘to-Mary’ and *knihu* ‘book’ can be reduced in an arbitrary order, thus these words are mutually independent.

We can continue in the same manner until the sentence is reduced to the pair *Se rozhodl.* ‘(He) decided.’ However, the simplified sentence is not a correct Czech sentence – the reflexive morpheme *se* is a clitic and thus it has to be located in the ‘second position’ in a correct sentence⁶ For this reason, the shift operation is applied which results in a correct simplified sentence *Rozhodl se.* ‘(He) decided.’

This pair represents a core structure as it cannot be further simplified; technical rules are applied for creating the edge (a verb being always a governor for its REFL clitic), see [6].

Figure 1 shows the resulting structure describing the previous sentence. It consists of the surface non-projective syntactic *a*-tree, of the string of triples [word form, lemma, tag] on the *m*-layer and of the string of word forms (with their translation) on the *w*-layer. The dotted lines interconnect corresponding nodes.

⁶ In Czech, clitics have specific constraints on their surface word order position: they occupy so called *Wackernagel’s position*: roughly speaking, the position after the first prosodic unit; its syntactic description can be found in [3], see also Sect. 4.

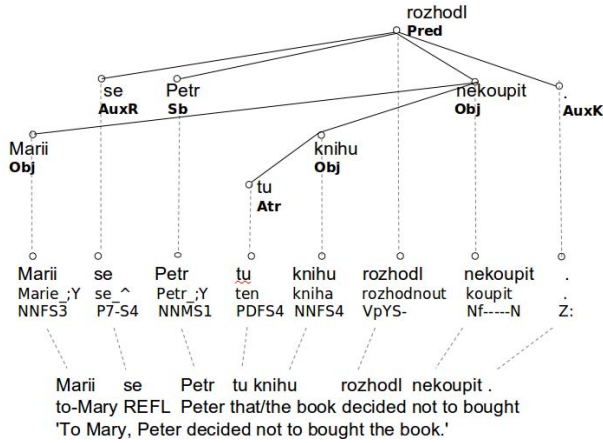


Fig. 1. Sentence (1) – representation on a -, m - and w -layers according to FGD

3 Formalization of Basic Notions

3.1 Delete/Dependency Trees (D-trees) and Characteristic Sentence

One of the important factors of our formalization of word order freedom is a choice of an appropriate data type. In this paper we work with tree structures denoted as a (surface or analytical) D-trees (Delete or Dependency trees), see e.g. [10]; D-tree is a rooted ordered tree with edges oriented from its leaves to its root. Nodes of each tree correspond to individual occurrences of word forms in a sentence. Moreover, we suppose a total ordering on the nodes that reflects word order in a sentence.

This means that each node of a D-tree is a pair $[i, a]$, where a represents an input word (referred to as a *lexical part of a node*) and i denotes a word order position in a sentence (called a horizontal index).

In fact, this version of D-tree actually constitutes a special case of a DR-tree introduced in [10] which does not consider rewriting. The complete formal definition of a D-tree can be found in the same paper.

The concept of D-tree reflects the analysis by reduction (AR) (without rewriting) – its structure corresponds to a way how individual words of a sentence are deleted in the course of the corresponding steps of the analysis by reduction. Each edge of a D-tree connects a word form $[i, a_i]$ to some other word form $[j, a_j]$, which cannot be deleted earlier than $[i, a_i]$ in (any branch of) analysis by reduction of the same sentence.

The root of such a D-tree is one of the nodes corresponding to the word forms which remain in the sentence after the last reduction step of AR.

For the investigation of the word order freedom it is also necessary to limit our scope and to exclude sentences which would bring into the play different phenomena than the word order. Let us therefore limit our considerations to *correct*

sentences of a natural language and their *correct syntactic and morphological analysis* based on the principles of FGD.

First, we can naturally integrate all relevant information from the FGD surface layers into a single D-tree. With respect to the one-to-one correspondence between items of the three surface layers, we assign all *a*-, *m*-, and *w*- information for an individual word form (or punctuation mark) to a single node of a D-tree; such a D-tree is referred to as a (*correct*) *surface D-tree* (see Fig. 2 for a correct surface D-tree for sentence (1)).

A set of such surface trees is denoted as CT.

We refer to a string $w = a_1, \dots, a_n$ corresponding to a correct surface D-tree as to a (*correct*) *characteristic sentence*. Thus, a (complex) symbol $a_i, i \in \{1, \dots, n\}$, reflects a word form enriched with the relevant information from each of *a*-, *m*-, and *w*-layers – we call such a complex symbol a *lexical bundle*. For example, the lexical bundle for the word form *rozhodl* ‘decided’ consists of the word form itself (*w*-layer), from its analytical function *Préd* (*a*-layer), and its lemma *rozhodnout* ‘to decide’ and morphological tag *VpYS-* (*m*-layer), see Fig. 2.

3.2 Measures of Non-projectivity

When considering word order freedom, we have to take into account one phenomenon which is common in languages with higher degree of word order freedom, namely non-projective constructions (for previous usage of this term see esp. [745]). In order to classify this phenomenon, it is necessary to define certain notions allowing for an easy definition of projectivity/non-projectivity and also for the introduction of useful measures of non-projectivity (these notions are formally defined in [4]).

The *coverage of a node u of a D-tree* identifies nodes from which there is a path to u in the D-tree (including empty path). It is expressed as a set of horizontal indices of nodes directly or indirectly dependent upon a particular node. For example, the coverage of the node of the verb *nekoupit* in Fig. 2 consists of the horizontal indices of nodes representing the words *Marii, tu, knihu, nekoupit*.

The notion of a coverage leads directly to a notion of *a hole in a subtree*. Such a hole exists if the set of indices in the coverage is not a continuous sequence. In Fig. 2 there is only a single subtree with at least one (actually two) hole in its coverage, the subtree rooted in the verb *nekoupit*.

We say that D-tree T is *projective* if none of its subtrees contains a hole; otherwise, T is *non-projective*.

3.3 Shift Operation

In order to be able to describe necessary word order shifts in the course of AR, we need to define a notion of equivalence for D-trees. Such equivalence (denoted as DP-equivalence) is defined as follows: *DP-equivalent trees* are those D-trees which have (i) the ‘same’ sets of nodes, i.e., the nodes have identical lexical parts and may differ only in their horizontal indices, and (ii) their edges always connect ‘identical’ pairs of nodes (nodes with identical lexical part). It actually means

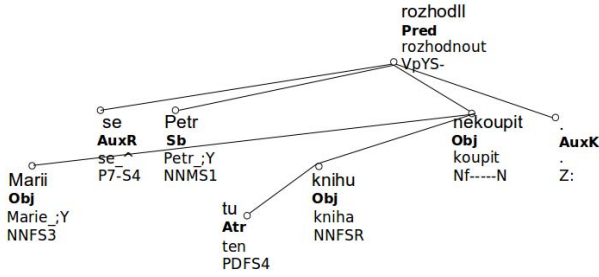


Fig. 2. Sentence (1) – correct surface D-tree

that a particular set of DP-equivalent trees contains the D-trees representing sentences created by a permutation of the words of the original sentence but having the same dependency relations.

Let T be a D-tree; the set of D-trees which are DP-equivalent to T will be denoted $DPE(T)$. In other words, $DPE(T)$ is a set of D-trees which differ only in the word order of their characteristic sentence.

The previous concepts allow us to introduce a new feature, a *number of reduction steps enforcing a shift* in a single branch of AR. Shifts make it possible to change word order and thus ‘recover’ from incorrect word order that may be incurred by an AR deleting step. The *shift operation* is such a change in a D-tree when (i) the ordering of all nodes except for one is preserved, and (ii) the edges are preserved (connecting ‘identical’ pairs of nodes with respect to their lexical parts). It means that both the original D-tree T and the modified one belong to the same set $DPE(T)$.

Let T be a D-tree, $T \notin CT$. Our goal is to find – if possible – a modified D-tree T' such that T' is a correct surface tree (i.e., $T' \in CT$) and T' is DP-equivalent to T (i.e., $T' \in DPE(T)$) by applying as small number of shift operations as possible.

4 Pilot Study on Czech Sentences

4.1 Description of the Experiment

In our experiment we are focusing on a development of a measure of word order freedom based upon the notions defined or introduced in previous sections and upon the data available in the Prague Dependency Treebank (PDT). Using the treebank is very important for a number of reasons, especially:

- It gives us a large representative sample of syntactically annotated sentences which can be used for the development and testing of the proposed measure.
- The annotated data from the treebank are independent of our experiments and thus we don’t have to discuss whether a particular sentence should have been annotated in this or that way, we take the annotation as a given objective fact.

The analysis of data from the Prague Dependency Treebank gave very interesting results from the point of view of word order freedom. According to [2], almost one quarter of sentences from PDT 1.0 contains non-projective constructions. More precisely, among the 73 088 sentences of training data in PDT 1.0, there are 23.2 % non-projective ones, i.e., 16 920 sentences. These sentences can be divided into the following categories:

1. non-projectivity given by a modal verb (or a verb with similar properties) with an infinite complementation ... 5 696 non-projectivities in 4 708 trees;
2. compound prepositions ... 5 894 non-projectivities in 5 388 trees;
3. adjectives ... 963 non-projectivities in 922 trees;
4. comparatives ... 379 non-projectivities in 369 trees;
5. others ... 10 938 non-projectivities in 8 045 trees.⁷

Out of these categories, verbal non-projectivities are very interesting from the linguistic point of view while the second most frequent category, compound prepositions, contains mostly very technical and linguistically irrelevant constructions. The remaining categories are less frequent (the last category contains large number of various phenomena with a low frequency). Let us therefore concentrate our efforts on verbal non-projectivities in the subsequent text.

The sentences with verbal non-projectivities demonstrate that Czech is a language with a high degree of word order freedom. It is usually possible to reduce the number of non-projective constructions to zero while preserving the correctness of a sentence simply by reordering the words in the sentence. The most regular exception from this rule are sentences containing clitics.

Clitics constitute a certain fixed point in a typical Czech sentence. They are usually located on the sentence second (Wackernagel's) position and thus they are both a frequent source of non-projective constructions and an obstacle which requires special treatment when we attempt to reduce the number of non-projectivities. The situation is even more complicated because the sentence second position may contain a larger number of clitics whose mutual order is not arbitrary in some cases. Let us consider the following example (taken from [3]):

- (2) *Opravit jsem se mu to včera snažil marně.*
 to-repair aux-1-sg REFL him it yesterday tried fruitlessly
 'I tried to repair it for him yesterday without success.'

In this sentence we may notice that the clitics are the main reason why the sentence is non-projective. While *jsem* 'aux-1-sg' and *se* REFL depend on the verb *snažil* 'tried', the pair of clitics *mu* 'him' and *to* 'it' depend on the infinite verb *opravit* 'to repair'. In this special case it is possible to make the sentence projective while preserving its correctness and all dependencies and morphological properties of all words by means of either swapping the two verbs and shifting the adverb slightly forward: *Snažil jsem se mu to marně včera opravit.*; or by swapping the pairs of clitics: *Opravit mu to jsem se včera snažil marně.*

⁷ We would like to express our special thanks to Daniel Zeman who has provided us with the data, see also

<http://ufal.mff.cuni.cz/~zeman/projekty/neprojs/index.html>

These examples actually show that our method might provide a clue for further investigation of the degree of word order freedom. The number of shifts or swaps performed in the course of the analysis by reduction with the purpose of preserving all important factors (grammatical correctness, morphological and syntactic information, dependency relations) in every step of the analysis might reflect the word order freedom of a particular language.

4.2 Evaluation

In order to obtain a deeper insight into the problem of mutual relationship between clitics, holes (non-projective constructions) and shifts necessary for the preserving a sentence correctness in the course of AR, we have chosen 100 non-projective sentences from the PDT, the portion with non-projectivity given by a modal verb (see above) and manually evaluated them. As we are concentrating primarily on the clitic / (non-)projectivity interplay here (and we want to eliminate other language phenomena) we have simplified the input sentences using AR in such a way that only words related to these phenomena are preserved. Note that discontinuous dependencies are allowed, i.e. dependent word in a position distant from its governor may be deleted.

The results of this evaluation are summarized in Tab. 1, which indicates relations between the number of clitics, the number of holes and the minimal necessary number of shifts. The table shows that although clitics are usually a primary reason why a sentence contains non-projective constructions, there surprisingly seems to be no correlation between the number of holes (number of individual non-projectivities), the number of clitics and the number of shifts. It is also quite interesting that the minimal number of necessary shifts does not exceed one regardless of the number of clitics or the number of holes.

Table 1. Sample non-projective sentences from PDT 1.0 – basic characteristics

# clitics	# sentences	# holes	# shifts	comments
0 clitics	21 / 14 / 3	1	0	main / dependent clause / question
	2	0	0	annot. error
	1	1	0	error in raw text
	1	2	1	two dependencies
1 clitic	17 / 17	1	0	main / dependent clause
	10 / 2 / 1	1	1	main / dependent clause / question
	1	2	0	main clause
	1	2	1	main clause
2 clitics	2	1	0	main clause
	6 / 1	1	1	main / dependent clause
	1	2	1	
3 clitics	1	1	0	
	1	1	1	

Table 2. Sample projective sentences from PDT 1.0 – basic characteristics

# clitics	# sentences	# holes	# shifts
1 clitic	11	0	0
	34	0	1
2 clitics	5	0	0

This observation inspired a second experiment – if the number of holes is irrelevant, how does the relationship between the number of clitics in projective sentences and the necessary number of shifts look like? The results of this experiment are presented in Tab. 2. In this case we have taken 50 randomly chosen projective sentences with clitics from the PDT and we have performed the same evaluation as in the previous experiment. The results are also quite interesting.

First, the number of clitics in projective sentences is generally lower. This supports the claim that clitics constitute one of the primary sources of non-projectivities in Czech. If the sentence contains more than two clitics, it is highly probable that it contains non-projective constructions as well.

Second, even in projective sentences it is often necessary to shift the words during the course of the analysis by reduction, otherwise some of the general constraints would be violated (usually the correctness preserving constraint). This actually supports the claim that neither the number of holes nor the number of clitics in a sentence correlates with the necessary number of shifts. The number of shifts indicates that clitics have rigid positions in Czech; however, the measure proposed does not sufficiently cover cases where the words with fixed positions do not enforce a shift during the reduction. Moreover, the fact that in both experiments it took maximally 1 shift to make the sentence projective in the course of the analysis by reduction indicates that this measure of word order freedom is rather too simplistic.

5 Conclusion

The results of the two experiments presented in this paper indicate that the number of shifts is an important factor providing different information than already existing measures reflecting the complexity of word order of individual sentences. However, the granularity of the proposed measure is too low and it will definitely require further refinement in the future. We plan to consider further characteristics of the delete and shift operations, esp. dis/continuous dependencies (whether dependent word is adjacent to its governing word or not) and a type of shifts (a shift of a verb / a shift across a verb).

The experiments also helped us to analyze the treebank data from a new perspective and to gain an increased insight into the phenomena responsible for higher complexity of syntactic structures of sentences of languages with higher degree of word order freedom.

Future research should aim at two important goals – to repeat the experiments with higher number of sentences (and different phenomena causing

non-projectivity) and, primarily, to extend them to a different language. The comparison between languages with lower degree of word order freedom and higher number of word-order constraints and Czech would definitely bring interesting results.

References

1. Gerdes, K., Kahane, S.: Defining dependencies (and constituents). In: Gerdes, K., Hajičová, E., Wanner, L. (eds.) *Proceedings of the International Conference of Dependency Linguistics, Depling 2011, Barcelona*, pp. 17–27 (2011)
2. Hajičová, E., Havelka, J., Sgall, P., Veselá, K., Zeman, D.: Issues of Projectivity in the Prague Dependency Treebank. *The Prague Bulletin of Mathematical Linguistics* 81, 5–22 (2004)
3. Hana, J.: *Czech Clitics in Higher Order Grammar*. Ph.D. thesis, The Ohio State University (2007)
4. Holan, T., Kuboň, V., Oliva, K., Plátek, M.: On Complexity of Word Order. *Les Grammaires de Dépendance – Traitement Automatique des Langues (TAL)* 41(1), 273–300 (2000)
5. Kunze, J.: Die Auslassbarkeit von Satzteilen bei koordinativen Verbindungen im Deutschen. In: *Schriften zur Phonetik, Sprachwissenschaft und Kommunikationsforschung*, vol. 14. Akademie-Verlag, Berlin (1972)
6. Lopatková, M., Plátek, M., Sgall, P.: Towards a Formal Model for Functional Generative Description: Analysis by Reduction and Restarting Automata. *The Prague Bulletin of Mathematical Linguistics* 87, 7–26 (2007)
7. Marcus, S.: Sur la notion de projectivité. *Zeitschrift Fur Mathematische Logik und Grundlagen Der Mathematik* 11(1), 181–192 (1965)
8. Mel'čuk, I.A.: *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany (1988)
9. Otto, F.: Restarting Automata and Their Relation to the Chomsky Hierarchy. In: Ésik, Z., Fülöp, Z. (eds.) *DLT 2003. LNCS*, vol. 2710, pp. 55–74. Springer, Heidelberg (2003)
10. Plátek, M., Mráz, F., Lopatková, M. (In)Dependencies in Functional Generative Description by Restarting Automata. In: Bordihn, H., Freund, R., Hinze, T., Holzer, M., Kutrib, M., Otto, F. (eds.) *Proceedings of NCMA 2010*. books@ocg.at, vol. 263, pp. 155–170. Österreichische Computer Gesellschaft, Wien (2010)
11. Sgall, P., Hajičová, E., Panevová, J.: *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht (1986)

Core-Periphery Organization of Graphemes in Written Sequences: Decreasing Positional Rigidity with Increasing Core Order

Md. Izhar Ashraf and Sitabhra Sinha

The Institute of Mathematical Sciences, CIT Campus, Taramani, Chennai 600113, India
{ashraf, sitabhra}@imsc.res.in

Abstract. The positional rigidity of graphemes (as well as words considered as single units) in written sequences has been analyzed in this paper using complex network methodology. In particular, the information about adjacent co-occurrence of graphemes in a corpus has been used to construct a network, where the nodes represent the distinct signs used. Core-periphery structure of this network has been uncovered using k -core decomposition technique suitably generalized for directed networks. This allows identification of a core signary or “graphem-ome” of the corresponding writing system, i.e., the group of frequently co-occurring graphemes. The distribution of the frequency with which such signs occur at different positions in a sequence (e.g., at the beginning or at the end or in the middle) shows that while signs belonging to the periphery often occur only at specific positions, those in the innermost cores may occur at many different positions. This is quantified by using a positional entropy measure that shows a systematic increase with core order for the different databases used in this study (corpus of English, Chinese and Sumerian sentences as well as a database of Indus civilization inscriptions).

Keywords: linguistic networks, core-periphery organization, positional entropy, core signary.

1 Introduction

Complex networks pervade all aspects of life, including language (Lamb, 1970; Biemann and Quasthoff, 2009; Choudhury and Mukherjee, 2009). Analysis of linguistic texts using the graph-theoretic perspective may reveal unexpected patterns such as the existence of a “small-world effect” in networks where pairs of words have been connected based on their co-occurrence in a corpus (Ferrer i Cancho and Sole, 2001). It has been hypothesized that such network properties may be an outcome of the evolutionary process undergone by lexicons. Theoretical models for such word co-occurrence networks have been proposed, at least one of which came up with the result that languages may have a “kernel lexicon” that remains relatively invariant throughout the evolution of the language (Dorogovtsev and Mendes, 2001). This forms an intriguing analogy to the concept of “core signary”, often used in the context of undeciphered or partially deciphered ancient inscriptions (Palaima *et al.*, 2000),

which is used to indicate the set of very frequently occurring signs. Identification of core signary may help in uncovering significant clues to the nature of the underlying language by focusing the attention of the decipherer on a subset of inscriptions that may have underlying well-defined rules of construction.

In this paper, we propose using the concept of core-periphery organization in complex networks, earlier used in different contexts (Holme 2005), including networks generated by web-search queries (Saha Roy *et al.*, 2011), to identify a *graphemome* or a core grapheme network for a corpus of written inscriptions. We define such a grapheme core to be a group of frequently co-occurring written signs or graphemes. Depending on the writing system used, these graphemes could be either letters (for alphabets) or complex logograms (for ideographic systems such as Chinese or logo-syllabic systems such as Sumerian cuneiform). We also analyze for comparison a corpus of English sentences, where the basic elements are taken to be the words (rather than the alphabetic characters) as in many cases a single grapheme in an ideographic or logo-syllabic system can represent an entire word.

We map each corpus to a network, whose nodes are the distinct graphemes (or words) and link them if a pair occurs at adjacent positions in a written sequence in the database used. The network can be either undirected or directed, depending on whether we take into account the direction of reading a sequence to assign a directed link between a pair of adjacent signs. We use k -core decomposition technique to peel away successive layers of k -order cores starting from the outermost periphery, eventually arriving at the innermost core. Our results show that the core-periphery organization of sequences written using alphabetic systems fundamentally differ from that of ideographic or logo-syllabic systems, in that the former has almost its entire repertoire of graphemes in its innermost cores. This is probably related to the fact that the co-occurrence of different alphabetic signs is relatively unrestricted (i.e., any alphabetic character can occur next to any other character, although with different frequencies). More importantly, we find that the frequency of signs belonging to the periphery is consistently lower than that of signs belonging to the inner cores. This suggests a close correspondence with the idea of a core signary (mentioned earlier) which is defined in terms of frequency of occurrence of the signs.

Our most important finding is that there is a correlation between the positional freedom of a grapheme to occur at any particular position in a sequence (e.g., either at the beginning or at the end or somewhere in the middle) and its core-order membership. In particular, we find that while signs belonging to the outer periphery are relatively rigid in their position of occurrence, those in the inner cores are likely to appear at many different positions. We quantify this freedom using the concept of positional entropy. By definition, the more freedom a sign has in occurring anywhere in a sequence, the larger is its positional entropy. We observe that in almost all cases, the average positional entropy increases systematically with core order regardless of the writing system being considered. It is possible that the core-decomposition thus allows segregation of graphemes into those that are used only in restricted contexts and those that are of much wider or general applicability. It may provide a useful tool for deciphering inscriptions written in hitherto unknown systems, such as that of the Indus Valley civilization (Parpola 1994).

2 Methods and Materials

The core-periphery analysis has been carried out on four different databases consisting of sequences written in different systems, viz., alphabetic (English sentences), ideographic (Chinese sentences), logo-syllabic (Sumerian cuneiform inscriptions) and unknown (undeciphered Indus inscriptions).

2.1 Database Description

English: This is a corpus of sentences collated from the e-text of *Adventures of Sherlock Holmes* obtained from <http://gutenberg.org>, having 6788 sentences that are composed of 60 words or less and comprising 8124 unique words.

Chinese: This is a corpus of multi-sign words and phrases constructed from the entries of an online Chinese-English dictionary CC-CEDICT (<http://www.mdbg.net/chindict/chindict.php?pahe=cc-cedict>) having 10351 multiple sign sequences comprising 3303 unique graphemes.

Sumerian Cuneiform: This is a corpus of Sumerian texts collected from ETCSL (Electronic Text Corpus of Sumerian Literature, <http://etcsl.orinst.ox.ac.uk>) with superscripts considered as separate graphemes. There are 15390 single-line sequences comprising a total of 1503 unique graphemes.

Indus inscriptions: This is a corpus of Indus inscriptions obtained by permission from B. K. Wells (Sinha et al, 2011). It consists of 1820 single-line sequences comprising 593 unique graphemes.

2.2 k-core Decomposition

To obtain a decomposition of the network into a densely connected node and a sparsely connected periphery, we use a pruning algorithm that recursively removes all nodes with degree (i.e., number of connections) less than k . The resulting k -th order core (in short, k -core) is the subnetwork of all nodes that are connected to at least k other nodes in the subnetwork. For example, the 1-core of a network is obtained simply by removing all isolated nodes and the 2-core is obtained by iteratively eliminating all nodes that are not part of a closed loop. One can carry out the process for all integer values of k up to a maximum value that cannot exceed the highest degree of the nodes comprising the network. We illustrate this method as applied to a set of sequences with an example in the Appendix. The method has been generalized to directed networks, where cores can be defined in two distinct ways, viz., in terms of the in-degree (i.e., incoming connections) and out-degree (i.e., out-going connections) (Chatterjee and Sinha, 2007). In this paper, we have analyzed the networks both as directed as well as undirected networks. We can also define a *shell* in terms of the cores of different orders as follows: the k -th shell of a network is defined to be the set of nodes that are the relative complement of nodes belonging to the $(k+1)$ -th core with respect to the k -th core (i.e., the set of nodes which are in the latter but not in the former).

2.3 Mapping Sequences of Varying Length to Constant Length

To compare the positional rigidity of different signs we need to be able to compare the relative positions of signs in different sequences. As the sequences may have differing lengths, we have mapped every sequence to a standardized length. For convenience, we defined the standardized length to be the length of the longest sequence L_{max} in the corpus. For English words, $L_{max} = 60$, for Chinese $L_{max} = 16$, for Sumerian $L_{max} = 35$ and for Indus inscriptions $L_{max} = 13$. For a sequence of length $L < L_{max}$, using a method adopted from (Fuls, 2010) we re-define the position of the graphemes such that the beginning grapheme is at position 1 and the ending grapheme is at position L_{max} , with the intermediate signs arranged in equidistant positions in between these two extremes. Thus if the original position of a sign in the sequence is i ($1 \leq i \leq L$), then its new position in the mapped sequence is $1 + [(i-1) * (L_{max}-1) / (L-1)]$ rounded to the nearest integer.

2.4 Positional Entropy

The positional entropy of each sign is computed by considering the probability of the sign to occur in each position j ($=1, \dots, L_{max}$) of the standardized length sequence. The probability $P_s(j)$ that a sign s appears in position j is obtained by dividing the number of sequences (mapped to the standard length as described above) in the corpus that it appears in that position by the total number of sequences (of length $L > 1$) in which it appears. Once these probabilities are obtained, the positional entropy of sign s is calculated as $-\sum P_s(j) \log_2 P_s(j)$, where the sum is over all values of j .

3 Results

During preliminary studies we had performed core-periphery analysis on a network of alphabetic characters and observed that almost all the signs belong to the innermost core. For example, for a corpus constructed out of Basic English words (obtained from the wikipedia entry on “Basic English”), the network consisted of 26 nodes corresponding to the letters of the alphabet. We observed that none of the nodes disappear from the cores of order upto $k=15$. Beyond this there is only one more core possible (i.e., $k_{max}=16$), so that the core-periphery organization of English letters appears to be almost a clique (a network where every node is connected to every other node, and the largest core order is equal to the total number of nodes).

However, when we analyze instead ideographic or logo-syllabic writing systems, it reveals a much richer core-periphery organization. As many of the graphemes in these writing systems can be thought of as representing single words, we compare our analysis with a network of English words (rather than alphabetic characters as above). We observe that in almost all cases, there is a smooth or continuous decrease in the size of the k -core as a function of the core order instead of the discontinuous decrease from N (the total number of nodes) to almost 0 as mentioned earlier for the alphabetic character network.

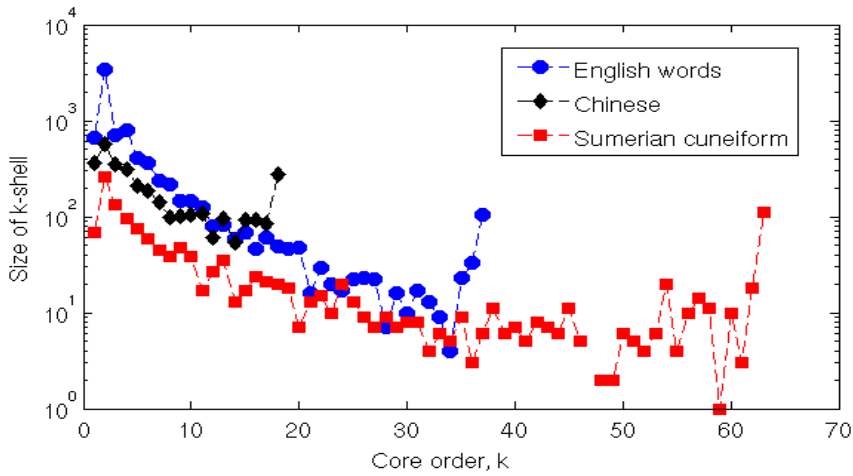


Fig. 1. Size of k -shell (i.e., the difference between the size of the k -core and the $(k+1)$ th core) shown as a function of the core order for the undirected networks of English words, Chinese and Sumerian cuneiform graphemes. The approximately smooth change in the variation of the shell size suggests a systematic monotonic decrease of the corresponding core sizes with core order.

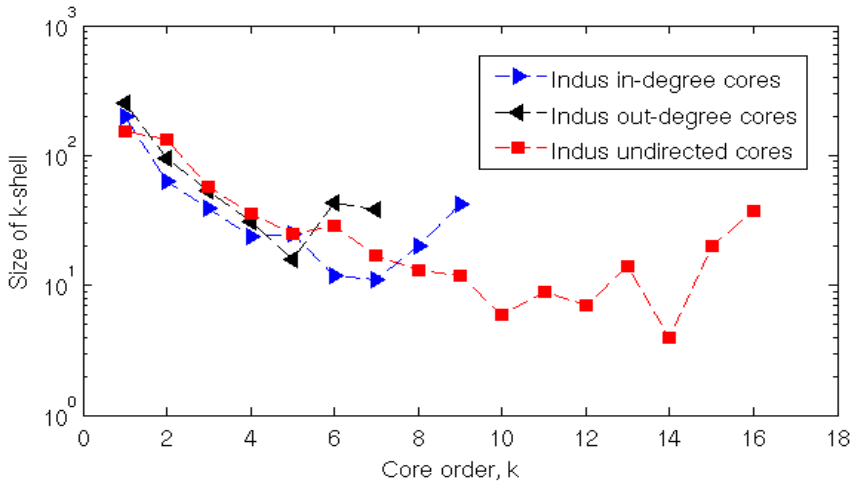


Fig. 2. Size of k -shell (i.e., the difference between the size of the k -core and the $(k+1)$ th core) shown as a function of the core order for the directed and undirected networks of the signs comprising the Indus inscriptions. The approximately smooth change in the variation of the shell sizes suggests a systematic monotonic decrease of the corresponding core sizes with core order.

Figs. 1-2 show the variation of shell size as a function of its order, k , for the different data-bases used in our study. As mentioned earlier one can consider both directed and undirected networks. Fig. 1 corresponds to the undirected networks for

English words and Chinese & Sumerian graphemes. Note that, there is almost a monotonic decrease in shell size as the core order increases, excepting very close to the innermost core. Indeed, there is a large increase in size at the highest core order indicating that there is a prominent core-periphery organization for all the data sets considered, each having a distinct set of inner core elements that is analogous to the “core signary” of the writing system. Fig. 2 corresponds to the undirected and directed networks (for both in-degree and out-degree cores) obtained from the database of Indus inscriptions which show similar properties.

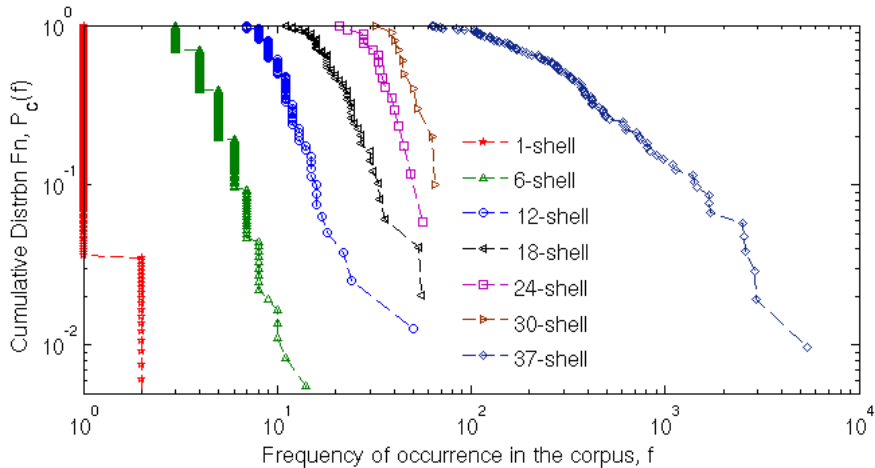


Fig. 3. Cumulative distribution function of frequency of occurrence of words in different shells for the English words database showing decreasing frequency as a function of the shell order. Similar patterns are seen for Chinese, Sumerian and Indus.

When we look at the frequency distribution of signs (e.g., Fig. 3 for English words) belonging to the different cores, we observe that for all databases considered, the signs in the periphery are the ones occurring with low frequency in the corpus, while those in the inner cores occur with very high frequency. As already mentioned, this offers a close connection to the concept of core signary which is defined in terms of frequently co-occurring signs.

Finally, we look at the positional entropy of the signs as a function of their core order k . As shown in Figs 4-7, for all the databases considered, the positional entropy increases logarithmically with the core order k [i.e., mean positional entropy for symbols occurring in the k -th shell $\sim \log(k)$]. This implies that for the outer periphery graphemes (or words, in the case of the English database) there is much higher positional rigidity or relatively less freedom in the position in a sequence where they can appear. By contrast, for an element in the inner periphery there is much less restriction and they can occur at almost any position within a given sequence. This indicates a significant qualitative difference (apart from the differences in the frequency of occurrences) in the signs or graphemes that appear at different cores.

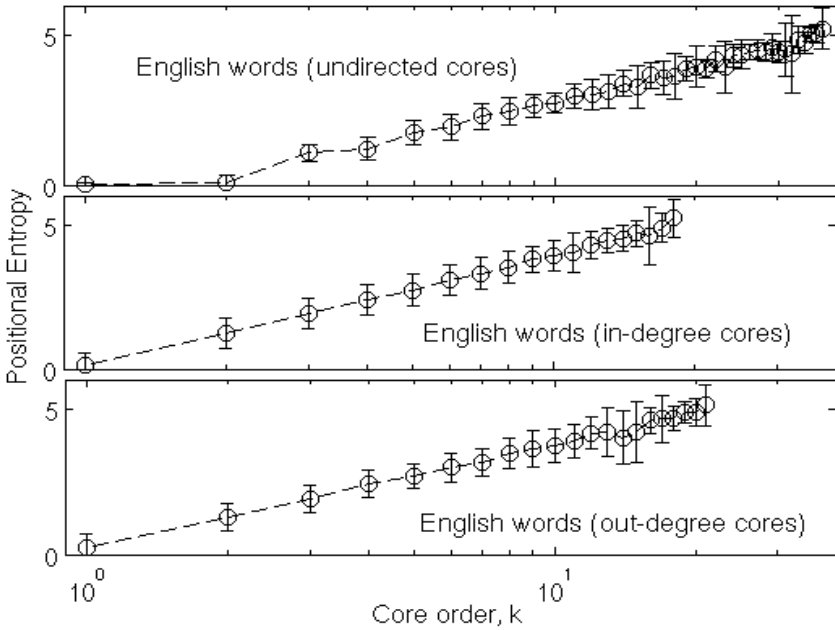


Fig. 4. Average positional entropy of words in English database as a function of the order k of the core in which they occur (shown in logarithmic scale). The bars indicate the standard deviation. For undirected networks (top), as well as, for the in-degree (center) and out-degree cores (bottom) in the directed networks of sign co-occurrence, the positional entropy is seen to increase with core order.

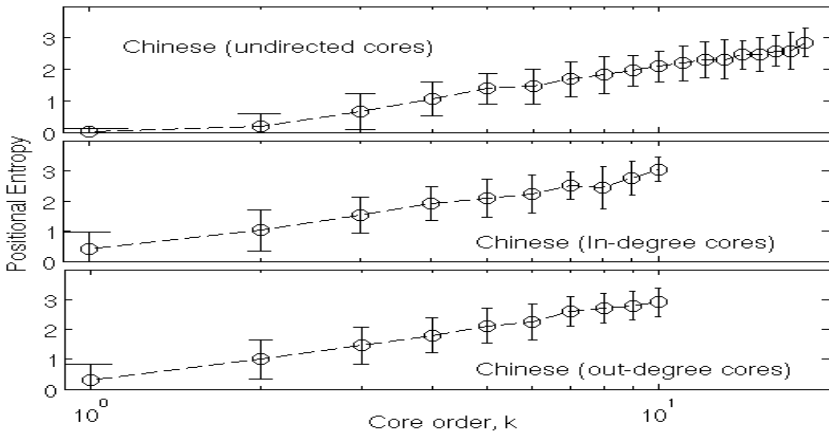


Fig. 5. Average positional entropy of logographic signs in Chinese as a function of the order k of the core in which they occur (shown in logarithmic scale). The bars indicate the standard deviation. For undirected networks (top), as well as, for the in-degree (center) and out-degree cores (bottom) in the directed networks of sign co-occurrence, the positional entropy is seen to increase with core order.

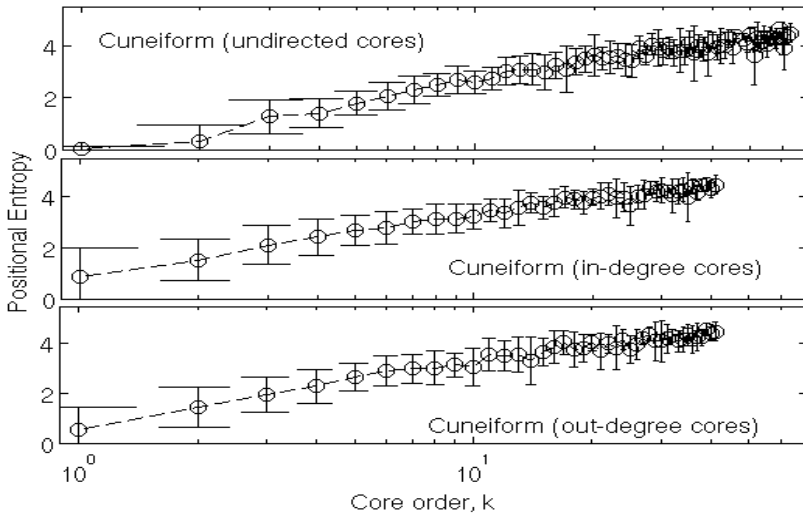


Fig. 6. Average positional entropy of logo-syllabic signs in Sumerian cuneiform as a function of the order k of the core in which they occur (shown in logarithmic scale). The bars indicate the standard deviation. For undirected networks (top), as well as, for the in-degree (center) and out-degree cores (bottom) in the directed networks of sign co-occurrence, the positional entropy is seen to increase with core order.

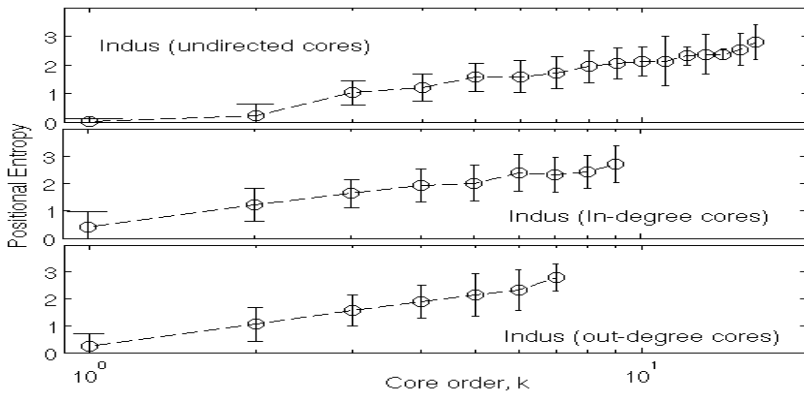


Fig. 7. Average positional entropy of the undeciphered signs in Indus inscriptions as a function of the order k of the core in which they occur (shown in logarithmic scale). The bars indicate the standard deviation. For undirected networks (top), as well as, for the in-degree (center) and out-degree cores (bottom) in the directed networks of sign co-occurrence, the positional entropy is seen to increase with core order.

4 Conclusions

We have analyzed four different databases from the perspective of complex network analysis. In particular we have looked at the core-periphery organization and identified an inner core in the network of grapheme relations. They constitute what we term as the “grapheme-ome”, or the fundamental group of co-occurring graphemes that are at the root of a given writing system. By using core decomposition techniques on both undirected and directed networks of signs relations we have classified the set of constituent signs into a number of disjoint classes in terms of the corresponding core-order. These groups of signs not only differ in terms of their frequency of occurrence (with the outer periphery signs being significantly less common than the ones as the inner cores) but also in terms of the freedom they have in appearing in different positions in a given sequence. While peripheral signs are rigidly localized in specific locations in a sequence (e.g., only appearing in the beginning or in the end or somewhere in the middle), core graphemes (or words) can appear in almost any position. This indicates qualitative differences in the nature of the elements belonging to the core and those belonging to the periphery, the former being more general while the latter being used in restricted contexts. Our findings appear to be consistent with the existence of phonotactic constraints on the possible combination of phonemes for at least some alphabetic writing systems. It may be intriguing to extend our methods to weighted networks, where the links between nodes are weighted by the conditional probability or some other statistical measure of significance of the sign pair.

We suggest that our results can be useful in deciphering inscriptions written in hitherto unknown writing systems. For example, in one of the most famous examples of deciphering an unknown script without the aid of any bilingual texts, viz., Linear B, one of the key steps was the establishment of the core signary of the system by Emmett L. Bennett in the late 1940s (Palaima *et al.*, 2000). This helped Michael Ventris to focus on the essential elements of the writing system, eventually achieving the breakthrough in 1952. Using the computational technique outlined here it should be possible to do an automated search for the core elements of an unknown script and then concentrate on understanding the rules of usage in this restricted set. As relatively uncommon elements (that are located in the periphery) are likely to have special rules of application different from the general rules that apply for the core elements, we believe that such a strategy would be more efficient in linguistic decipherment than trying to use the entire available corpus at the initial stage for inferring possible syntactic rules.

References

- Biemann, C., Quasthoff, U.: Networks generated from natural language text. In: Ganguly, N., et al. (eds.) *Dynamics on and of Complex Networks*, pp. 167–185. Birkhauser, Boston (2009)
- Chatterjee, N., Sinha, S.: Understanding the mind of a worm. *Progress in Brain Research* 168, 145–153 (2007)

- Choudhury, M., Mukherjee, A.: The structure and dynamics of linguistic networks. In: Ganguly, N., et al. (eds.) *Dynamics on and of Complex Networks*, pp. 145–166. Birkhauser, Boston (2009)
- Dorogovtsev, S.N., Mendes, J.F.: Language as an evolving word web. *Proceedings of the Royal Society of London B* 268(1485), 2603–2606 (2001)
- Ferrer i Cancho, R., Sole, R.V.: The small world of human language. *Proceedings of the Royal Society of London B* 268(1482), 2261–2265 (2001)
- Fuls, A.: Entwicklung einer geographisch-epigraphischen datenbank der indusschrift. In: Weisbruch, S., Kaden, R. (eds.) *Entwicklerforum Geodäsie und Geoinformationstechnik 2010*. Technische Universität, Berlin (2010)
- Holme, P.: Core-periphery organization of complex networks. *Physical Review E* 72(4), 046111(1-4) (2005)
- Lamb, S.M.: Linguistic and cognitive networks. In: Garvin, P. (ed.) *Cognition: A Multiple View*, pp. 195–222. Spartan Books, New York (1970)
- Palaima, T.G., Pope, E.I., Kent Reilly, F.: Unlocking the secrets of ancient writing. Catalogue of an exhibition in conjunction with the 11th International Mycenological Colloquium. The University of Texas at Austin (2000)
- Parpola, A.: *Deciphering the Indus Script*. Cambridge University Press, Cambridge (1994)
- Saha Roy, R., Ganguly, N., Chowdhury, M., Singh, N.K.: Complex network analysis reveals kernel-periphery structure in web search queries. In: *2nd International ACM SIGIR Workshop on Query Representation and Understanding (QRU 2011)*, pp. 5–8 (2011)
- Sinha, S., Izhar, A.M., Pan, R.K., Wells, B.K.: Network analysis of a corpus of undeciphered Indus civilization inscriptions indicates syntactic organization. *Computer Speech and Language* 25(3), 639–654 (2011)

Appendix: Example of Core Decomposition

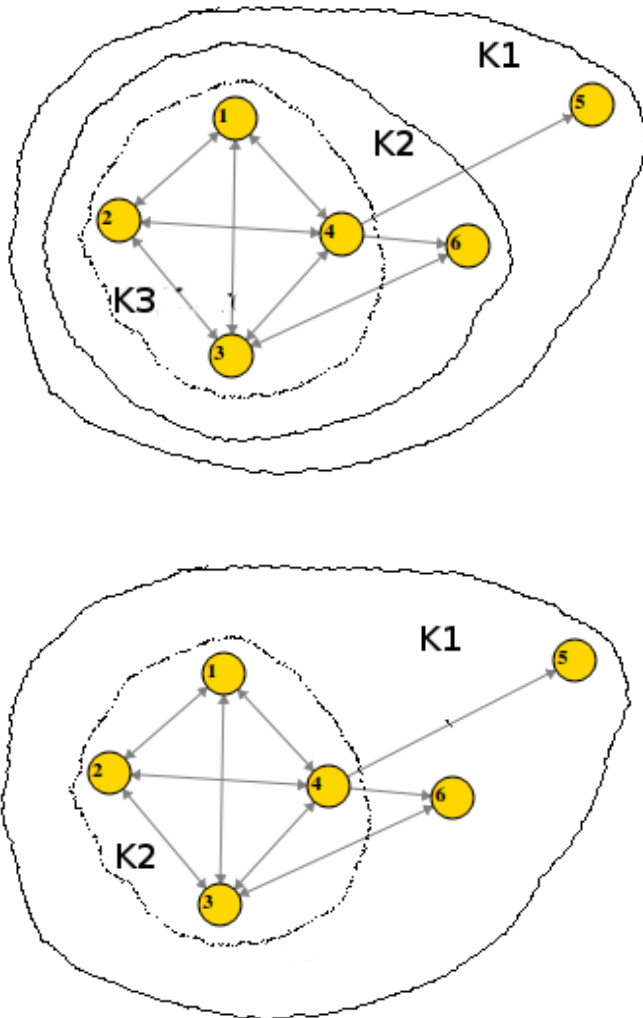


Fig. A.1. The core decomposition of a directed network obtained from a set of three sequences constructed out of 6 unique signs (see text) using (top) in-degree and (bottom) out-degree. The nodes 1-6 correspond to the signs a, b, c, d, e and f respectively.

Here we illustrate with an example the core decomposition of a network of signs constructed from a corpus of sequences. Let us consider a group of 3 sequences which comprise signs from the 6-element set $\{a, b, c, d, e, f\}$:

Sequence I : *bcd*e

Sequence II: *dfcbad*c*ac*f

Sequence III: *dabdb*

First, each of the six signs are mapped to integers 1-6 (i.e., $a \rightarrow 1$, $b \rightarrow 2$, etc.). These will label the six nodes of the network, with a pair of nodes connected if the two signs occur adjacent to each other in any sequence (i.e., the occurrence of “*ab*” in any sequence would mean that in the network node 1 will be connected to node 2 by a link directed from 1 to 2). Proceeding in this way, we can reconstruct a directed network, with the in-degree of a node indicating how many links point to it from other nodes while its out-degree indicates how many links from the node point to other nodes.

After constructing the network, we begin with the core decomposition by first obtaining the 1-core. For the in-degree (out-degree) core this is done by removing all nodes whose in-degree (out-degree, respectively) is less than 1. This gives us the first or outermost core (i.e., K1) containing the entire set of signs $\{a, \dots, f\}$. After this we obtain the in-degree (out-degree) 2-core by removing all nodes whose in-degree (out-degree, respectively) is less than 2. This gives us the second core (K2). Note that, for the in-degree core decomposition, node 5 (sign *e*) is part of the 1-core but not of the 2-core. Hence this node is the only member of the in-degree 1-shell (i.e., the relative complement of 2-core with respect to 1-core). Similarly, the two nodes 5 and 6 (corresponding to signs *e* and *f*) belong to the out-degree 1-shell. We can proceed in this manner to construct cores and shells of higher orders until finally we are left with a null set. The highest core order for which the core retains at least one node is the innermost core of the network (in this case, 3-core for in-degree and 2-core for out-degree).

Discovering Linguistic Patterns Using Sequence Mining

Nicolas Béchet¹, Peggy Cellier², Thierry Charnois¹, and Bruno Crémilleux¹

¹ GREYC Université de Caen Basse-Normandie
Campus II science 3

14032 Caen CEDEX, France

{nicolas.bechet, thierry.charnois, bruno.cremilleux}@unicaen.fr

² INSA Rennes/IRISA,

Campus de Beaulieu

35042 Rennes cedex, France

peggy.cellier@irisa.fr

Abstract. In this paper, we present a method based on data mining techniques to automatically discover linguistic patterns matching appositive qualifying phrases. We develop an algorithm mining sequential patterns made of itemsets with gap and linguistic constraints. The itemsets allow several kinds of information to be associated with one term. The advantage is the extraction of linguistic patterns with more expressiveness than the usual sequential patterns. In addition, the constraints enable to automatically prune irrelevant patterns. In order to manage the set of generated patterns, we propose a solution based on a partial ordering. A human user can thus easily validate them as relevant linguistic patterns. We illustrate the efficiency of our approach over two corpora coming from a newspaper.

1 Introduction

Due to the explosion of available textual data, the need for efficient processing of texts has become crucial for many applications; for instance, extraction of biological knowledge from biomedical texts, monitoring opinion from newspapers or forums. Natural Language Processing (NLP), and Information Extraction (IE) in particular, aim to provide accurate parsing to extract specific knowledge such as named entities (e.g., gene, person, company) and relationships between the recognized entities (e.g., gene-gene interactions). A common feature of the information extraction methods is the need for linguistic resources (grammars or linguistic rules). This paper deals with this problem and proposes a method for automatically discovering linguistic patterns.

Indeed, NLP approaches apply rules such as regular expressions for surface searching [10] or syntactic patterns [9]. However, these rules are handcrafted and thus those methods are time consuming and very often devoted to a specific corpus [11]. In contrast, machine learning based methods such as support vector machines or conditional random fields [13], are less time consuming than NLP methods. Although they provide good results, they still need many features. Moreover, their outcomes are not really understandable by a user, nor they can be used as linguistic patterns in NLP systems (because the produced models are numerical). Furthermore, the annotation process of training corpora requires a substantial investment of time, and cannot be reused in other

domains [11] (annotation of new corpora in new domains requires to repeat this time consuming work).

A promising avenue is the trade-off coming from the cross-fertilization of information extraction and machine learning techniques which aims at automatically learning linguistic resources such as lexicons or patterns [14]. Most of these symbolic approaches are supervised. RAPIER, a well-known system based on inductive logic programming, learns information extraction rules [3] but uses annotated corpora difficult to acquire as previously explained. A few of unsupervised approaches have been designed: one of these earliest works presents a method to acquire linguistic patterns from plain texts but it needs a syntactic parsing [16]. Therefore, the quality of learned patterns stems from the syntactic process results. New works take advantage of an hybridization of data mining and NLP techniques. An advantage of data mining techniques is to enable the discovery of implicit, previously unknown, and potentially useful information from data [8]. For instance, *Cellier et al.* [4] aim at discovering linguistic rules to extract relationships between named entities in new corpora. That approach is not supervised and does not need syntactic parsing nor external resources except the training corpus. It relies on extraction of frequent sequential patterns where a sequence is a list of literals called *items*, and an item is a word (or its lemma) within textual data. A well-known limitation of data mining techniques is the large set of discovered patterns. It needs to be filtered or summarized in order to return only relevant patterns. In the sequel, we present how we address this problem thanks to constraints and partial order.

The contribution of this paper is twofold. First, we improve works such as [4] by being able to handle sequences of *itemsets* instead of *single-items*. That means that a word can be represented by a set of features conveying several pieces of information (e.g., words, lemma) and not only a single information. Thus the extracted patterns combine different levels of abstraction (e.g., words, lemma, part of speech tags) and express information according to different levels of genericity, for example $\langle\langle\textit{champion NOUN}\rangle\rangle$ (*of PRP*) (*the DET*) (*world NOUN*) and $\langle\langle\textit{champion NOUN}\rangle\rangle$ (*PRP*) (*DET*) (*NOUN*) (see Section 2.3 for details). We have developed an algorithm for discovering such sequential patterns under constraints. Indeed, constraints enable to add user knowledge into the discovery process in order to give prominence to the most significant patterns. Secondly, we tackle the problem of pattern selection by proposing a tool allowing a user to easily navigate within the pattern space and validate sequential patterns as linguistic patterns. The navigation and validation take advantage of the partial order between patterns.

We apply our approach on learning linguistic patterns for discovering phrases denoting judgment or sentiment in French texts, and more generally qualification as given in Table 1 and called appositive qualifying phrases. It is important to note that our approach is not dedicated to a specific kind of linguistic patterns nor a specific language, but it can easily be adapted to other information extraction applications (e.g., relationships between named entities) or other languages. Indeed, our method is based on sequence mining techniques which are not language-dependent.

In the remaining of the paper, Section 2 introduces the method to extract sequential patterns and validate them. Section 3 presents and discusses experiments about appositive qualifying phrases.

Table 1. Excerpt of sequential database for French texts: “*homme de culture*” (*intellectual man*), “*femme de conviction*” (*woman of conviction*), “*charismatique et ambitieux*” (*charismatic and ambitious*), “*réputé pour sa cruauté*” (*famous for his violence*)

sid	Sequence
1	$\langle\langle\text{hommes homme NOUN}\rangle\rangle(\text{de PRP})(\text{culture NOUN})$
2	$\langle\langle\text{femmes femme NOUN}\rangle\rangle(\text{de PRP})(\text{conviction NOUN})$
3	$\langle\langle\text{charismatique ADJ}\rangle\rangle(\text{et KON})(\text{ambitieux ADJ})$
4	$\langle\langle\text{réputé réputer VER pper}\rangle\rangle(\text{pour PRP})(\text{sa son DET POS})(\text{cruauté cruauté NOUN})$

2 Extraction of Linguistic Patterns

Our approach is a two step method. First we extract sequential patterns thanks to our sequence mining algorithm. Secondly, we organize them in a data structure according to a partial order so that a linguist expert can easily validate extracted sequential patterns as linguistic patterns. More precisely, Section 2.1 introduces background knowledge about sequential patterns. Then we explain how constraints are at the core of the process to produce relevant candidate linguistic patterns (cf. Section 2.2). Finally, Section 2.3 presents the validation step.

2.1 Sequential Pattern Mining

Sequential pattern mining is a well-known data mining technique introduced in [1] to find regularities in a sequence database. There is a lot of algorithms to extract sequential patterns [19,21,20,22]. That point is discussed in Section 2.2.

In sequential pattern mining, an *itemset* I is a set of literals called *items*, denoted by $I = (i_1 \dots i_n)$. For example, (homme NOUN) is an itemset with two items: *homme* and *NOUN*. A *sequence* S is an ordered list of itemsets, denoted by $s = \langle I_1 \dots I_m \rangle$. For instance, $\langle\langle\text{hommes homme NOUN}\rangle\rangle(\text{de PRP})(\text{culture NOUN})$ (coming from Table 1) is a sequence of three itemsets. A sequence $S_1 = \langle I_1 \dots I_n \rangle$ is *included* in a sequence $S_2 = \langle I'_1 \dots I'_m \rangle$ if there exist integers $1 \leq j_1 < \dots < j_n \leq m$ such that $I_1 \subseteq I'_{j_1}, \dots, I_n \subseteq I'_{j_n}$. The sequence S_1 is called a *subsequence* of S_2 , and we note $S_1 \preceq S_2$. For example, $\langle\langle\text{NOUN}\rangle\rangle(\text{de PRP})$ is included in $\langle\langle\text{hommes homme NOUN}\rangle\rangle(\text{de PRP})(\text{culture NOUN})$. A sequence database SDB is a set of tuples (sid, S) , where sid is a sequence identifier and S a sequence. For instance, Table 1 depicts a sequence database of four sequences. A tuple (sid, S) contains a sequence S_1 , if $S_1 \preceq S$. The *support* of a sequence S_1 in a sequence database SDB , denoted $sup(S_1)$, is the number of tuples in the database containing S_1 ¹. For example, in Table 1 $sup(\langle\langle\text{NOUN}\rangle\rangle(\text{de PRP})) = 2$, since Sequences 1 and 2 contain $\langle\langle\text{NOUN}\rangle\rangle(\text{de PRP})$. A *frequent sequential pattern* is a sequence such that its support is greater or equal to a given support threshold $minsup$.

The set of frequent sequential patterns can be very large. Condensed representations, such as closed sequential patterns [21], have been proposed in order to eliminate redundancy without loss of information. A frequent sequential pattern S is closed if there is

¹ Note that the *relative support* is also used: $sup(S_1) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \wedge (S_1 \preceq S)\}|}{|SDB|}$.

no other frequent sequential pattern S' such that $S \preceq S'$ and $sup(S) = sup(S')$. For instance, with $minsup = 2$, the sequential pattern $\langle(NOUN)(NOUN)\rangle$ from Table 1 is not closed whereas $\langle(NOUN)(de PREP)(NOUN)\rangle$ is closed.

The constraint-based pattern paradigm [6] brings useful techniques to express the user's interest in order to focus on the most promising patterns. A very well-used constraint is the frequency. A sequence S is frequent if and only if $sup(S) \geq minsup$ where $minsup$ is a threshold given by a user. However, it is possible to define many other useful constraints such as the gap constraint. A gap is a sequence of itemsets which may be skipped between two itemsets of a sequence S . $g(M, N)$ represents a gap whose size is within the range $[M, N]$ where M and N are integers. The range $[M, N]$ is called a *gap-constraint*. A sequential pattern satisfying the gap-constraint $[M, N]$ is denoted by $P_{[M,N]}$. It means there is a gap $g(M, N)$ between every two neighbor itemsets of $P_{[M,N]}$. For instance, in Table 1, $P_{[0,2]} = \langle(PR P)(NOUN)\rangle$ and $P_{[1,2]} = \langle(PR P)(NOUN)\rangle$ are two patterns with gap constraints. Indeed, $P_{[0,2]}$ matches three sequences (1, 2 and 4) whereas $P_{[1,2]}$ matches only Sequence 4.

2.2 Algorithm to Extract Sequential Patterns

We present in this section our algorithm mining the closed sequential patterns of itemsets under constraints. There are already in the literature many algorithms to extract sequential patterns (e.g. GSP [19], SPADE [22], PrefixSpan [15]) or closed sequential patterns (e.g. CloSpan [21], BIDE [20]). But, to the best of our knowledge, there is no algorithm mining closed sequential patterns made of itemsets under constraints able to take into account the field of knowledge. In this paper, we address this open issue by proposing an algorithm mining sequential patterns made of itemsets under various constraints.

Adding constraints to the sequential pattern mining process is not trivial. The combination of constraints and the closure must be properly managed [2] in order to get the correct condensed representations of patterns with respect to the constraints. That is why our algorithm considers the closure after applying constraints to provide the pattern condensed representation. More precisely, sequential patterns satisfying the frequency and gap constraints are firstly produced, then the closed patterns are computed. Details of the algorithm are not given in this article because it is out of the scope of the paper.

We introduce the *begin_with* constraint which is very useful on textual data. A sequential pattern P satisfies the *begin_with* constraint if there is at least one sequence from SDB having its first itemset containing the first itemset of P . For instance, the sequential pattern $\langle(NOUN)(PR P)(culture NOUN)\rangle$ satisfies the *begin_with* constraint in SDB (cf. Table 1) because its first itemset $(NOUN)$ belongs to the first itemset of Sequence 1 $\langle(hommes homme NOUN)(de PR P)(culture NOUN)\rangle$. This constraint is precious to highlight appositive qualifying phrases. This one means that the appositive qualifying phrases has to appear at the beginning of a sequence. Moreover, we use a gap constraint of $g(0, 0)$ because appositive qualifying phrases are often made up of contiguous elements, which means that extracted patterns need to have contiguous itemsets according to the original sequences.

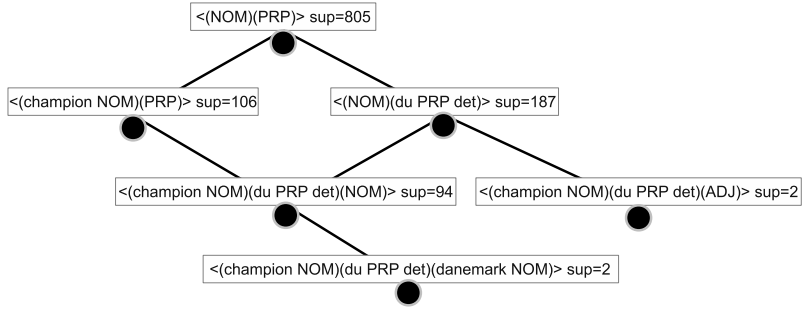


Fig. 1. Excerpt of the partial order on the patterns extracted from a corpus

2.3 Validation of Sequential Patterns as Linguistic Patterns

Constraints and closure reduce the set of extracted sequential patterns by pruning irrelevant patterns. Nevertheless, the number of extracted patterns can remain high. It is thus difficult for a human expert to validate them by hand as relevant linguistic patterns.

However, the set of extracted sequential patterns is partially ordered. Indeed, some patterns are more specific than others. For example, $\langle\langle\textit{champion NOUN} \textit{ du PRP det} \textit{ danemark NOUN}\rangle\rangle$ is more specific than $\langle\langle\textit{champion NOUN} \textit{ du PRP det} \textit{ NOUN}\rangle\rangle$. Thus the sentences matched by the pattern $\langle\langle\textit{champion NOUN} \textit{ du PRP det} \textit{ danemark NOUN}\rangle\rangle$ are also matched by the pattern $\langle\langle\textit{champion NOUN} \textit{ du PRP det} \textit{ NOUN}\rangle\rangle$. Therefore, when an expert selects a sequential pattern to promote it as a relevant linguistic pattern, she does not have to take care of more specific ones. We propose to take advantage of that partial order to organize the sequential patterns in a data structure, in order to help an expert to explore and select sequential patterns as linguistic patterns. The data structure is given in the form of a Hasse diagram [5]. Figure 1 shows an excerpt of a Hasse diagram for six sequential patterns extracted from one of our corpora. Nodes are sequential patterns, and edges between nodes represent the partial order relation.

The Hasse diagram can be very large. That is why we propose to use Camelis [7], a tool which allows to navigate in partial orders. Figure 2 shows the Camelis interface. At the bottom part, the *navigation tree* displays the patterns. The partial order over the set of patterns is highlighted by the navigation tree. The navigation tree is not a tree structure but represents a partial order and a pattern can have several parents. It explains why in Figure 2 the pattern $\langle\langle\textit{champion NOUN} \textit{ du PRP det} \textit{ NOUN}\rangle\rangle$ appears twice in the navigation tree (this pattern has two parents). The number on the left of a pattern is the number of patterns which are more specific. For example, in Figure 2, 235 sequential patterns are more specific than the pattern $\langle\langle\textit{NOUN} \textit{ PRP}\rangle\rangle$. The support of $\langle\langle\textit{NOUN} \textit{ PRP}\rangle\rangle$ is 805 meaning that in the learning corpus 805 phrases contain a noun followed by a preposition.

At the top, the query view displays the current query. In Figure 2, the query is “not Valid Linguistic Pattern”, i.e. the displayed patterns are the patterns not already selected

² <http://www.irisa.fr/LIS/ferre/camelis/index.html>

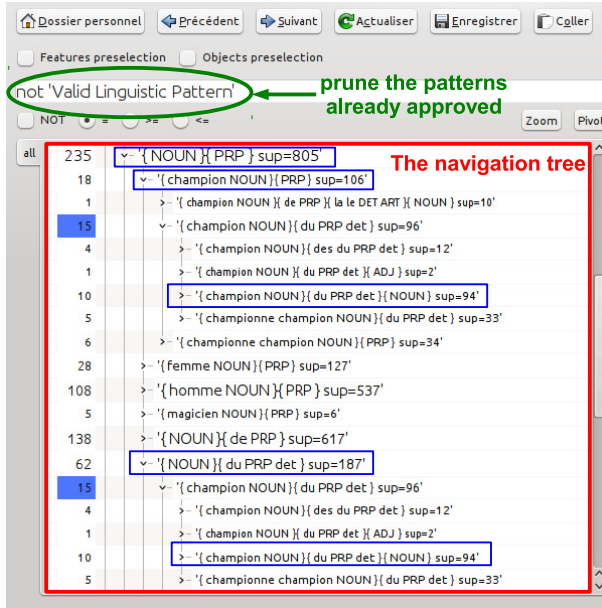


Fig. 2. Example of navigation from Camelis in order to validate linguistic patterns

as relevant linguistic patterns. Indeed, when exploring the patterns the expert may add some information about the patterns. The two main advantages of the process are: first, it enables the user to easily navigate in the sequential pattern set in order to validate them and, secondly, it allows to prune patterns without interest (i.e. sequential patterns already selected as linguistic patterns or patterns identified as not linguistic patterns) and thus reduce the exploration space. If a pattern P is selected as a relevant linguistic pattern, all more specific patterns than P are filtered out, i.e. these patterns do not have to be explored because the phrases matched by them are also matched by P .

3 Application: The Appositive Qualifying Phrases

3.1 Appositive Qualifying Phrases

In the opinion analysis framework, one crucial task is the extraction of phrases expressing judgement or qualification (the distinction is out of our scope). In the remaining of the paper, we call that kind of phrases: *appositive qualifying phrases*. Those phrases check some syntactic criteria: they have a relative free position in the sentence ; they are bounded by punctuation ; they are compounded of contiguous words (see [12] for more linguistic details). Some examples are given below (in bold font):

- (1) *Mais, en politicien expérimenté, élu pour la première fois à la Knesset il y a trente-cinq ans, il a su résister aux roquettes de ses adversaires politiques. (But, as a real politician, elected for the first time at the Knesset 35 years ago, he managed to face his political opposant attacks.)*

- (2) **Ni trop sentimental, ni trop énergique, il maîtrise, avec une finesse quasi mozartienne, un lyrisme généreux.** (Neither very romantic, nor very energetic, he masters, with great delicatessen as Mozart's, a generous lyrism.)
- (3) **Militant mais opportuniste, franc-tireur mais habile, sociable mais anticonformiste, le directeur de l'Opéra de Paris sait manier les paradoxes pour parvenir à ses fins.** (Militant but opportunist, dynamic but rigorous, sociable but anti-conformist, the Paris Opera's director knows how to handle paradoxes in order to reach his goals.)

Jackiewicz [12] provides about 20 handcrafted linguistic patterns to automatically extract appositive qualifying phrases. Some examples of those patterns³ are:

- Nominal groups (NG): (det) N de NG
 - *Femme de tête, X* (stubborn woman, X);
 - *X, le maestro de la désinflation* (X, the master of deflation).
- Adverbs: *courageusement, X* (courageously, X);
- Prepositional groups: *en mauvaise posture, X* (in a bad shape, X);
- Adjectival groups: *imprévisible et fantasque, X* (unpredictable and little bit crazy, X);
- Participle groups : *réputé pour son caractère bourru, X* (known for his obstinated personality, X).

Obviously, the definition of those linguistic patterns by hand is a tedious task. It shows the interest of our approach. In the sequel, we describe the process to help the linguistic expert to discover linguistic patterns characterizing qualifying phrases.

3.2 Corpora Constitution

As there is no available corpus with qualifying phrases, we have built two corpora.

The first corpus, called AXIOLO, is a set of occurrences obtained with linguistic patterns coming from [12]. Patterns are applied on the articles of the French newspaper “*Le Monde*”, of the topic “*Portrait*” (*i.e.* profile), from July to December of 2002 (884 articles). The building process of this first corpus leads to corpus almost without noise.

The second corpus, called ARTS, is also generated from the French newspaper “*Le Monde*” from articles of the topic “*Arts*” in 2006 (3,539 articles). We first applied the Treetagger tool [18] on the corpus to split sentences in constituents bounded by punctuations⁴. Our method is tolerant regarding Treetagger errors. Actually, if a wrong tag commonly occurs, this tag impacts resulting patterns without disrupting the result quality. Then, we used heuristics to filter out irrelevant constituents from sequential patterns, the ones that have no qualification. For instance, a proposition with a conjugated verb, a circumstantial group of time, of space, a goal, a cause, a condition are irrelevant qualifying phrases. Applying heuristics consists of testing for instance if a verb occurs in a given phrase, or if there exists a temporal term such as “Monday”. The list of irrelevant terms is built according to the Leff lexicons [17]. We also manually added

³ In the examples, the *X* represents the subject of the qualification.

⁴ Treetagger is used with the original training set.

Corpus	# seq	# items	# avg. itemsets per seq.	# max. itemsets in seq.	# avg. items per seq.	# max. items in seq.
AXIOLO	4,063	4,135	3	17	7	43
ARTS	13,576	20,796	6	32	16	89

Fig. 3. Corpora Characteristics

to this list some of typical French expressions such as “*d’une part (from one hand)*”, “*en référence (as referred to)*”, and so on. Finally, using heuristics allows to remove 113,812 constituents from the 127,388 originals. The resulting corpus is partially noisy. We have manually evaluated 32% of noise from a sample of 1,000 phrases. Fig 3 gives the characteristics of corpora.

3.3 Extraction of Sequential Patterns

In order to extract the closed sequential patterns of itemsets, we used our algorithm with both gap (with $g(0, 0)$) and *begin_with* constraint (cf. Section 2.2). We have conducted 10 experiments for each corpus with a relative support threshold between 0.05% to 50%. With the AXIOLO corpus, it means that a sequential pattern is frequent as soon as it appears in respectively 2 to 2,031 sequences. With the ARTS corpus, a sequential pattern is frequent as soon as it appears in respectively 6 to 6,788 sequences.

Results indicate that high *minsup* values provide very generic patterns, with only grammatical categories in itemsets (i.e. without lemmas or inflected forms of a term). According to our application on the discovery of linguistic patterns of appositive qualifying phrases, it is more relevant to use a low *minsup* to obtain sequential patterns combining the different levels of word abstraction. However, a low *minsup* produces an high number of patterns. For instance, with $minsup = 0.05\%$, 8,536 patterns are extracted from the ARTS corpus.

The validation task of patterns is difficult because of the high number of extracted sequential patterns. It shows the interest of our method based on the partial order of patterns and the Camelis tool (cf. Section 2.3). For each sequential pattern, P , the set of phrases matched by P and coming from a given corpus are grouped and filtered together. A linguist can then easily check the set of phrases matched by a pattern, it is especially interesting with noisy corpora. Then the validation of sequential patterns as linguistic patterns becomes easier for a linguist.

3.4 Experimental Results

Runtime. Our first aim is to evaluate the gain of the integration of constraints in the mining algorithm. For that purpose, we measure the runtime of the *Clospan* algorithm proposed in Illimine⁵ which is a very competitive prototype, and the runtime of our algorithm. We did the process on 10 experiments with the ARTS corpus. Experiments were conducted with an Intel Core 2 Duo Processor T9600 with 8 GHz of RAM.

⁵ <http://illimine.cs.uiuc.edu/>

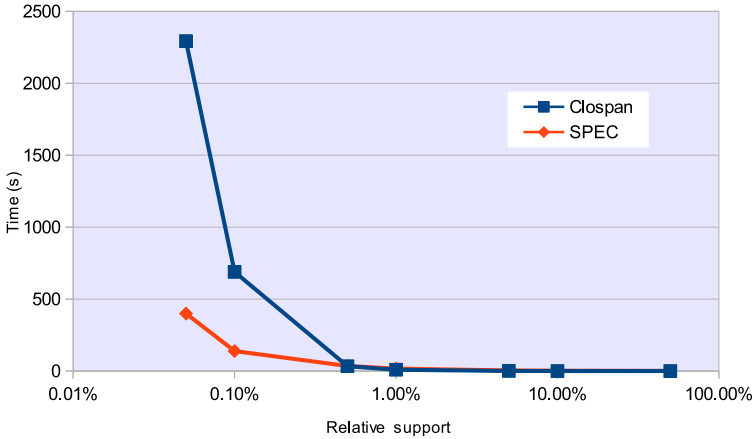


Fig. 4. Runtime comparing Clospan and our algorithm: SPEC

Figure 4 shows that our algorithm, SPEC (Sequential Pattern Extraction with Constraints) is much faster than *Clospan* for small relative supports. Note that *Clospan* only extracts the frequent closed sequential patterns and it does not integrate the gap constraint neither the *begin_with* constraint. When considering *Clospan*, we should take into account the time needed for the application of the constraints in a post-processing step. Therefore, the whole runtime of the process with *Clospan* would be higher.

Qualitative results. Evaluating an unsupervised method is a difficult task. A first way is to compare the results obtained by the method to a reference corpus, but such a corpus can be missing. A second way is to conduct an evaluation with an expert, but it requires a lot of time. In our case, we do not have a reference corpus on appositive qualifying phrases. Thus, we present our experimental results rather on the qualitative way, showing the originality and the usefulness of our method. Its success relies on the joint use of itemsets in sequences to catch several levels of information and the hierarchical property of patterns to validate them.

First, we want to evaluate the interest of sequential patterns made of itemsets. For that purpose, we have conducted on both corpora the mining of sequential patterns only made of items. We consider three kinds of sequences: the lemma, the grammatical category or combining the lemma and the grammatical category of a term. Results indicate that the obtained patterns are very specific or very generic. Examples of specific patterns are: $\langle \text{homme de conviction} \rangle$ (*man of conviction*) on lemma sequences ; $\langle \text{homme/NOUN de/PRP conviction/NOUN} \rangle$ on the combinations, which is almost the same sequence as the sequence obtained with lemma. With sequential patterns of items, we have the same level of abstraction for each word. For instance, we can have a generic pattern like $\langle \text{NOUN PRP NOUN} \rangle$ with only grammatical categories. Our experiments indicate that patterns with different levels of abstraction can be **only discovered by using itemsets**. For instance, the pattern $\langle \langle \text{hommes homme NOUN} \rangle \langle \text{de PRP} \rangle \langle \text{NOUN} \rangle \rangle$ uses inflected forms, lemmas, and grammatical categories.

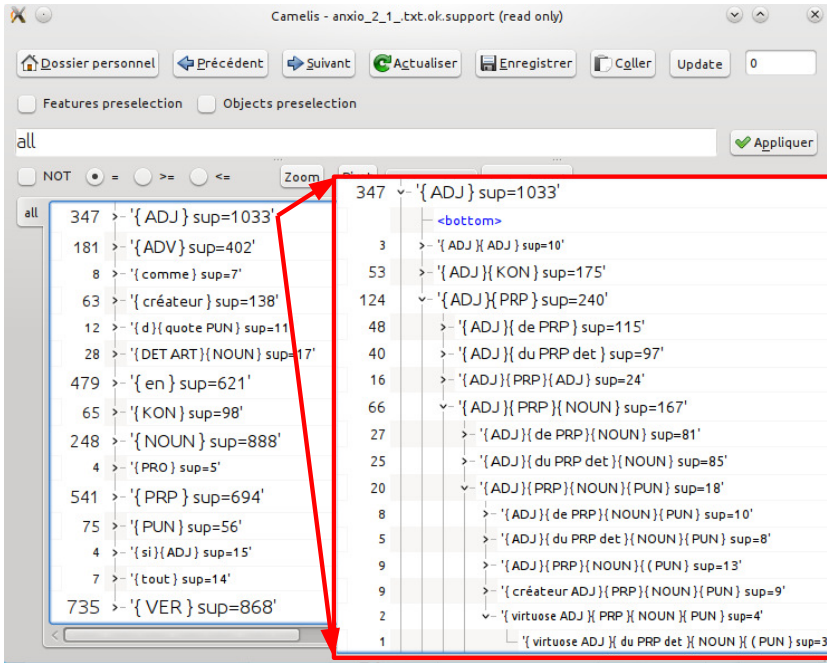


Fig. 5. Pattern discovering: Example with the ART corpus

Results on AXOLIO corpus show that our method is able to *automatically* recover all the handcrafted linguistic patterns presented in Section 3.2. Even better, our method **declines generic patterns on different specific ways**. Note that the very specific patterns are obtained by setting a low support threshold. For instance the specific patterns for the generic pattern $\langle (NOUN) (PRP) (NOUN) \rangle$ are:

- $\langle (NOUN) (du PRP det) (NOUN) \rangle$;
- $\langle (NOUN) (de PRP) (NOUN) \rangle$;
- $\langle (spécialiste NOUN) (de PRP) (NOUN) \rangle$;
- $\langle (homme NOUN) (de PRP) (NOUN) \rangle$;
- $\langle (homme NOUN) (de PRP) (conviction NOUN) \rangle$.

In addition, results produce syntagmatic constructions with various forms and expansions. The example below shows some extracted constructions of adjectival group:

$\langle (ADV) (ADJ) \rangle$; $\langle (ADV) (ADJ) (PRP) (VER\ infi) \rangle$; $\langle (ADV) (ADJ) (et KON) (ADJ) \rangle$; $\langle (ADJ) (mais KON) (ADJ) \rangle$; $\langle (ADJ) (et KON) (VER\ pper) \rangle$;
 $\langle (ADV) (ADJ) (à PRP) (VER\ infi) \rangle$; $\langle (ADV) (ADV) (ADJ) \rangle$; $\langle (ADV) (plus ADV) (ADJ) \rangle$, and so on.

Results on the ARTS corpus show the interest of the method with noisy data. Let us recall that this corpus was automatically generated and the phrases have not been tagged. Therefore, some sequential patterns extracted from the corpus may suggest non relevant patterns. Thanks to the hierarchical navigation proposed in the process by using the Camelis tool, such noisy patterns can be easily removed and the selection of relevant

linguistic patterns is easy. Figure 5 depicts an excerpt of the pattern hierarchy within this corpus. Then, we can discover **new linguistic patterns** (compared to those proposed in [12]), resulting of a manual extraction) in order to extract qualifying appositive phrases. For instance, we discover the pattern $\langle (ADJ) (pour) (DET) (NOUN) \rangle$ which matches phrases such as: “*célèbre pour son monastère*” (“*famous for its monastery*”), “*baroque pour une histoire d’amour*” (“*baroque for a love story*”). We also discover some variations or extensions: $\langle (ADV) (ADJ) (pour) \rangle$ (e.g., “*très célèbre pour*” (“*very famous for*”)), $\langle (ADJ) (pour) (VER) \rangle$ (e.g., “*indispensable pour assurer*” (“*essential to ensure*”)).

4 Conclusion

We have proposed an approach based on the extraction of sequential patterns which aims at automatically discovering linguistic patterns. Whereas existing methods are based on single-item sequences, our approach extracts sequences of itemsets. It leads to more expressiveness in the discovered patterns by combining the different levels of word abstraction (word, lemma, grammatical category). In addition, the extracted patterns are understandable by a human unlike machine learning based methods. Moreover, sequence mining approaches are not language-dependent. We have designed an algorithm for mining such sequential patterns. An outstanding idea of our algorithm is to take into account constraints in order to reduce the number of extracted patterns and therefore also to reduce the time processing. However, the number of sequential patterns can remain high. In order to address that problem, we have proposed to take advantage of the partial order between patterns and use a tool allowing a user to easily navigate within the pattern space and validate sequential patterns as relevant linguistic patterns. We have conducted some experiments to discover linguistic patterns to extract appositive qualifying phrases. Results show that even with a noisy corpus. In addition thanks to the navigation tool, an expert can easily select relevant patterns.

Further work is the evaluation of the patterns according to a task without gold standard as the task that we consider in this paper. This is a well-known issue in unsupervised methods. Another further work is to enhance the algorithm with new constraints in order to reduce the number of extracted sequential patterns. For example, a constraint of maximum support can be used to filter out patterns very general. Finally, the approach presented in this paper is not specific to the detection of appositive qualifying phrases. It can also be used to extract other kinds of linguistic patterns, acquiring new resources as lexicons or extraction rules. For instance, mining sequential patterns of itemsets in order to extract the relationships between named entities (such as interaction between genes) would improve the state-of-the-art works.

Acknowledgements. This work is partly supported by the ANR (French National Research Agency) funded project Hybride ANR-11-BS02-002.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: ICDE. IEEE (1995)
2. Bonchi, F.: On closed constrained frequent pattern mining. In: Proc. IEEE Int. Conf. on Data Mining, ICDM 2004, pp. 35–42. Press (2004)

3. Califf, M.E., Mooney, R.J.: Relational learning of pattern-match rules for information extraction. In: AAAI 1999, pp. 328–334 (1999)
4. Cellier, P., Charnois, T., Plantevit, M.: Sequential Patterns to Discover and Characterise Biological Relations. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 537–548. Springer, Heidelberg (2010)
5. Davey, B.A., Priestley, H.A.: Introduction To Lattices And Order. Cambridge University Press (1990)
6. Dong, G., Pei, J.: Sequence Data Mining. Springer, Heidelberg (2007)
7. Ferr, S.: Camelis: a logical information system to organize and browse a collection of documents. *Int. J. General Systems* 38(4) (2009)
8. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. In: KDD, pp. 1–30. AAAI/MIT Press (1991)
9. Fundel, K., Küffner, R., Zimmer, R.: RelEx - relation extraction using dependency parse trees. *Bioinformatics* 23(3), 365–371 (2007)
10. Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: EACL (2006)
11. Hobbs, J.R., Riloff, E.: Information extraction. In: Indurkha, N., Damerau, F.J. (eds.) *Handbook of Natural Language Processing*, 2nd edn. CRC Press, Taylor and Francis Group, Boca Raton, FL (2010)
12. Jackiewicz, A.: Structures avec constituants détachés et jugements d'évaluation. *Document Numérique* 13(3), 11–40 (2010)
13. Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A.: Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology* 9 (2008)
14. Nédellec, C.: Machine learning for information extraction in genomics - state of the art and perspectives. In: *Text Mining and its Applications: Results of the NEMIS Launch Conf., Studies in Fuzziness and Soft Comp., Sirmakessis, Spiros* (2004)
15. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Prefixspan: Mining sequential patterns by prefix-projected growth. In: ICDE, pp. 215–224. IEEE Computer Society (2001)
16. Riloff, E.: Automatically generating extraction patterns from untagged text. In: AAAI/IAAI 1996 (1996)
17. Sagot, B., Clément, L., de La Clergerie, E., Boullier, P.: The lefff 2 syntactic lexicon for french: architecture, acquisition, use. In: LREC 2006, Genoa, Italy (2009)
18. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: *Proceedings of International Conference on New Methods in Language Processing* (September 1994)
19. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
20. Wang, J., Han, J.: Bide: Efficient mining of frequent closed sequences. In: ICDE, pp. 79–90. IEEE Computer Society (2004)
21. Yan, X., Han, J., Afshar, R.: Clospan: Mining closed sequential patterns in large databases. In: Barbará, D., Kamath, C. (eds.) SDM. SIAM (2003)
22. Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal* 42(1/2), 31–60 (2001) (special issue on Unsupervised Learning)

What about Sequential Data Mining Techniques to Identify Linguistic Patterns for Stylistics?

Solen Quiniou^{1,2}, Peggy Cellier³, Thierry Charnois¹, and Dominique Legallois²

¹ GREYC Université de Caen Basse-Normandie, Campus 2, 14000 Caen

² CRISCO Université de Caen Basse-Normandie, Campus 1, 14000 Caen

³ IRISA-INSA de Rennes, Campus de Beaulieu, 35042 Rennes Cedex

Abstract. In this paper, we study the use of data mining techniques for stylistic analysis, from a linguistic point of view, by considering emerging sequential patterns. First, we show that mining sequential patterns of words with gap constraints gives new relevant linguistic patterns with respect to patterns built on n -grams. Then, we investigate how sequential patterns of itemsets can provide more generic linguistic patterns. We validate our approach from a linguistic point of view by conducting experiments on three corpora of various types of French texts (*Poetry*, *Letters*, and *Fiction*). By considering more particularly poetic texts, we show that characteristic linguistic patterns can be identified using data mining techniques. We also discuss how to improve our proposed approach so that it can be used more efficiently for linguistic analyses.

1 Introduction

The study of phraseology - including stylistics - is a research field that has been investigated over the past 30 years by the linguistic community. More recently, there has been a particular interest in studies from corpus linguistics. Two main approaches can be identified: corpus-based and corpus-driven. *Corpus-based* approaches assume the existence of linguistic theories and use corpora to analyze their application and hence to validate them. *Corpus-driven* approaches consider that linguistic constructs emerge from corpus analysis. This analysis allows the discovery of co-occurring word patterns that will be the basis of linguistic analyses. Our work is part of the corpus-driven approaches since our goal is to assist linguists in discovering new linguistic constructs without any prior knowledge.

One of the first corpus-driven approach was proposed by Renouf and Sinclair [1]. It consists on a study of collocational frameworks thanks to corpora; *collocational frameworks* represent discontinuous sequences of two grammatical words enclosing a lexical word (e.g., “*many + ? + of*” that means *many* followed by a variable lexeme - symbolized by *?* - itself followed by *of*). However, this approach is not entirely corpus-driven since the studied collocational frameworks were pre-selected by Renouf and Sinclair. In fact, most of the so-called corpus-driven approaches are partly corpus-based [2]. More recently, Biber presented an

¹ See [2] for a more detailed state of the art on corpus-driven approaches.

interesting approach, entirely corpus-driven, to identify frequent patterns from corpora [2]. To do so, he relies both on a preliminary work on the identification of *lexical bundles* (i.e., frequent sequences of contiguous words, aka *n*-grams) and on collocational frameworks to identify fixed and variable elements in the patterns he extracted. Furthermore, Biber considers two language registers (conversation and academic writing) and shows the interest of using a corpus-driven approach to study the specificities of patterns appearing in each register.

In this paper, we present a first and original study which aims at showing the interest of data mining methods for the stylistic analysis of large texts. The goal is to provide to the linguist experts some prominent, relevant, and understandable patterns which can be characteristic of a specific type of text so that these experts can carry out a stylistic analysis based on these patterns. In fact, our work is in the continuity of Biber's but we consider various text types (instead of language registers) that we study from a stylistic point of view. To do so, we set up a methodology based on sequential data mining, from the extraction of patterns to the selection of the most relevant. We apply this methodology to stylistics. To the best of our knowledge, data mining methods have not yet been used in the field of stylistics whereas one of their advantages is to offer an interpretable result to users, as opposed to numerical methods such as Hidden Markov Models or Conditional Random Fields. Indeed, the latter methods have been shown to achieve good results for tasks like text categorization or information extraction but they produce outputs hardly understandable by humans. Thus, the approach that we propose is based on *frequent sequential patterns* [3], a well-known data mining technique to automatically discover frequent patterns based on the sequential order of data. We consider two types of sequential patterns: single-item patterns (an *item* represents a single piece of information, e.g. a word form); and *itemset* patterns. In this second type of patterns, a word is represented by a set of features. Therefore, extracted itemset patterns may combine different levels of abstraction (word forms, lemmas, POS tags, etc.): for instance, $\langle\langle\text{PREP}\rangle\rangle\langle\langle\text{DET}\rangle\rangle\langle\langle\text{NC}\rangle\rangle$ or $\langle\langle\text{to}\rangle\rangle\langle\langle\text{the DET}\rangle\rangle\langle\langle\text{NC}\rangle\rangle$ [2]. Furthermore, as we set our study in the field of stylistics, the end-goal is to extract patterns that are characteristic of a certain type of text. This is the reason why we focus on a specific type of sequential patterns: *emerging patterns*. Emerging patterns can capture contrast characteristics between classes or datasets [4]. Furthermore, these patterns can be analyzed by experts to discover new relationships in a given domain for a better understanding of it. Here, extracted emerging patterns could then be analyzed by linguists to discover linguistic patterns, characteristic of a certain type of text.

The rest of this paper is organized as follows. First, our methodology based on sequential data mining is introduced in Section 2. Then, Section 3 presents experimental results on the use of our methodology for stylistics both from a quantitative and a linguistic point of view. Finally, Section 4 discusses the leads to further investigate, while Section 5 draws some conclusions.

² PREP, DET, and NC are the POS tags for prepositions, determiners, and common nouns respectively.

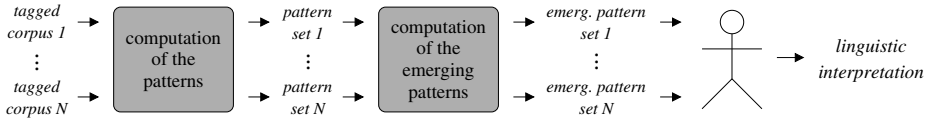


Fig. 1. Overview of our proposed approach

2 Methodology

In this section, we give an overview of the proposed approach to identify characteristic linguistic patterns for each type of text (Section 2.1). Then, we present the sequential data mining techniques on which our approach is based: frequent sequential patterns (Section 2.2) and emerging patterns (Section 2.3).

2.1 Overview of the Proposed Approach

Figure 1 illustrates the various steps of our approach. N corpora are used as the inputs of the process, with one corpus corresponding to a considered type of text. Each corpus is first pre-processed and then all its words are labeled with their lemma and their POS category (see Section 3.1). In the first step of our approach, sequential patterns are extracted for each corpus: N sets of sequential patterns are therefore obtained. Then, in the second step, sets of emerging patterns are selected for each corpus, from the N sets of sequential patterns previously extracted. Lastly, the N sets of emerging patterns are given to a linguist so that he can use them to perform a linguistic interpretation. The first and second steps are presented in greater details in the next sub-sections.

2.2 Sequential Pattern Mining

Sequential pattern mining is a well-known data mining technique used to find regularities in sequence databases, by considering the temporal order of the data. This technique was introduced by Agrawal *et al.* in [3].

An *itemset*, I , is defined as a set of literals called *items*, denoted by $I = (i_1 \dots i_n)$. For example, $(a\ b)$ is an itemset with two items: a and b . A *sequence*, S , is defined as an ordered list of itemsets, denoted by $S = \langle I_1 \dots I_m \rangle$. For instance, $\langle (a\ b)(c)(d)(a) \rangle$ is a sequence of four itemsets. It should be noted that a lot of applications need only one item in their itemsets (*e.g.* DNA strings or protein sequences). These particular kinds of sequences are called *single-item sequences*; for the sake of clarity, they are denoted by $S = \langle i_1 \dots i_n \rangle$, where $i_1 \dots i_n$ are items. Several algorithms have been developed to efficiently mine that kind of specific sequences, for example [5]. In the rest of the paper, both kinds of sequences will be considered, *i.e.* single-item sequences and itemset sequences. A sequence $S_1 = \langle I_1 \dots I_n \rangle$ is *included* in a sequence $S_2 = \langle I'_1 \dots I'_m \rangle$ if there exist integers $1 \leq j_1 < \dots < j_n \leq m$ such that $I_1 \subseteq I'_{j_1}, \dots, I_n \subseteq I'_{j_n}$. The sequence S_1 is thus called a *subsequence* of S_2 , which is noted $S_1 \preceq S_2$. For example, we have the following relation: $\langle (c)(a) \rangle \preceq \langle (a\ b)(c)(d)(a) \rangle$. A sequence

Table 1. SDB_1 : a sequence database

Sequence identifier	Sequence
1	$\langle(a\ b)(c)(d)(a)\rangle$
2	$\langle(d)(a)(e)\rangle$
3	$\langle(d)(a\ b\ e)(c\ d\ e)\rangle$
4	$\langle(c)(a)\rangle$

database SDB is a set of tuples (sid, S) , where sid is a sequence identifier and S a sequence. For instance, Table 1 represents a sequence database of four sequences. A tuple (sid, S) contains a sequence S_1 , if $S_1 \preceq S$. The support of a sequence S_1 in a sequence database SDB , denoted $sup(S_1)$, is the number of tuples containing S_1 in the database. For example, in Table 1, $sup(\langle(a)(e)\rangle) = 2$ since sequences 2 and 3 contain an itemset with a followed by an itemset with e . The relative support of sequences may also be used, as defined by Equation 1:

$$sup(S_1) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \wedge (S_1 \preceq S)\}|}{|SDB|} \tag{1}$$

A frequent pattern is a sequence such that its support is greater or equal to a given threshold: $minsup$. Sequential pattern mining algorithms thus extract all the frequent sequential patterns that appear in a sequence database.

Because the set of frequent sequential patterns can be very large, there exists a condensed representation which eliminates redundancies without loss of information: closed sequential patterns [6]. A frequent sequential pattern S is closed if there exists no other frequent sequential pattern S' such that $S \preceq S'$ and $sup(S) = sup(S')$. For instance, with $minsup = 2$, the sequential pattern $\langle(b)(c)\rangle$ from Table 1 is not closed whereas the pattern $\langle(a\ b)(c)\rangle$ is closed. Moreover, in order to drive the mining process towards the user objectives and to eliminate irrelevant patterns, one can define constraints [7,8]. The most commonly used constraint is the frequency constraint (that assigns a value to $minsup$). Another widespread constraint is the gap constraint. A sequential pattern with a gap constraint $[M, N]$, denoted by $P_{[M, N]}$, is a pattern such as at least $M - 1$ itemsets and at most $N - 1$ itemsets are allowed between every two neighbor itemsets, in the original sequences. For instance, let $P_{[1,3]} = \langle(c)(a)\rangle$ and $P_{[2,3]} = \langle(c)(a)\rangle$ be two patterns with two different gap constraints and let us consider the sequences of Table 1. Sequences 1 and 4 are occurrences of pattern $P_{[1,3]}$ (sequence 1 contains one itemset between (c) and (a) whereas sequence 4 contains no itemset between (c) and (a)), but only sequence 1 is an occurrence of $P_{[2,3]}$ (only sequences with one or two itemsets between (c) and (a) are occurrences of this pattern).

In this paper, the considered databases correspond to corpora. Furthermore, two kinds of sequential patterns are considered: single-item patterns and itemset patterns. In that last case, itemsets can be made up of three types of items: word forms, lemmas, and POS tags.

2.3 Emerging Patterns

Emerging patterns are defined as sequential patterns whose support increases significantly from one dataset to another one [4]. More specifically, emergent patterns are sequential patterns whose *growth rate* - the ratio of the supports in the two datasets - is larger than a given threshold: ρ . Thus, a sequential pattern P from a dataset D_1 is an *emerging pattern* to another dataset D_2 if $GrowthRate(P) \geq \rho$, with $\rho > 1$ and with $GrowthRate(P)$ being defined by:

$$GrowthRate(P) = \begin{cases} \infty, & \text{if } sup_{D_2}(P) = 0 \\ \frac{sup_{D_1}(P)}{sup_{D_2}(P)}, & \text{otherwise} \end{cases} \quad (2)$$

with $sup_{D_1}(P)$ ($sup_{D_2}(P)$ respectively) being the relative support of the pattern P in D_1 (D_2 respectively). Since we are only interested in patterns belonging to D_1 , we do not consider patterns P with $sup_{D_1}(P) = 0$.

In the case of stylistic analyses, each dataset contains the frequent sequential patterns of a corpus and thus of the corresponding type of text. It corresponds to the patterns extracted during the first step of our approach (see Section 2.2). Because we consider more than two types of text, we compute the emerging patterns of a considered type of text with respect to every other type, according to Equation 2. Finally, only the patterns that are emerging to every other type of text are kept as emerging patterns for a considered type of text. The computation of all the emerging patterns is done efficiently based on [9].

3 Experimental Evaluation

In this section, we report the results of our experimental evaluation on using sequential pattern mining techniques for stylistics. First, in Section 3.1, we describe the used corpora as well as the setup of the various parameters used to extract emerging sequential patterns. Then, we present an analysis of the extracted sequential patterns, at two levels: from a quantitative point of view (in Section 3.2), and from a linguistic point of view for stylistics (in Section 3.3).

3.1 Experimental Setup

Corpora. We created three corpora, corresponding to various types of text: *Poetry*, *Letters*, and *Fiction*. To build each corpus, we selected all the texts of the 1800-1900 era - provided by the French resources of the CNRTL³ - corresponding to the considered type of text. For example, authors from *Poetry* include Lamartine and Musset, whereas Hugo and Lamennais are part of the authors of *Letters*, and Chateaubriand and Zola are authors of *Fiction*. Then, these three corpora were pre-processed. The pre-processing steps consisted in setting the words in lower-case, and then splitting the texts into sequences at punctuation marks of the set: $\{', ', '?', '!', '...', ';;', ';;', ';;', ';;'\}$. Table 2 gives some details on each corpus: the number of authors, of works, of sequences, and of words.

³ Centre National des Ressources Textuelles et Linguistiques : www.cnrtl.fr

Table 2. Characteristics of the corpora *Poetry*, *Letters*, and *Fiction*

Corpus	#authors	#works	#sequences	#words
<i>Poetry</i>	27	48	151 116	1 167 422
<i>Letters</i>	5	9	234 997	1 562 543
<i>Fiction</i>	37	52	663 860	5 105 240

After being pre-processed, the corpora were POS tagged using *Cordial*⁴, a tagger that is known to outperform TreeTagger on French texts. Thus, each word of the corpora was associated with its form, its lemma and its POS tag. After first experimentations, it turns out that the POS tags given by Cordial were too much specific; we thus post-processed them to reduce their number (as a consequence, it reduces the number of extracted patterns). First, too specific categories were merged into more general ones. For example, the adjective category was initially decomposed into 16 categories (depending on the gender, the number, or whether the word starts with a mute *h* letter). Thus, the following categories were created, to replace their corresponding sub-categories: adjectives (ADJ), determiners (DET), common nouns (NC), proper nouns (NP), demonstrative pronouns (PD), relative pronouns (PR), indefinite pronouns (PI), and past participles (VPARP). Then, categories corresponding to personal pronouns were decomposed into 2 tags: one for the personal pronoun (PPER), and one for the person (*e.g.* 1S for the singular first person). Moreover, categories corresponding to verbs were decomposed into 3 tags: one for the verb (V), one for the mode of the verb (*e.g.* INDP for the present of the indicative mode), and one for the person (the same ones as for the personal pronouns). At the end, we had a set of 35 tags instead of the 133 initial tags. Using this new set of tags, the phrase “*a rose that we smell*” is translated as $\langle (a a DET) (rose rose NC) (that that PR) (we we PPER 1P) (smell smell V PRES 1P) \rangle$.

Mining Single-Item Sequences. First, we considered single-item sequences of words. To perform the mining task on the three corpora, we used *dmt4*⁵ that allows the definition of various constraints on the extracted single-item sequential patterns: the length, the frequency (by setting *minsup*, the support threshold), or the gaps (by choosing the values of $[M, N]$). We set the length of the patterns to be between 2 and 20. We chose the value of *minsup* empirically as a trade-off between having interesting patterns with a low support (thus setting a low value to *minsup*) and having not too many patterns (thus setting a high value to *minsup*). Because of the differences in the corpora sizes (*Fiction* is five times bigger than *Poetry*), we chose a relative threshold whose value is 0.001 %; it corresponds to the following absolute thresholds: 16 for *Poetry*, 12 for *Letters*, and 51 for *Fiction*. That means that only patterns appearing in at least 16 sequences are kept for *Poetry*, for example. For the gap constraints, we chose to consider different values in the following experiments (see Section 3.2): $[1, 1]$, $[1, 2]$, $[1, 3]$, and $[1, 5]$. It is worth noting that the $[1, 1]$ gap constraint corresponds

⁴ The Cordial tagger is developed by Synapse Développement (www.synapse-fr.com)

Table 3. Number of patterns and ratio of emerging ones (in brackets) for the corpora

Corpus	Single-item patterns with gaps				Itemset patterns
	[1, 1]	[1, 2]	[1, 3]	[1, 5]	
<i>Poetry</i>	18 816 (30.7 %)	37 933 (27.0 %)	55 762 (24.3 %)	86 901 (22.6 %)	2 245 326 (11.4 %)
<i>Letters</i>	16 936 (50.2 %)	36 849 (50.7 %)	56 755 (50.4 %)	96 549 (50.0 %)	10 128 288 (57.4 %)
<i>Fiction</i>	78 210 (6.1 %)	175 645 (5.3 %)	282 967 (4.9 %)	512 647 (4.6 %)	11 681 913 (71.2 %)
Total	113 962 (16.7 %)	250 427 (15.3 %)	395 484 (14.2 %)	696 097 (13.2 %)	24 055 527 (59.8 %)

to considering n -gram patterns. Indeed, patterns extracted under this constraint correspond to sub-sequences of consecutive words of the corpus.

Mining Itemset Sequences. Finally, we considered itemset sequences, where each itemset represents a word with its form, its lemma, and its POS tag. To mine these itemset sequences, we chose CloSpan [6] that extracts closed sequential itemset patterns. CloSpan allows to set only one constraint: the support threshold *minsup*. We also chose empirically the value of *minsup* to be 0.15%. Note that, because no gap constraint can be set in CloSpan, we had to choose a higher value for *minsup* to limit the total number of patterns that are generated and hence to limit the computation time. The drawback of that choice is that interesting patterns may not be extracted because their support may be too low (for example, the absolute support threshold is 1 000 for *Fiction*).

Selecting Emerging Patterns. To select the emerging patterns of the corpora, we set the threshold ρ just above 1: $\rho = 1.001$. This threshold is used on both single-item patterns and itemset patterns.

3.2 Quantitative Analysis of the Patterns

In this sub-section, we present quantitative results on the single-item patterns and on the itemset patterns. The set of extracted patterns being large, this quantitative analysis allows us to select the patterns that will be actually analyzed from a linguistic point of view, for the stylistic task (see Section 3.3).

Table 3 gives the number of extracted patterns for the three corpora, by considering the two types of patterns: single-item patterns (with various gap constraints) and itemset patterns. The ratio of emerging patterns is also given for each type of patterns. Thus, among the 18 816 patterns extracted from *Poetry* (by setting the gap constraint to [1, 1]), 30.7 % of the patterns are emerging ones (corresponding to 5 776 patterns). First, we can see that selecting emerging patterns allows a large reduction of the total number of sequential patterns to analyze. Moreover, it allows to focus our attention on more interesting patterns in the context of stylistics. That is why we will only consider emerging patterns

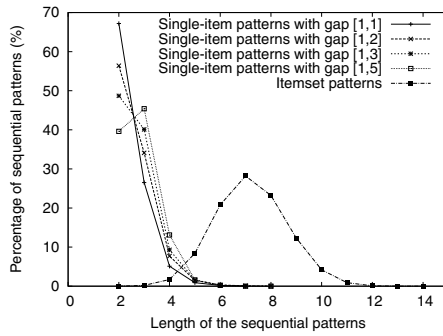


Fig. 2. Distribution of the emerging patterns w.r.t. the length

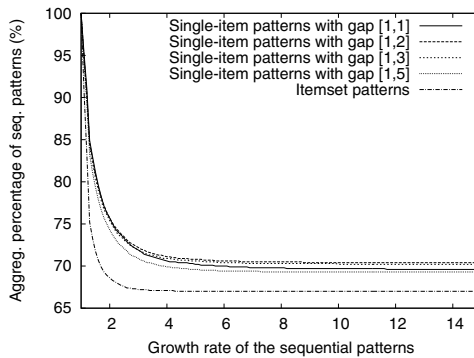


Fig. 3. Distribution of the emerging patterns w.r.t. the growth rate

in the rest of the analyses. Furthermore, we can see that by increasing the gap constraint, the rate of single-item emerging patterns tends to decrease: it means that additional extracted patterns tend to be non-specific patterns of the studied types of text. For the stylistic analysis presented in Section 3.3, we set the gap constraint to [1, 3] as a tradeoff between the total number of extracted single-item patterns and their relevance. Finally, we can see that many more itemset patterns are extracted, compared to the number of extracted single-item patterns.

Then, we study the distribution of the emerging patterns w.r.t. their length. Figure 2 plots the relative number of patterns for the various pattern lengths, for the single-item patterns (the length is given as the number of items) and for the itemset patterns (the length is given as the number of itemsets). The pattern distributions are computed on the three corpora considered as a whole, for each gap constraint value considered. We can see that most of the single-item patterns contain between 2 and 5 items whereas most of the itemset patterns contain between 4 and 11 itemsets. Therefore, itemset patterns represent longer linguistic patterns. Moreover, there are a lot of single-item patterns of length 2 but they are not as instructive as longer patterns - from a linguistic point of

view. That is why we will only consider patterns whose length is greater than 2, for the stylistic analysis.

Finally, we study the distribution of the emerging patterns w.r.t. growth rates. Figure 3 plots the aggregate relative number of emerging patterns as a function of the growth rate, by considering the three corpora as a whole. It means that, for example, 67.1 % of the emerging itemset patterns have a growth rate greater than 4. We can see that most of the emerging patterns have an infinite growth rate as the aggregate rate of emerging patterns is stable for growth rates greater than 10. It means that most of the emerging patterns appear only in a certain type of text (and not at all in the other types of text). In the stylistic analysis, we consider only emerging patterns with an infinite growth rate.

Finally, only itemsets containing both POS tags and word forms or lemmas are considered during the stylistic analysis. Patterns containing only POS tags are therefore removed as they are too general and patterns containing only word forms or lemmas are also removed as they are too specific. In fact, most of the itemset patterns contain both POS tags and words since these patterns represent 93.5 % of all the itemset patterns. That concurs with Biber’s conclusions [2] on the extracted patterns that contain both variable and fixed elements (patterns only with POS tags thus contain only variable elements whereas patterns only with words contain only fixed elements).

3.3 Stylistic Analysis of the Emerging Patterns

In this sub-section, we present a stylistic analysis of some extracted emerging patterns. We focus our attention more particularly on the *Poetry* corpus.

First of all, we consider single-item patterns. By studying them, we can find some interesting patterns, characteristic of *Poetry*. Table 4 shows examples of such identified characteristic patterns. In the patterns, the symbol * is used to represent a gap of one or more words⁵. Furthermore, we also illustrate each pattern with examples of underlying sequences in *Poetry*. The extracted patterns allow the observation of schematic grammatical structures that are relatively lexicon-independent. Indeed, fixed elements of these patterns are grammatical words whereas variable elements (*i.e.*, filling the gaps) are generally lexical words (*e.g.*, nouns, verbs, or adjectives). We also show the interest of gap constraints that are given as intervals. The pattern “*some*more*than*” allows the identification of two sequences, among others, where the first gap is filled with a different number of words (see Table 4): in the first one, the word *bites* fills the gap whereas it is filled with the words *angular rocks* in the second sequence. This illustrates the generalization capability of single-item patterns with gap constraints (w.r.t. *n*-gram patterns, for instance).

Table 5 gives the correspondence between the single-item patterns presented in Table 4 and their associated itemset patterns. First, it can be seen that several

⁵ Symbol * corresponds to symbol ? used in [1]. Note that symbol * is also used in [2] but it represents a single variable lexeme whereas, in our approach, this symbol represents a gap of one or more words.

Table 4. Examples of characteristic single-item patterns from *Poetry*

Single-item pattern	Example (<i>with English translation</i>)
des*plus*que (<i>some*more*than</i>)	il a des morsures plus venimeuses que celles de ta bouche (<i>he has some bites more venomous than those from your mouth</i>) des cailloux anguleux plus brillants que des marbres (<i>some angular rocks brighter than some marbles</i>)
on*et*on (<i>we*and*we</i>)	une rose qu' on respire et qu' on jette (<i>a rose that we smell and that we throw</i>) sur des tombeaux divins qu' on brise et qu' on insulte ? (<i>on divine tombs that we break and that we insult?</i>)
le/la/l'*qui*et*qui (<i>the*that*and*that</i>)	la nuit qui m'opresse et qui trouble mes yeux (<i>the night that oppresses me and that troubles my eyes</i>) le grelot qui résonne et le troupeau qui bêle (<i>the bell that resounds and the flock that bleats</i>)
le*du*qui*dans (<i>the*of the*that*in</i>)	le vent du soir qui meurt dans le feuillage (<i>the wind of the night that dies in the foliage</i>) le bruit du vieux qui bêche dans la nuit (<i>the sound of the old that digs in the night</i>)
est*un*qui (<i>is*a*that</i>)	est -ce un goéland qui bat de l'aile ? (<i>is it a gull that flaps its wing?</i>) ta grâce est comme un luth qui vibre au fond du bois (<i>your grace is like a lute that vibrates deep in the wood</i>)

itemset patterns may correspond to the same single-item pattern. Furthermore, extracted itemset patterns allow to obtain the POS categories of the variable elements. Therefore, in the context of a stylistic study of types of text, the work of linguists consists in selecting relevant patterns among automatically extracted itemset patterns: this directly gives them grammatical patterns characteristic of a considered type of text.

In fact, the grammatical patterns we consider correspond to *collocational frameworks* in the sense of Renouf and Sinclair [1], *i.e.* collocations on grammatical units and not on lexical units. However, as opposed to their work, we do not chose *a priori* the patterns that are then studied but we automatically discover them from corpora. We can also compare our work to Biber's [2] - who works also on collocational frameworks - but there are some differences. Indeed, our approach allows to directly extract single-item patterns with gaps as well as itemset patterns (corresponding to grammatical patterns) whereas Biber first extracts frequent sequences from corpora and then analyze them one by one to identify variable and fixed elements to finally build various types of patterns that he studies afterwards. Since Renouf and Sinclair paper, works on collocational frameworks have been done in English corpus linguistics, but not in French. Nonetheless, the analysis of collocational frameworks can be full of insights when associated to an actual usage theory considering that grammatical forms come from a linguistic usage (*i.e.* corpus-driven approaches) and are not the result of integrated rules (*i.e.* corpus-based approaches). Therefore, it is interesting to have approaches that automatically extract patterns to provide these collocation frameworks, as it is the case with our proposed approach.

Table 5. Grammatical patterns corresponding to some identified single-item patterns

Single-item pattern	(English translation)	Grammatical pattern
des*plus*que	(some*more*than)	some N more ADJ than
on*et*on	(we*and*we)	N that we V and that we V
le/la/l'*qui*et*qui	(the*that*and)	the N that V and (that) V the N that V and the N that V
le*du*qui*dans	(the*of the*that*in)	the N of the N that V in the N
est*un*qui	(is*a*that)	is it a N that V is like a N that V

4 Discussion

In the previous section, we have shown that sequential patterns can be interpreted by linguists for stylistic analyses. However, a huge number of sequential patterns are extracted with data mining techniques, from which the interesting ones have to be identified. In this section, we discuss the improvements that could be brought to our current approach to make it easier for linguists to deal with the presented sequential patterns. To this end, we identified two leads.

First, in order to focus our attention on the interesting sequential patterns, it is necessary to be able to set new constraints during the data mining process to narrow the number of extracted patterns down. Thus, it would be interesting to also set gap constraints on itemset patterns (as it is already the case for single-item patterns). In addition, as we can set a minimum threshold, *minsup*, for the pattern supports, it would be interesting to set a maximum threshold, *maxsup*, for the pattern supports as well. Indeed, most interesting sequential patterns generally appear in few sequences. Thus, by not considering too frequent sequential patterns, the total number of patterns would be reduced (for instance, by setting *maxsup* = 50, 21.2 % of the *Poetry* single-item patterns would not be extracted). Moreover, it would allow us to set *minsup* to a lower value, and hence to discover rarer sequential patterns without increasing the total number of patterns. In addition, membership constraints on a certain item type could also be defined to filter out more sequential patterns (*e.g.* only considering sequential patterns containing at least one verb).

Lastly, it would be of interest to provide tools allowing the ordering of the patterns, their filtering, or their exploration jointly with the sequences of the corpus they refer to. Therefore, it would be easier for linguists, in particular, to analyze the extracted sequential patterns (more particularly for itemset patterns or for sequential patterns with gap constraints).

5 Conclusion

In this paper, we have presented a first study on using data mining techniques for stylistics by proposing a methodology based on the extraction of sequential

patterns and applied to stylistics. Thus, we have considered two types of sequential patterns: single-item patterns and itemset patterns (based on word forms, lemmas and POS tags). Moreover, we focused our attention on specific sequential patterns: emerging patterns. A quantitative analysis of the sequential patterns extracted from three corpora (representing various types of text, aka *Poetry*, *Letters*, and *Fiction*) has shown that sequential patterns are more powerful than n -grams to express linguistic patterns. That has been confirmed by a linguistic analysis of the extracted emerging sequential patterns since some grammatical patterns characteristic of *Poetry* were identified from these sequential patterns. We also compared our methodology to the one proposed by Biber [2] by showing that ours allows to directly obtain patterns characteristic of types of text. Lastly, we have discussed the improvements that could be brought to our proposed approach both by limiting the total number of extracted sequential patterns and hence to analyze (by defining new constraints on the patterns), and by making it easier for linguists to explore and analyze the patterns (by developing suitable tools for this task). Therefore, these discussions give us leads to investigate in future studies; some of these works are already in progress.

Acknowledgments. This work is partly supported by the French Région Basse-Normandie and by the ANR (National Research Agency) funded project Hybride ANR-11-BS02-002.

References

1. Renouf, A., Sinclair, J.: Collocational Frameworks in English. In: English Corpus Linguistics: Studies in Honour of Jan Svartvik, pp. 128–143. Longman (1991)
2. Biber, D.: A corpus-driven approach to formulaic language in English. *International Journal of Corpus Linguistics* 14, 275–311 (2009)
3. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of ICDE 1995, pp. 3–14 (1995)
4. Dong, G., Li, J.: Efficient minig of emerging patterns: Discovering trends and differences. In: Proc. of SIGKDD 1999, pp. 43–52 (1999)
5. Nanni, M., Rigotti, C.: Extracting trees of quantitative serial episodes. In: Proc. of KDID 2007, pp. 170–188 (2007)
6. Yan, X., Han, J., Afshar, R.: Clospan: Mining closed sequential patterns in large databases. In: Proc. of SDM 2003 (2003)
7. Dong, G., Pei, J.: *Sequence Data Mining*. Springer, Heidelberg (2007)
8. Ng, R., Lakshmanan, L., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proc. of SIGMOD 1998, pp. 13–24 (1998)
9. Plantevit, M., Crémilleux, B.: Condensed Representation of Sequential Patterns According to Frequency-Based Measures. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 155–166. Springer, Heidelberg (2009)

Resolving Syntactic Ambiguities in Natural Language Specification of Constraints

Imran Sarwar Bajwa, Mark Lee, and Behzad Bordbar

School of Computer Science, University of Birmingham,
B15 2TT Birmingham, UK
{i.s.bajwa,m.g.lee,b.bordbar}@cs.bham.ac.uk

Abstract. In the NL2OCL project, we aim to translate English specification of software constraints to formal constraints such as OCL (Object Constraint Language). In the used approach, the Stanford POS tagger and the Stanford Parser are employed for syntactic analysis of English specification and the output of syntactic analysis is given to our semantic analyzer for the detailed semantic analysis. However, in few cases, the Stanford POS tagger and parser are not able to handle particular syntactic ambiguities in English specifications of software constraints. In this paper, we highlight the identified cases of syntactic ambiguities and we also present a novel technique to automatically resolve the identified syntactic ambiguities. By addressing the identified cases of syntactic ambiguities, we can generate more accurate and complete formal (OCL) specifications.

Keywords: Syntactic Ambiguities, Attachment Ambiguity, Ambiguity resolution, English constraints.

1 Introduction

In the recent years, a few research contributions have been presented in the area of automatic translation of natural language (NL) specifications to formal specifications such as UML (Unified Modeling Language) models [1] and SQL (Structured Query Language) queries [2]. However, the available tools are limited to 65%-70% levels of accuracy in real time software development. Researchers have shown that the key reason of less accuracy is the inherent ambiguity of the natural languages. For example, Mich [3] showed that 71.8% of a sample of NL software specification is ambiguous. Hence, the ambiguous and incomplete specifications lead to inconsistent and absurd formal specifications such the software models or the software constraints.

In the NL2OCL project [4], we aim to translate the English specification of constraints to the formal constraints such as OCL (Object Constraint Language) [5]. Our contribution is a semantic analyzer that performs semantic role labeling and generates a logical representation in English to OCL translation. Our semantic analyzer relies on the output of syntactic analysis (such as typed dependency [6]) performed by the Stanford parser [7]. However, we have identified a few cases where the Stanford parser is not able to handle particular syntactic ambiguities such as

attachment ambiguity and homonymy. In result of a wrong syntax analysis, the semantic analysis goes wrong and finally the wrong OCL is generated. In this paper, we discuss the identified cases of syntactic ambiguities not resolved by the Stanford parser and we also present a novel technique to automatically resolve the identified cases of syntactic ambiguities in English specification. By addressing the identified cases of syntactic ambiguities, we can generate a more accurate and complete OCL specification.

The remaining paper is structured as follows: Section 2 provides the detailed description of the problem and the solution of the problem is given in Section 3. Section 4 presents the evaluation of the presented approach. Section 5 states the related work and the paper is concluded to discuss future work finally.

2 Description of the Problem

The NL2OCL project deals with the automated translation of English specification of constraints to OCL constraints via SBVR [12]. The most important phase in English to OCL translation is the processing of English text and generation of a logical representation such as FOL (First-Order-Logic). Finally, the logical representation is mapped to OCL syntax. In English text processing, the English text is syntactically and semantically analyzed. For syntactic analysis, the Stanford POS (Part-of-Speech) tagger [8] and the Stanford parser are used. The Stanford POS tagger is used to tag the English text and is capable of 97.0% [9] accuracy. Similarly, the Stanford parser is employed for the generation of the parse tree and the Typed Dependencies and is 84.1% [7] accurate. Accuracy of the Stanford parser is low for real-time applications and we need to improve the accuracy for robust and precise machine translation of English text.

In the semantic analysis phase, the output of the syntactic analysis is used for shallow and deep semantic parsing. In shallow semantic parsing, the semantic role labeling heavily relies on the typed dependencies generated by the Stanford parser. In result, the wrong typed dependencies lead to wrong semantic role labeling. We have identified many cases where the Stanford parser generates the correct parse tree but the typed dependencies go wrong. There are some other cases where the Stanford POS tagger wrongly tags the tokens and the error propagates in rest of the stages of syntax analysis such as parse tree generation and typed dependency generation. For correct translation of English to OCL, we need to resolve these cases.

We have identified various cases where the typed dependencies are wrongly identified because of the attachment ambiguity [10]. Similarly, the most of the errors done by the Stanford POS tagger are due to homonymy (ibid). Following are the details of both types of syntactic ambiguity:

2.1 Attachment Ambiguity

Attachment ambiguity is a type of syntactic ambiguity where a prepositional phrase or a relative clause in sentence can be lawfully attached to one of the two parts of that

sentence [10]. We have also identified some cases where the Stanford parser generates wrong dependencies due to attachment ambiguity. An example of such cases is shown in Fig. 1. In this example, it is shown that the typed dependencies generated by the Stanford parser are wrong such as `prep_with(employees-7, bonus-9)`. However, the correct typed dependency for this example should be `prep_with(pay-2, bonus-9)` to represent the actual meanings of the example i.e. the pay with bonus is given to all the employees.

English: The pay is given to all employees with bonus.

Tagging: [The/DT] [pay/NN] [is/VBZ] [given/VBN] [to/TO] [all/DT]
[employees/NNS] [with/IN] [bonus/NN] [./.]

Parse Tree: (ROOT
(S
(NP (DT The) (NN pay))
(VP (VBZ is)
(VP (VBN given)
(PP (TO to)
(NP
(NP (DT all) (NNS employees))
(PP (IN with)
(NP (NN bonus)))))))))
(. .)))

Typed Dependency (Collapsed) : `det(pay-2, The-1)`
`nsubjpass(given-4, pay-2)`
`auxpass(given-4, is-3)`
`det(employees-7, all-6)`
`prep_to(given-4, employees-7)`
`prep_with(employees-7, bonus-9)`

Fig. 1. Typed dependencies (collapsed) generated by the Stanford Parser

2.2 Homonymy

In linguistics, homonymy is a type of syntactic ambiguity in which a word in a phrase or a sentence exhibits different syntactic representations in different cases [10]. In this section, we aim to discuss various cases of homonymy that result in wrong POS tagging of English text. This problem becomes more serious, when the Stanford POS tagger wrongly identifies POS tags and due to the wrong POS tagging, the Stanford parser generates wrong parse trees and wrong dependencies. In NL2OCL project, we are totally relying on the performance of the Stanford parser. Hence, for the accurate and robust syntactic analysis, we need to address the possible cases of homonymy so that the Stanford parser may correctly parse the English text. Following are the few

examples, where the Stanford POS tagger failed to identify right POS tags and the error was propagated in the later stages of the syntax analysis performed by the Stanford parser:

English: A customer books two items.

Tagging: [A/DT] [customer /NN] [books/NNS] [two/CD] [items/NNS] [./.]

Parse Tree: (ROOT
 (NP
 (NP (DT A) (NN customer) (NNS books))
 (NP (CD two) (NNS items))
 (. .)))

Typed Dependencies, collapsed: det(books-3, A-1)
 nn(books-3, customer-2)
 num(items-5, two-4)
 dep(books-3, items-5)

Fig. 2. Parts-of-Speech tagging and parsing of text by the Stanford POS Tagger & Parser

In Fig. 2, token ‘books’ is wrongly judged as ‘NNS’ by the Stanford parser. However, the token ‘books’ is verb and the correct POS tag is ‘VBZ’. It is also shown in Fig 1 that parse tree is also wrong as VP is missing. Similarly, the typed dependencies (collapsed) are also wrongly identified `det(books-3, A-1)` should be `det(customer-2, A-1)`, `nn(books-3, customer-2)` should be `nsubj(books-3, customer-2)`, and `dep(books-3, items-5)` should be `nobj(books-3, item-5)`.

Another example of homonymy is “A customer can bank on manager”. In this example, word ‘bank’ is wrongly POS tagged ‘NN’ but the correct POS tag is ‘VB’. A similar example is “The manager made him type on typewriter.” In this example word ‘type’ is wrongly tagged as ‘NN’, while the correct tag is ‘VB’. Due to the wrong POS tagging, the parse trees of both these example is also wrongly generated by the Stanford parser.

Similar to homonymy cases, the cases for attachment ambiguity are also very important to resolve as the output of the Stanford parser is used in semantic analysis of English text and finally the output of the semantic analysis is mapped to OCL. Hence, the wrong syntax analysis results in wrong semantic analysis that ultimately generates wrong OCL.

3 Solution for Resolving Syntactic Ambiguity

To address the both types of syntactic ambiguities, discussed in section 2, we present a novel approach. We have identified that the both ambiguities are due to the absence of the context and by suing the context of the English text the correct interpretation of the ambiguous words and phrases is possible. In NL2OCL project, to translate NL

specification of constraints to OCL constraints, two inputs are required: English specification of a constraint and a UML class model. We propose the use of the information (such as classes, methods, attributes, associations, etc) available in the input UML class model for correct syntactic analysis.

The used approach for addressing the both types of syntactic ambiguities is explained in remaining part of the section.

3.1 Addressing Attachment Ambiguity

Similar to homonymy, attachment ambiguity can be resolved using the context. For generating correct dependencies of input English sentences, we again use the information on hand in the input UML class model. As, attachment ambiguity is due to the ambiguous role of noun with a preposition in a sentence. To correctly identify the attachment of the noun with other two nouns, we map the (three) candidate English elements (such as nouns) to the classes in the UML class model.

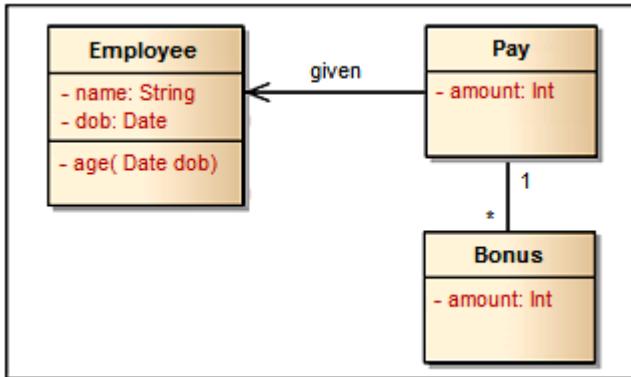


Fig. 3. A UML class model

The used mapping for attachment ambiguity resolution is slightly different from the mapping used for homonymy. First of all, the three (can be four or more) nouns are mapped to the class names in the input UML class model. Once the three classes are identified, the associations in those three classes are analyzed. With the help of the associations in the candidate classes the relationships in nouns are correctly identified. For example, the case of attachment ambiguity discussed in section 2.1 involves three nouns ‘pay’, ‘employees’, and ‘bonus’. All these three nouns are mapped to three classes (such as ‘Employee’, ‘Pay’, and ‘Bonus’) in the UML class model given in Fig. 3. After this mapping, the associations in all three classes are analyzed. The Stanford parser wrongly identifies that noun ‘bonus’ is attached to the noun ‘employees’. However, the UML class model shows that there is no relationship in classes ‘Bonus’ and ‘Employee’. While, there is a relationship in class ‘Pay’ and class ‘Bonus’. By using this information, we can correct the wrong dependencies. The corrected parse tree and dependencies are shown in Figure 4.

English: The pay is given to all employees with bonus.

Tagging: [The/DT] [pay/NN] [is/VBZ] [given/VBN] [to/TO] [all/DT]
[employees/NNS] [with/IN] [bonus/NN] [./.]

Parse Tree: (ROOT
(S
(NP
(NP (DT The) (NN pay))
(PP (IN with)
(NP (NN bonus))))
(VP (VBZ is)
(VP (VBN given)
(PP (TO to)
(NP
(NP (DT all) (NNS employees))))))
(. .)))

Typed Dependency (Collapsed) : det (pay-2, The-1)
nsubjpass (given-4, pay-2)
auxpass (given-4, is-3)
det (employees-7, all-6)
prep_to (given-4, employees-7)
prep_with (pay-2, bonus-9)

Fig. 4. Corrected typed dependencies (collapsed)

We have generalized the used approach so that all variations of the discussed type of attachment ambiguity can be handled. For this purpose, the analysis of the relationships in classes of a UML class model such as associations (directed and undirected), aggregations and generalizations can play a key role.

3.2 Addressing Homonymy

As, we have explained earlier that the absence of the context is the major reason of ambiguity. For correct POS tagging of all English sentences especially the case of homonymy, we aim to use the available information in the target UML class model such as class names, attribute names, method names, associations, etc. In syntactic analysis, once we get the output of the Stanford POS tagger, we map all the words and their tags with the UML class model and confirm that all POS tags are correctly identified.

The process of mapping of POS tagged text to the UML class model is very simple. The POS tags of all the words are mapped to the elements of the UML class model. A set of mappings were defined for this purpose as shown in Table 1. If the token matches to an operation-name or a relationship name then it is a verb or if the ambiguous token matches to a class-name or attribute-name then it is classified as a common noun or proper noun.

Table 1. Mapping of English elements to UML class model elements

UML class model elements		English language elements
Class names	→	Common Nouns
Object names	→	Proper Nouns
Attribute names	→	Generative Nouns, Adjectives
Method names	→	Action Verbs
Associations	→	Action Verbs

By using the information shown in Table 1, we can correctly POS tag the example of homonymy discussed in Section 2.2. In Figure 5, it is shown that ‘books’ is an association in two classes ‘Customer’ and ‘Item’. By using such information, it is identified that ‘books’ cannot be a noun in the context of UML class model (see figure 3). However ‘books’ can be a verb and the correct POS tag of token ‘books’ should be ‘VBZ’ as the token ‘books’ comes after a model verb (MD) ‘can’.

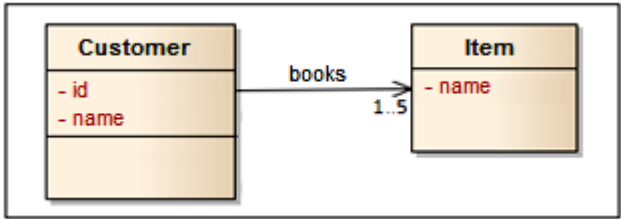


Fig. 5. A UML Class model

Once the POS tag is corrected, the parse tree and dependencies are also corrected as shown in the Figure 6.

English: A customer books two items.

Tagging: [A/DT] [customer /NN] [books/VBZ] [two/CD] [items/NNS] [./.]

Parse: (ROOT
(S
(NP (DT A) (NN customer))
(VP (VBZ books)
(NP (CD two) (NNS items)))
(. .)))

Typed Dependencies, collapsed: det(customer-2, A-1)
nsubj(books-3, customer-2)
num(items-5, two-4)
dobj(books-3, items-5)

Fig. 6. Corrected Parts-of-Speech tag, parse tree and dependencies

4 Evaluation

To evaluate the impact of the presented approach in translation of English constraints to formal (OCL) constraints, we have calculated accuracy of the NL2OCL tool before resolving cases of syntactic ambiguities and after resolving syntactic ambiguities. The cases of attachment ambiguity and homonymy are separately evaluated as below:

4.1 Evaluating NL2OCL Tool for Attachment Ambiguity Cases

We have classified the results into two types: number of correctly parsed sentences ($N_{correct}$) and number of wrongly (inaccurate) parsed sentences ($N_{incorrect}$). The Recall value and Precision value calculated for the results is shown in Table 2.

Table 2. NL2OCL Evaluation results for Attachment Ambiguity

<i>Example</i>	N_{Total}	$N_{correct}$	$N_{incorrect}$	<i>Rec%</i>	<i>Prec%</i>
Inputs	14	13	1	92.85	92.85

We have also compared the results of NL2OCL with other available tools that can perform automated analysis of the NL requirement specifications. Recall value was not available for all tools, so we have used the precision value for comparison as shown in Table 3:

Table 3. NL2OCL Evaluation results after Ambiguity Resolution

The Stanford Parser accuracy	<i>Recall</i>	<i>Precision</i>
The Stanford Parser Accuracy before Ambiguity Resolution	84.1%	84.2%
The Stanford Parser Accuracy after Ambiguity Resolution	92.8%	92.8%

Comparison in table 3 shows that by handling attachment ambiguity the accuracy of the Stanford parser was improved from 84.1% to 92.85.

4.2 Evaluating NL2OCL Tool for Homonymy Cases

Similar to the evaluation of the attachment ambiguity, the results of homonym cases are divided into two types: number of correctly parsed sentences ($N_{correct}$) and number of wrongly (inaccurate) parsed sentences ($N_{incorrect}$). The accuracy (%age) calculated for the homonymy cases is 99%.

Table 4. NL2OCL Evaluation results after resolving Homonymy

The Stanford Parser accuracy	<i>Accuracy</i>
The Stanford Parser Accuracy before Homonymy Resolution	97.0%
The Stanford Parser Accuracy after Homonymy Resolution	99.0%

Table 4 shows that the accuracy of the Stanford parser is improved from 97% to 99%. The results of this initial performance evaluation are very encouraging and support both the approach adopted in this paper and the potential of this technology in general.

5 Related Work

Natural languages are inherently ambiguous and resolution of all types of ambiguities such as lexical, syntactic, semantic ambiguities is a long standing challenge. Much work has been done in the field of natural language ambiguity identification and resolution. Some of the researchers [3], [10], [11], [13] have presented approaches to identify the various types of ambiguities in a natural language text especially the natural language software requirements. Mich showed that 90% of the software requirements are captured in a natural language [3] such as English. Hence, the resolution of ambiguities in natural language specifications of software requirements and software constraints become more critical. However, translation of natural language such English to OCL is relatively a new area of research. We aim to contribute this area of research to improve the automated software modeling from natural language software requirements that also contains constraints.

6 Conclusion and Future Work

The primary objective of the paper was to address the challenge of resolving various cases of syntactic ambiguity such as attachment ambiguity and homonymy. By resolving the said cases of syntactic ambiguity the accuracy of machine processing can be improved. To address this challenge we have presented a NL based automated approach that uses a UML class model as a context of the input English (constraints) and by using the available information in the UML class model (such as classes, methods, associations, etc) we can resolve attachment ambiguity and homonymy. The results show a significant improvement in the accuracy of the Stanford POS tagger and the Stanford parser. By improving the accuracy of the Stanford POS tagger and the Stanford parser, the accuracy of English to OCL translation is also improved to 92.85% that was earlier 84.7%.

To further improve the accuracy of English to OCL translation we need to work on semantic ambiguities, and extra semantic ambiguities such as implicatures and presuppositions.

References

1. Harmain, H.M., Gaizauskas, R.: CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis. *Automated Software Engineering* 10(2), 157–181 (2003)
2. Giordani, A., Moschitti, A.: Semantic Mapping Between Natural Language Questions and SQL Queries Via Syntactic Pairing. In: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) *NLDB 2009. LNCS*, vol. 5723, pp. 207–221. Springer, Heidelberg (2010)
3. Mich, L., Franch, M., Inverardi, P.N.: Market research for requirements analysis using linguistic tools. *Requir. Eng.*, 40–56 (2004)

4. Bajwa, I.S., Bordbar, B., Lee, M.G.: OCL Constraints Generation from NL Text. In: IEEE International EDOC Conference 2010, Vitoria, Brazil, pp. 201–214 (2010)
5. OMG: Object Constraint Language (OCL) Standard v. 2.0, Object Management Group (2006), <http://www.omg.org/spec/OCL/2.0/>
6. Marneffe, M.C., Bill, M., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: LREC 2006 (2006)
7. Cer, D., Marneffe, M.C., Jurafsky, D., Manning, C.D.: Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. In: Proceedings of LREC 2010 (2010)
8. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL 2003, pp. 252–259 (2003)
9. Manning, C.D.: Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In: Gelbukh, A.F. (ed.) CICLing 2011, Part I. LNCS, vol. 6608, pp. 171–189. Springer, Heidelberg (2011)
10. Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D.: Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements Engineering* 13(3), 207–239 (2008)
11. Uejima, H., Miura, T., Shioya, I.: Improving text categorization by resolving semantic ambiguity. *Communications, Computers and Signal Processing*, 796–799 (2003)
12. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR Business Rules Generation from Natural Language Specification. In: AAI 2011 Spring Symposium -Artificial Intelligence for Business Agility (AI4BA), San Francisco, USA, pp. 2–8 (March 2011)
13. Umber, A., Bajwa, I.S.: Minimizing Ambiguity in Natural Language Software Requirements Specification. In: IEEE 6th International Conference on Digital Information Management (ICDIM 2011), Melbourne, Australia, pp. 102–107 (September 2011)

A Computational Grammar of Sinhala

Chamila Liyanage¹, Randil Pushpananda¹, Dulip Lakmal Herath²,
and Ruvan Weerasinghe¹

^{1,2} University of Colombo School of Computing, 35, Reid Avenue,
Colombo 00700, Sri Lanka
{cml, rpn, arw}@ucsc.lk, dulip.herath@gmail.com

Abstract. A Computational Grammar for a language is a very useful resource for carrying out various language processing tasks for that language such as Grammar checking, Machine Translation and Question Answering. As is the case in most South Indian Languages, Sinhala is a highly inflected language with three gender forms and two number forms among other grammatical features. While piecemeal descriptions of Sinhala grammar is reported in the literature, no comprehensive effort to develop a context-free grammar (CFG) has been made that has been able to account for any significant coverage of the language. This paper describes the development of a feature-based CFG for non-trivial sentences in Sinhala. The resulting grammar covers a significant subset of Sinhala as described in a well-known grammar book. A parser for producing the appropriate parse tree(s) of input sentences was also developed using the NLTK toolkit. The grammar also detects and so rejects ungrammatical sentences. Two hundred sample sentences taken from primary grade Sinhala grammar books were used to test the grammar. The grammar accounted for 60% of the coverage over these sentences.

Keywords: Natural Language Processing, Context Free Grammar, Sinhala Grammar, Computational Grammar.

1 Introduction

Sinhala is the official language of Sri Lanka and it is the language spoken by a majority of Sri Lankans – nearly 70% of the population [7]. From a historical point of view, Sinhala is a modern *Indo-Aryan* language, which is related to the *Vedic* language or *Old Sanskrit* in India [10]. Modern Sinhala has subsequently gained through its association with Tamil, English, Portuguese, and Dutch [8]. There are two main varieties of Sinhala based on its usage, namely the literary and the spoken, which differ from each other in important ways [5]. In addition Sinhala has an *alphasyllabary* writing system, also called *abugida*; it is a segmental writing system in which consonant-vowel sequences are written as single units [18].

Natural Language Processing (NLP) is an area of research that explores how computers can be used to understand and manipulate natural languages [16]. Currently there are many research areas related to NLP in Sinhala such as Speech

Processing, Machine Translation, Information Retrieval, Text Summarization among others. Developing a computational grammar for Sinhala can profit such efforts. Therefore in this research we report work carried out in developing a feature-based context-free grammar for Sinhala using the open source Natural Language Tool Kit, NLTK [3].

2 Related Work

Very little research has been reported in the literature on efforts to develop a formal grammar for Sinhala. The following section describes a brief survey on grammar development reported for Indic languages including Sinhala.

Hettige and Karunananda have implemented a computational model of grammar for Sinhala [9]. Morphological and syntactic analysis of Sinhala has been considered in this work, which is modeled using a Finite State Transducer (FST) and a Context-Free grammar. Developed as part of a Machine Translation system, the parser in this system handles only simple sentences containing 8 constituents, namely, Attributive adjunct of Subject, Subject, Attributive adjunct of Object, Object, Attributive adjunct of Predicate, Attributive adjunct of the complement of predicate, Complement of predicate and Predicate.

In a research carried out for the Kannada language by Sagar et al., noun phrase – verb phrase agreement in Kannada sentences has been modeled [17]. They have classified noun phrases in to three sub categories as adjective noun, noun and pronoun, but they have only considered the gender and number as features of the grammar. Similar to the case of the Sinhala language, Kannada verbs need to agree with the subject of their sentences in number and gender. Therefore, the suffix of the verb is extracted to check masculine, feminine and plural verb endings. Here they have used the context free grammar (CFG) to write the grammar rules and used Python as the programming language. A Recursive Descent Parser, a simple top down parser, from NLTK has been used to test the grammar. This is limited to resolve noun – verb agreement and indicate whether the sentence is syntactically acceptable or not.

Sagar et al carried out another research which highlights the process of generating a Context Free Grammar for simple Kannada sentences [16]. Here they have checked the sentences with both a Top-Down (Recursive Descent) Parser and a Bottom-Up (Shift-Reduce) Parser. According to the authors, two conflicts; Shift-Reduce and Reduce-Reduce occurred when the sentences were parsed using the Bottom-Up parser. Therefore the Top-Down parser was selected as the more suitable parser to parse the given sentences.

Mosaddeque and Haque have done a research to propose a way of producing a context-free grammar for the Bangla [2]. This work reports that only sentences of seven to eight words in length are used for testing. They have taken 10 ad hoc sentences from a newspaper article as the basis for designing the grammar. They have then tagged all the words in the sentences with their respective parts-of-speech (POS) tags and used NLTK's Shift-Reduce Parser to test the grammar. Only one sentence has been successfully parsed of these ten sentences.

Naira Khan and Mumit Khan have implemented a Computational Grammar for Bengali using the Head-Driven Phrase Structure Grammar (HPSG) formalism [14]. The Linguistic Knowledge Building (LKB) system was used to implement this grammar, which allows the user to build a parser along with a generator. A set of instructions for using the HPSG formalism to parse the grammar and to generate grammatical sentences of Bengali is given in this paper.

3 Structure of Sinhala

Sinhala is a free word order language. Its unmarked word order is SOV; variant orders are also possible with discourse – pragmatic effects. A sentence can have all the possible orders of the main constituents with proper intonation [11]. Figure 1 shows all the free word order forms of the English sentence “Father hit the younger brother with a stick”.

- i. තාත්තා | මල්ලිට | කෝටුවකින් | ගැසුවේ ය.
 ʔa:ʔʔa: | malli:ʔə | ko:ʔuvəkin | gæsuvə:yə
 Father | to the younger brother | with a stick | hit
- ii. තාත්තා | කොටුවකින් | මල්ලිට | ගැසුවේ ය.
 ʔa:ʔʔa: | ko:ʔuvəkin | malli:ʔə | gæsuvə:yə
 Father | with a stick | to the younger brother | hit
- iii. මල්ලිට | තාත්තා | කෝටුවකින් | ගැසුවේ ය.
 malli:ʔə | ʔa:ʔʔa: | ko:ʔuvəkin | gæsuvə:yə
 to the younger brother | Father | with a stick | hit
- iv. මල්ලිට | කෝටුවකින් | තාත්තා | ගැසුවේ ය.
 malli:ʔə | ko:ʔuvəkin | ʔa:ʔʔa: | gæsuvə:yə
 to the younger brother | with a stick | Father | hit
- v. කෝටුවකින් | තාත්තා | මල්ලිට | ගැසුවේ ය.
 ko:ʔuvəkin | ʔa:ʔʔa: | malli:ʔə | gæsuvə:yə
 with a stick | Father | to the younger brother | hit
- vi. කෝටුවකින් | මල්ලිට | තාත්තා | ගැසුවේ ය.
 ko:ʔuvəkin | malli:ʔə | ʔa:ʔʔa: | gæsuvə:yə
 with a stick | to the younger brother | Father | hit

Fig. 1. Free word order in Sinhala

Sinhala is a head-final language, in which the complements and modifiers appear before their heads [11].

- (NP) ගමේ මිනිස්සු
 /game: minissu/
 Village-GENITIVE people
 ‘People of the village’

- (ADJP) බොහොම ලස්සන
/bohomə lassənə/
Much beautiful
Very beautiful

- (VP) සෙමින් කියවයි
/semin kiyəvayi/
Slowly Read-non past/3rd person singular
Read slowly

Traditionally, a sentence is divided in to two parts; Noun Phrase (NP), and Verb Phrase (VP). In Sinhala grammar, *uktha* (subject) and *akyatha* (predicate) are the two parts of a sentence. Subject and predicate in Sinhala sentences agree in number, gender and person [12].

The studies of sentence structures of Sinhala have been made by a number of scholars [8] [1] [4]. According to Abayasingha [1] Sinhala has 25 types of simple sentence structures. However in the present work, we have covered only the main sentence structures and a few complex structures. These are described in the following sections.

3.1 Noun Phrase

The Noun Phrase, denoted by NP, can be a common noun (N), pronoun (PrN) or a proper noun (PropN). In addition to the head noun, the Sinhala noun phrase consists of adjectival phrases and determiners. Sinhala NP has a very complex grammatical structure. It can consist of various clause structures, such as adjectival clauses, relative clauses, and subordinate clauses. Therefore building a computational grammar, covering all the NP structures is complex. Figure 2 below shows the NP structure we have covered in the grammar developed in this work.

An adjectival phrase (ADJP) is constructed with adjectives. According to Sinhala grammar, an adjectival phrase comes before the Noun (N) and after the Determiner (Det), if there is any determiner in the noun phrase. If the adjective is a qualitative adjective, then it can be constructed with Degrees (Deg) to intensify its meaning.

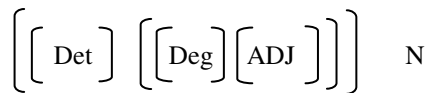


Fig. 2. structure of the NP in Sinhala

In the traditional grammar of Sinhala, *nama visheshana* (adjectives) denote some quality or attribute of the noun. It can be divided into three classes, namely qualitative, quantitative and demonstrative [8]. However in our grammar we do not consider the features of the ADJP. The words which denote the degree of the adjectives are added only before qualitative adjectives. i.e. the ADJP ‘ඉතා හොඳ’ /iṭa:

hoⁿḍə/ (very good) is an adjectival phrase and it consists of a degree ‘ඉතා’ /iṭa:/ (very) and a qualitative adjective ‘හොඳ’ /hoⁿḍə/ (good), which appears before the adjective.

3.2 Verb Phrase

According to generative grammar, a Verb Phrase (VP) is a phrase headed by a verb. In addition to the verb, it consists of noun phrases and Adverbial Phrases (ADVP). The verb in Sinhala can be categorized as single verbs, compound verbs and auxiliary verbs. In this grammar we only consider single verbs. Generally ADVP occurs before the verb in Sinhala sentences. However according to the features of the adverb, the position where the adverb occurs is decided. Figure 3 below shows the VP structure in Sinhala and what we have covered in the grammar. According to the structure, the verb appears in the final position. ADVPs may appear both before the verb and after the NP. If an adverb of manner occurs in the ADVP, the adverb can be combined with degrees to intensify the meaning. For example ‘ඉතා වේගයෙන්’ /iṭa: ve:gəyen/ (very fast) is an ADVP which appears as an adverb of manner.



Fig. 3. structure of the VP in Sinhala

4 Grammatical Features of Sinhala

According to the Sinhala language the noun is inflected for number, gender, person, tense, case and definiteness. The verb is inflected for number, gender, person, tense and volition. Subject and predicate agree for the features of number, gender and person.

4.1 Grammatical Features of the NP

The words that are marked for grammatical features *linga* (gender), *vachana* (number), *niyatha-aniyatha* (definiteness) and *vibhakthi* (case), are recognized as nouns in Sinhala [12]. Therefore in developing the grammar, we consider the features of number, gender, case and definiteness for common nouns; number, gender and case for proper nouns; and number, gender, case and person for pronouns.

As a highly inflected language, common nouns in Sinhala are inflected for number, definiteness and case. Sinhala nouns are also divided into animate and inanimate classes on the basis of their inflection. Animate nouns inflect for number (singular and plural), definiteness (definite and indefinite) and five cases (nominative, accusative, dative, genitive, and instrumental). The definiteness distinction applies only in the singular form of nouns [6].

As enumerated in Table 1, Sinhala nouns, which are inflected for five cases, have five forms. However the same form may occur in several cases for nouns. For example form 5 can occur in the cases instrumental, ablative and auxiliary. Therefore we have defined the five cases in this grammar specification. All inflections relating to animate nouns are shown in the table.

Table 1. Examples for inflections of animate common nouns

Form	Case	Singular				Plural
		Masculine		Feminine		
		Def.	Indef.	Def.	Indef.	
1	Nominative	/miɳisa:/ (the man)	/miɳisek/ (a man)	/kella/ (the girl)	/Kellak/kellek/ (a girl)	/miɳissu/ (men)
2	Accusative	/miɳisa:/ (the man)	/miɳiseku/ (a man)	/kella/ (the girl)	/kellakə/ (a girl)	/miɳisun/ (men)
3	Dative	/miɳisa:tə/ (to the man)	/miɳisekətə/ (to a man)	/kella:tə/ (to the girl)	/kellekətə/ kellakətə (to a girl)	/miɳisuntə/ (to men)
4	Genitive	/miɳisa:ge:/ (the man's)	/miɳisekuge:/ (a man's)	/kellage:/ (the girl's)	/kellakəge:/ (a girl's)	/miɳisunge:/ (men's)
5	Instrumental	/miɳisa:geɳ/ (from the man)	/miɳisekugeɳ/ (from a man)	/kellageɳ/ (from the girl)	/kellakəgeɳ/ (from a girl)	/miɳisungeɳ/ (from men)

Sinhala inanimate nouns are inflected similarly to the animate nouns for number and definiteness. However they only have four cases – direct, dative, genitive and instrumental [6]. Forms 3, 4, and 5 in Table 2 are similar to those in Table 1. However form 1 accounts for the direct case. Table 2 shows all the inflections for an inanimate noun as covered in the grammar developed.

Table 2. Examples for inflections of inanimate common nouns

Form	Case	Singular		Plural
		Definite	Indefinite	
1	Direct	/gasə/ (the tree)	/gasak/ (a tree)	/gas/ (trees)
3	Dative	/gasətə/ (to the tree)	/gasəkətə/ (to a tree)	/gaswələtə/ (to the trees)
4	Genitive	/gase:/ /gasehi/ on the tree	/gasəkə/ on a tree	/gaswələ/ on the trees
5	instrumental	/gaseɳ/ /gasɳ/ (from the tree)	/gasəkɳ/ (from a tree)	/gaswələɳ/ (from trees)

Determiners of Sinhala do not carry any grammatical features and can engage with any noun without agreement of features. Therefore grammatical features for determiners were not considered. e.g. ‘එ’ /e:/ is a determiner of Sinhala which combines with any noun without considering grammatical features of number, gender or case. The noun phrases ‘එ ළමයා’ /e: laməja:/ (that child) and ‘එ ළමයි’ /e: lamaji/ (those children) differ in the number feature, but have the identical determiner ‘එ’.

4.2 Grammatical Features of the VP

Number, gender, tense, person and volition are considered as the grammatical features of the verb phrase (VP) in Sinhala. There are two tenses in Sinhala; past and non-past. The non-past form can refer either to past or future. The future tense is expressed using time adverbials. Therefore the single form that is used to denote both tenses Present and Future is termed non-past. For example ‘යයි’/jaji/ is a verb form in Sinhala which means ‘goes’ and carries the grammatical features singular, 3rd person, non-past. i.e. the sentence ‘ඔහු පාසලේ යයි’ /ohu pa:sal jaji/ (he goes to school) is in the present tense. If we add the time adverbial ‘හෙට’/heṭṭə/ to denote the future, then the sentence would be in future tense; ‘ඔහු හෙට පාසලේ යයි’ /ohu heṭṭə pa:sal jaji/ (he will go to school tomorrow). In these two sentences the verb ‘යයි’ /jaji/ is a pure verb. In addition to the pure form, the form which used to denote the non-past tense is called the *krudantha*. We can change the above sentence with a *krudantha* form as ‘ඔහු හෙට පාසලේ යන්නේය’ /ohu heṭṭə pa:sal jaṅṅe:jə/. In this sentence ‘යන්නේය’ /jaṅṅe:jə/ is similar to the form ‘යයි’. However, according to Kekulawala [13], the *krudantha* form is non-past; which uses -න්නෙ- /-ṅṅe-/ suffix, is used to denote the future tense in Sinhala. In modern Sinhala writings, *krudantha* forms are used more frequently than the pure forms in both past and non-past tenses. In this grammar, we considered the tense as past and non-past other than past, present and future.

Volition (VLT) is another feature of the verb which in our grammar is considered to be either true or false. For example ‘යයි’ is a *volitive* form of the verb “go”, while its equivalent *involitive* form is ‘යැවෙයි’ /jæveji/. The other features of number, gender and person are the same as their equivalents in the NP. Figure 4 gives an overview of the grammatical features of the Sinhala Verb.

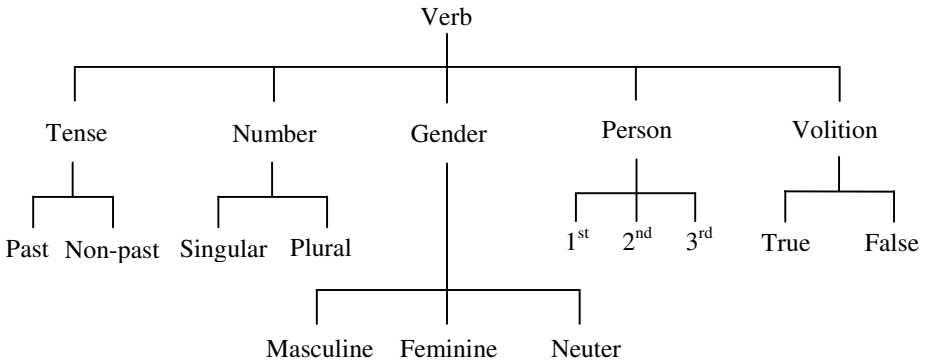


Fig. 4. Grammatical Features of the Verb Phrase

5 The Sinhala CFG

According to Abhayasinghe (1998) there are 25 types of simple sentence structures in Sinhala [1]. In this research, we considered the following ten types of sentence structures in developing a CFG for Sinhala.

1. සඳු | බැබළෙයි.
saⁿḍə | bæbøleji
moon | is shining.
The moon is shining.
2. ගසෙන් | බිමට | ගෙඩියක් | වැටිණි.
gaseṅ | biməṭə | gedijak | væṭiṅi
from the tree | to the floor | a fruit | fell
A fruit fell (to the floor) from the tree.
3. බල්ලෝ | බුරුනි.
ballo: | burəṅi
dogs | bark
Dogs bark.
4. අයියා | අඹ | කඩයි.
ajja: | a^mbə | kaḍaji
the elder brother | mangoes | plucks
The elder brother plucks mangoes.
5. තාත්තා | පුතාට | තෑග්ගක් | දෙයි.
ta:ṭṭa: | puṭa:ṭə | tæ:ggak | deji
father | to the son | a gift | gives
Father gives a gift to the son.
6. මිනිසා | ගසට | නගීයි.
miṅisa: | gasaṭə | ṅagiji
the man | to the tree | climbs
The man climbs the tree.
7. හිඟන්නා | මගෙන් | රුපියලක් | ඉල්ලීය.
hi^ggaṅṅa: | magenṅ | rupiyəlak | illi:jə
the beggar | from me | one rupee | asked
The beggar asked me one rupee.
8. ඔහුට | වත්තක් | තිබේ.
ohuṭə | vattak | tibe:
to him | an estate | has
He has an estate.
9. මට | සින්දුවක් | ඇසෙයි.
maṭə | siṅḍuvak | əseji
to me | a song | hear
I hear a song.
10. ළමයාට | ඇඬිණි.
ləməja:ṭə | æⁿḍiṅi
to the child | cried
The child cried.

After analyzing each of the above sentences, all their constituents were identified. According to this constituent structure, separate CFG productions were generated for each type of sentence. After identifying all the grammar rules needed to cover the

phenomena above, they were merged together to form and optimize a generic CFG for Sinhala. In addition, some more complexity in the grammatical rules was also introduced to the Sinhala CFG in order to increase its overall coverage. The grammatical and lexical productions of the CFG developed are given below.

Grammar Productions

----- *S expansion productions* -----

S -> NP[NUM=?n, GEN=?G, PER=?P, DEF=?TF, CASE=F1] VP[NUM=?n, GEN=?G, PER=?P, CASE=?CS]
 S -> NP[NUM=?n, GEN=?G, PER=?P, DEF=?TF, CASE=F3] VP[NUM=?n, GEN=?G, PER=?P, CASE=?CS]
 S -> NP[NUM=?n, GEN=?G, PER=?P, DEF=?TF, CASE=F4] VP[NUM=?n, GEN=?G, PER=?P, CASE=?CS]
 S -> NP[NUM=?n, GEN=?G, PER=?P, DEF=?TF, CASE=F5] VP[NUM=?n, GEN=?G, PER=?P, CASE=?CS]

----- *NP expansion productions* -----

NP[NUM=?n, CASE=?CS, GEN=?G, DEF=?TF] -> N[NUM=?n, CASE=?CS, GEN=?G]
 NP[NUM=?n, CASE=?CS, GEN=?G, PER=?P] -> PrN[NUM=?n, CASE=?CS, GEN=?G, PER=?P]
 NP[NUM=?n, CASE=?CS] -> PropN[NUM=?n, CASE=?CS]
 NP[NUM=?n, CASE=?CS, GEN=?G, DEF=?TF] -> Det N[NUM=?n, CASE=?CS, GEN=?G]
 NP[NUM=?n, CASE=?CS, GEN=?G, DEF=?TF] -> ADJP N[NUM=?n, CASE=?CS, GEN=?G]
 NP[NUM=?n, CASE=?CS, GEN=?G, DEF=?TF] -> Det ADJP N[NUM=?n, CASE=?CS, GEN=?G]

----- *VP expansion productions* -----

VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> IV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP IV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP NP TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP NP IV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP NP ADVP TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> ADVP IV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> ADVP TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP ADVP IV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]
 VP[TENSE=?t, NUM=?n, GEN=?G, PER=?P] -> NP ADVP TV[TENSE=?t, NUM=?n, GEN=?G, PER=?P]

----- *ADJP expansion productions* -----

ADJP -> Adj
 ADJP -> Adj ADJP

----- *ADVP expansion productions* -----

ADVP -> Adv
 ADVP -> Adv ADVP

----- *Sample Lexical Productions* -----

N[NUM=sg, GEN=MA, CASE=F1, DEF=TRUE] -> 'බල්ලා' | 'මිනිසා' | 'ලමයා' | 'තාත්තා' | 'හිතන්නා' | 'මල්ලී' | 'අයියා'
 N[NUM=sg, GEN=MA, CASE=F2, DEF=TRUE] -> 'බල්ලා' | 'මිනිසා' | 'ලමයා' | 'කොල්ලා' | 'තාත්තා' | 'සර්පයා'
 N[NUM=sg, GEN=MA, CASE=F3, DEF=TRUE] -> 'බල්ලාට' | 'මිනිසාට' | 'ලමයාට' | 'පුතාට' | 'සර්පයාට' | 'අයියාට'
 N[NUM=sg, GEN=MA, CASE=F1, DEF=False] -> 'බල්ලෙක්' | 'මිනිසෙක්' | 'ලමයෙක්' | 'කොල්ලෙක්' | 'හිතන්නෙක්' | 'පුතෙක්'
 N[NUM=sg, GEN=NE, CASE=F1, DEF=False] -> 'තැඟ්ගෙක්' | 'රුපියලක්' | 'වත්තක්' | 'සින්දුවක්' | 'පොතක්' | 'සෙවියක්'
 N[NUM=sg, GEN=NE, CASE=F3, DEF=TRUE] -> 'ගසට' | 'මලට' | 'අත්තට' | 'බතට' | 'ගෙදරට' | 'පොතට' | 'කෝටුවට' | 'බිමට'
 N[NUM=sg, GEN=NE, CASE=F5, DEF=TRUE] -> 'ගසෙන්' | 'මලෙන්' | 'ගෙදරින්' | 'පොතින්' | 'කෝටුවෙන්' | 'පොරවෙන්'
 N[NUM=sg, GEN=NE, CASE=F1, DEF=TRUE] -> 'ගස' | 'මල්' | 'අත්ත' | 'අඹ' | 'ගෙදර' | 'පොත' | 'කෝටුව' | 'කළය' | 'සඳ'
 N[NUM=pl, GEN=MA, CASE=F1] -> 'බල්ලෝ' | 'මිනිස්සු' | 'ලමයි'
 N[NUM=pl, GEN=FE, CASE=F2] -> 'ගැහැණුන්' | 'කෙල්ලන්'
 PrN[NUM=sg, CASE=F3, PER=F] -> 'මට'
 PrN[NUM=sg, CASE=F5, PER=F] -> 'මගෙන්' | 'මාගෙන්'
 PrN[NUM=pl, CASE=F1, PER=T] -> 'මවුහු' | 'ඒගොල්ලෝ'
 Det -> 'ඒ' | 'මේ' | 'ඊ' | 'මය' | 'සමහර' | 'ඈතම'
 Adj -> 'ලස්සන' | 'කැත' | 'මහන' | 'සුදු' | 'කලු' | 'ලොකු' | 'පොඩි' | 'පුළු' | 'උස'
 Adv -> 'පත්සල්' | 'ගෙදර' | 'පාසලට' | 'තරගයට' | 'වෙහෙයන්' | 'ලස්සනට' | 'ගොදට' | 'ඉක්මනින්' | 'සෙමෙන්'

TV[TENSE=nPast, NUM=sg, GEN=MA, VLT=True, PER=T] -> 'දෙයි' | 'කයි' | 'කඩයි' | 'කියයි' | 'ගසයි' | 'සිටිය' | 'කිවේ'
 TV[TENSE=nPast, NUM=sg, VLT=False] -> 'කැවෙයි' | 'කියවෙයි' | 'ඇසෙයි' | 'පෙවෙයි'
 TV[TENSE=nPast, NUM=pl, VLT=True, PER=F] -> 'දෙමු' | 'කමු' | 'මොමු' | 'කියමු'
 TV[TENSE=past, NUM=sg, GEN=MA, VLT=True, PER=T] -> 'කැවෙය' | 'දුන්නේය' | 'කිවෙය' | 'ඉල්ලීය' | 'ගසුවෙය'
 IV[TENSE=nPast, NUM=sg, GEN=MA, VLT=True, PER=T] -> 'බුරයි' | 'නගිය' | 'ඇවිදීයි' | 'යයි'
 IV[TENSE=nPast, NUM=sg, GEN=NE, VLT=False, PER=T] -> 'බැබලෙයි' | 'පෙපයි'
 IV[TENSE=nPast, NUM=pl, VLT=True, PER=S] -> 'බුරුහු' | 'ඇවිදිහු' | 'යහු'
 IV[TENSE=past, NUM=sg, VLT=False, PER=T] -> 'බිටිණි' | 'ඇඬිණි' | 'වැටිණි'

Following are the parse trees that have been produced using the Recursive Decent parser from the NLTK toolkit [3].

-----Sentence 3-----
 (S[]
 (NP[CASE='F1', DEF=?TF, GEN='MA', NUM='pl']
 (N[CASE='F1', GEN='MA', NUM='pl'] බල්ලෝ))
 (VP[GEN=?G, NUM='pl', PER='T', TENSE='pres']
 (IV[NUM='pl', PER='T', TENSE='nPast', +VLT] බුරනි)))

-----Sentence 5-----
 (S[]
 (NP[CASE='F1', DEF=?TF, GEN='MA', NUM='sg']
 (N[CASE='F1', DEF='TRue', GEN='MA', NUM='sg'] තාත්තා))
 (VP[GEN='MA', NUM='sg', PER='T', TENSE='pres']
 (NP[CASE='F3', DEF=?TF, GEN='MA', NUM='sg']
 (N[CASE='F3', DEF='TRue', GEN='MA', NUM='sg'] පුතාට))
 (NP[CASE='F1', DEF=?TF, GEN='NE', NUM='sg']
 (N[CASE='F1', -DEF, GEN='NE', NUM='sg'] තැන්ගක්))
 (TV[GEN='MA', NUM='sg', PER='T', TENSE='nPast', +VLT]
 දෙයි)))

-----Sentence 9-----
 (S[]
 (NP[CASE='F3', GEN=?G, NUM='sg', PER='F']
 (PrN[CASE='F3', NUM='sg', PER='F'] මට))
 (VP[GEN=?G, NUM='sg', PER=?P, TENSE='pres']
 (NP[CASE='F1', DEF=?TF, GEN='NE', NUM='sg']
 (N[CASE='F1', -DEF, GEN='NE', NUM='sg']
 සින්දුවක්))
 (TV[NUM='sg', TENSE='nPast', -VLT] ඇසෙයි)))

6 Evaluation and Results

In order to test and evaluate the grammar, two hundred sample sentences taken from primary grade Sinhala Grammar books [19] [20] were used. According to the test, 118 sentences were parsed the grammar correctly and 82 were not parsed. Out of 82

sentences two sentences are structured incorrectly and therefore they were restricted from the grammar. Several sentences were not parsed because of the free word order. For example, in this grammar ADVP is used before the verb and after the NP. However, the sentences which have ADVP at the beginning were also not parsed through the grammar.

If an inanimate noun occurs in the subject NP, it does not agree on number with the predicate VP. i.e. the following sentence ‘මල පිපෙයි’ /malə pipeji/ (the flower blooms) contains a singular NP and singular VP, while ‘මල් පිපෙයි’ /mal pipeji/ (flowers bloom) contains a plural NP and singular VP. According to Sinhala language, both of these sentences are correct. However the second type of sentences; which does not consider the number, has not been covered in this grammar. Sentences which have compound verbs, auxiliary verbs, present participles, past participles, the verbs which have imperative mood and negation of the verbs are also not parsed through this grammar.

The test results are shown below.

Total Number of Sentences	200
Correct sentences parsed	118
Correct sentences not parsed	80
Incorrect sentences not parsed	2

According to the result, accuracy of this grammar is 60%.

7 Discussion

Free Word order

The grammar developed covers the default Sinhala sentence structure in the SOV order. The first two sentences of Figure 1 are in SOV order, and only they can be successfully parsed using the grammar developed. The rest of the sentence structures can't be parsed using the existing grammar. In natural language processing, dependency grammars are used to solve the free word-order problem.

Word segmentation

In written Sinhala there is no unique method for word segmentation. The linguistics literature reports on collections of rules for segmenting Sinhala words [15]. However most users of the language are not aware of these rules and do not follow them closely for word segmentation. For example the word-ending particle ‘ය’ is often used inconsistently. The Sinhala language has two types of verbs, namely *shudda kriya* ‘pure verbs’ and *krudanta kriya* ‘participial verbs’. When a participial verb occurs in the sentence ending position there are two ways to write it. One is by separating the sentence-ending particle as in the case of ‘ගියේ ය’ “(he) went” and adding it to the participial verb as ‘ගියේය’. Owing to this, it is desirable to have a word segmentation algorithm to check whether the text is in a normalized form before the CFG parser is employed.

Non verbal sentences

There are number of sentence structures in Sinhala which do not contain a verb. These types of sentences end with adjectives, oblique nominals, locative predicates and adverbials among others, and the current grammar does not cover such non-verbal sentences of Sinhala.

8 Conclusion and Future Work

This paper describes the development of a CFG for a non-trivial subset of Sinhala using the NLTK toolkit. Ten simple sentence structures were selected and used to design the grammar. Two hundred simple sentences were used to test the grammar and 60% sentences were analyzed accurately the parser. In the future, it is hoped to use a morphological analyzer and a word segmentation algorithm to develop a more wide-coverage grammar for Sinhala.

Acknowledgment. We are grateful to all the members of Language Technology Research Laboratory of the University of Colombo School of Computing, Sri Lanka, who helped in various ways to make this work bear fruit.

References

1. Abhayasinghe, A.A.: Sinhala bhashave sarala vakya vibagaya (1998)
2. Ayesha Binte Mosaddeque, A.B., Haque, N.: Context-Free Grammar for Bangla. BRAC University, Dhaka
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media (2009)
4. Disanayaka, J.B.: Bashavaka rata samudaya. Lake house investment Co. Ltd., Colombo 2 (1969)
5. Fairbanks, G.H., Gair, J.W., Silva, M.W.S.D.: Colloquial Sinhalese. Cornell University, New York (1968)
6. Gair, J.W., Karunatilaka, W.S.: Literary Sinhala inflected forms: A Synopsis with a Translation Guide to Sinhala script. Cornell University, New York
7. Gair, J.W., Karunatilaka, W.S.: Literary Sinhala. Cornell University, New York (1974)
8. Gunasekara, A.M.: A Comprehensive Grammar of the Sinhalese Language. Godage International Publishers (PVT) Ltd. (2008)
9. Hettige, B., Karunananda, A.S.: Computational Model of Grammar for English to Sinhala Machine Translation. In: Proceedings of the International Conference on Advances in ICT for Emerging Regions (2011)
10. Jayawardhane, T.: The surface case system in Sinhala. KALYANI, pp. 264–277. University of Kelaniya (1996)
11. Kariyakarawana, S.M.: The Syntax of Focus and Wh-Questions in Sinhala. Karunaratne & Sons Ltd. (1998)
12. Karunatilaka, W.S.: Sinhala bhasha vyakaranaya. M. D. Gunasena & Co. Ltd. (2009)
13. Kekulawala, S.L.: The future tense in Sinhalese – an ‘unorthodox’ point of view. Journal of the Vidyalandara University of Ceylon (1972)

14. Khan, N., Khan, M.: Developing a Computational Grammar for Bengali Using the HPSG Formalism. In: Proceedings of the 9th International Conference on Computer and Information Technology, ICCIT 2006 (2006)
15. Rajapaksha, D.: Sinhala bhashave pada bedima saha virama lakshana bhavithaya (2008)
16. Sagar, B.M., Shobha, G., Kumar, R.: Context Free Grammar (CFG) Analysis for simple Kannada sentences. In: Proceedings of the International Conference [ACCTA-2010] on Special Issue of IJCCT, vol. 1(2, 3, 4) (2010)
17. Sagar, B.M., Shobha, G., Kumar, R.: Solving the Noun Phrase and Verb Phrase Agreement in Kannada Sentences. International Journal of Computer Theory and Engineering 1(3) (August 2009)
18. Wikipedia (English), http://en.wikipedia.org/wiki/Sinhala_language
19. Dasanayaka, A.E.S.: Kumara rachanaya; Grade 4, M. D. Gunasena & Co. Ltd. (1990)
20. Dasanayaka, A.E.S.: Kumara rachanaya; Grade 5, M. D. Gunasena & Co. Ltd. (2005)

Automatic Identification of Persian Light Verb Constructions

Bahar Salehi¹, Narjes Askarian¹, and Afsaneh Fazly²

¹ School of Electrical and Computer Engineering, Shiraz University, Iran
baharsalehi@gmail.com, askarian@cse.shirazu.ac.ir

² School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran
afsaneh.fazly@gmail.com

Abstract. Multiword expressions pose a challenge to the development of large-scale, semantically-rich Natural Language Processing (NLP) systems. We use a bilingual parallel corpus for automatically extracting Light Verb Constructions (LVCs), a very common type of multiword expressions in many languages, including Persian. Using two classifiers, we investigate the usefulness of seven linguistically-informed features for automatically identifying Persian LVCs. To our knowledge, this is the first attempt at the automatic detection of a broad class of Persian LVCs. Results of our experiments show that the proposed features are reasonably successful at the task.

1 Introduction

Automatic identification of multiword expressions (MWEs) plays an important role in the development of NLP applications, such as machine translation, textual entailment, and summarization. In particular, verb-based MWEs are widely used in many languages, including English [Stevenson et al., 2004], Urdu [Buti, 2003], Persian [Karimi, 1997], among others. Light verb constructions (LVCs) are a type of verb-based MWEs, in which a semantically-light *basic verb*¹ is combined with a co-verbal component that can be a noun, a verb, an adjective, or a prepositional phrase. LVCs have distinct properties, the most important of which being that most of the predicative meaning of the full expression comes from the co-verbal element, as in *take a walk* or *give a talk* in English. Our focus here is on the automatic detection of LVCs in Persian. This is a particularly challenging and important problem since (i) LVCs are very common and highly productive in Persian, and they greatly outnumber simple verbs [Khanlari, 1973]; and (ii) to our knowledge there has not been much work on their automatic identification.

Nonetheless, there is much research in the linguistics literature on the special syntax and semantics of Persian LVCs [Karimi, 1997, Karimi-Doostan, 1997, Dabir-Moghaddam, 1997, Megerdoomian, 2004, Karimi-Doostan, 2005]. In addition, there are studies on the automatic processing of LVCs and other MWEs in other languages, often using statistical measures [Baldwin and Villavicencio, 2002,

¹ Following [Fazly, 2007] we call these verbs *basic* since they refer to states or acts that are central to human experience.

Stevenson et al., 2004, Evert and Krenn, 2005, Villada Moirón and Tiedemann, 2006, Villavicencio et al., 2007, Fazly and Stevenson, 2007].

We draw on the above studies to develop an appropriate method for the computational treatment of Persian LVCs. Specifically, our goal is to distinguish Persian LVCs, such as *âdash zadan* (literally *fire hit*, meaning ‘to burn sth.’ or ‘to put sth. on fire’), from non-LVCs, which may include similar-on-the-surface literal combinations such as *sang zadan* (literally *stone hit*, meaning ‘to hit sth. with a stone’), as well as erroneous combinations. We treat this problem as a classification task, and use seven linguistically-informed features for the purpose. We extend and use some of the features introduced in previous studies [Melamed, 1997, Villada Moirón and Tiedemann, 2006], but also introduce novel features.

We first explain our features in Section 2. Section 3 then presents our experimental setup, and Section 4 reports the results of the experiments. Finally, Section 5 concludes the paper.

2 Features

We use seven features drawn from the translational distributions of our candidate Persian expressions in a second (target) language (here, English). We use a word-aligned bilingual (English–Persian) corpus to extract these features.

Word alignments are links between the word tokens of two languages in a bilingual corpus, which are constructed as a pre-step for building translation models [Brown et al., 1993]; see Figure 1 for example word alignments/links between words of a Persian sentence (*Sima be Ali sang zad*, literally *Sima to Ali stone hit*) and its corresponding English translation (*Sima hit Ali with a stone*). We use a word aligner that produces asymmetric alignments (see Section 3 for details) — i.e., it is possible to have many-to-one alignments where more than one source words are aligned to one target word. Thus, source to target (here, Persian to English or Pr2En) word alignments are not the same as target to source (here, English to Persian or En2Pr) alignments. Some of our proposed features make use of this asymmetry.

We include three groups of features in our study: The first group consists of three entropy-based features (#1–#3), inspired by the work of [Melamed 1997] and [Villada Moirón and Tiedemann 2006]. The second group contains two novel features, (#4 and #5) that compare the links of the components of a candidate expression in the two alignment directions (Pr2En and En2Pr). The last group of features (#6 and #7) look into the frequency of two different types of alignments in the Pr2En direction. Next, we elaborate on each of these groups of features.

2.1 Entropy-Based Features

[Melamed 1997] presents a method for measuring semantic entropy of words using translational distributions of words in word-aligned parallel corpora. This entropy is interpreted as a measure of semantic ambiguity — that is, a word with a high entropy is very ambiguous and its translations are thus less accurate. [Villada Moirón and Tiedemann 2006] and [Villavicencio et al. 2007] use entropy-based measures for identifying MWEs (in Dutch and English, respectively), and show

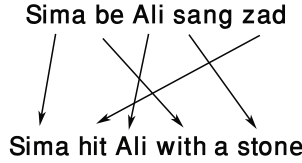


Fig. 1. Sample Persian-to-English word alignment

that they are reasonably successful at the task. We use the two entropy-based measures proposed by Villada Moirón and Tiedemann [2006] as features in our classification experiments (Features #1 and #2); and in addition include a third measure that combines these two (Feature #3).

Feature #1. Forward entropy, defined as the entropy of the translation links of a Persian candidate expression in the Pr2En alignment direction.

Like Villada Moirón and Tiedemann [2006], here we assume that a verb that is not part of an LVC has a literal meaning, and hence its translations into English tend to be consistent. In contrast, a basic verb in an LVC has a non-compositional meaning so we expect an automatic word aligner to produce a large variety of links for such a verb. For example a simple verb such as *zad* ('hit') in *Sima be Ali sang zad* (literally *Sima to Ali stone hit*, meaning 'Sima hit Ali with a stone') is expected to be consistently aligned to *hit*. But, the same verb (*zad*) used in the context of an LVC *âdash zad* — literally *fire hit*, meaning 'burned' or 'put on fire' — may be aligned to any of the words *burned*, *put*, *on*, *fire*, or none, among others. Thus, we expect the number of different translation links of *zad* in an LVC to be more than those of *zad* as a simple verb. We use entropy as a measure of this uncertainty in translation.

To compute the entropy of the translation links of a source bigram S , we collect all the translation links of the two components of the bigram. Each component is a word s (in Persian). Thus, for each bigram, we gather two lists of translation links $T_{s|S}$. For example, if the components of *âdash zad* are respectively aligned to *fire* and *put* in one case, and in another case are both aligned to *burned*, the two translation lists are:

$$T_{zad|\hat{a}dash\ zad} = \{put, burned\}$$

$$T_{\hat{a}dash|\hat{a}dash\ zad} = \{fire, burned\}$$

First, we calculate an entropy for each component s in a candidate expression S , as in:

$$H(T_{s|S}|s) = - \sum_{t \in T_{s|S}} P(t|s) \log P(t|s) \quad (1)$$

where $P(t|s)$ is the proportion of target word t among all alignments of the source word s in the corpus, and is calculated as in:

$$P(t|s) = \frac{\text{frequency}(s \rightarrow t \text{ alignments})}{\text{size}(T_{s|S})} \quad (2)$$

Next, we measure the forward entropy of the bigram S as the average of the entropies of its components, as also done by [Villada Moirón and Tiedemann \[2006\]](#).

Feature #2. Backward entropy, defined as the entropy of the translation links of a Persian candidate expression in the En2Pr alignment direction.

LVCs are very common in Persian and much less common in English. We thus expect many Persian LVCs to have a one-word translation in English, e.g., *âdash gereft* is translated to *burned*. However, since our aligner cannot produce any one-to-many alignments (one English word to more than one Persian words) in the En2Pr direction, an English verb such as *burned* can only be aligned to one of the components of its Persian translation, *âdash* or *gereft*. Recall that the noun component of an LVC is the main contributor of the predicative meaning. Hence, it is more likely that in such cases, the English verb (*burned*) is aligned to the noun component of the Persian LVC (*âdash*), and that the basic verb (*gereft*) is not aligned to any words in English (a.k.a. No-Link). If we count the No-Links as different links, we expect the entropy of an LVC to be higher than that of a non-LVC in the En2Pr alignment direction.

Feature #3. Average entropy, which is the average of the forward and the backward entropy.

2.2 Features Comparing Alignment Directions

Our next two features compare the alignments between the components of a Persian candidate expression and its English translation in the two alignment directions. For these two features, we use the notion of *common links*, which we define to be those links that appear in both alignment directions of the same sentence pair. For example, consider this pair of parallel sentences:

Persian: *Sima be Ali sang zad*
 English: *Sima hit Ali with a stone*

If we see the link *hit*→*zad* in the En2Pr alignment direction, and the link *zad*→*hit* in the Pr2En direction, this link is counted as a common link.

We note that, even after filtering, our list of candidate expressions contains noise (see Section 3.1 for details). Feature #4 below is meant to separate LVCs from literal non-LVCs, but Feature #5 is designed to separate noise from both LVCs and literal phrases. Next, we explain each feature in detail.

Feature #4. Proportion of the common links of the verb to those of the noun component. As mentioned above (for Feature #2), for an LVC, we expect to see notable differences in the two alignment directions, mainly because of the asymmetry of our aligner. In contrast, for a non-LVC, the words are expected to be linked to their English translations in both alignment directions. For example, for the non-LVC *sang zad* (literally *stone hit*), we expect to have *sang* aligned to *stone* and *zad* to *hit* in both alignment directions. We thus expect the noun component of an LVC to have many common links, whereas

the basic verb is expected to have very few common links — that is, we expect the following proportion to be small for LVCs, and large for literal phrases.

$$\frac{\text{common links(verb)}}{\text{common links(noun)}} \quad (3)$$

Feature #5. Proportion of the common links of the noun component, as in:

$$\frac{\text{common links(noun)}}{\text{links(noun)}} \quad (4)$$

where links(noun) refers to the total number of alignments of the noun in the Pr2En alignment. We expect the above proportion to be high for both LVCs and literal non-LVCs, since in these combinations the noun contributes an independent meaning (the predicative meaning in an LVC, and a nominal meaning in a literal non-LVC). In contrast, for the erroneous combinations, we expect there to be few common links, and this proportion to be small.

2.3 Features Tapping into Alignment Type

We look at two types of alignments in the Pr2En direction: 2-to-1 alignments, and 2-to-2 alignments. A 2-to-1 alignment is when two source (Persian) words are aligned to the same target (English) word, whereas a 2-to-2 alignment is when the two Persian words are each aligned to a different English word.

Features #6 and #7. measure the frequency of 2-to-1 alignments and the frequency of 2-to-2 alignments (for a Persian candidate expression), respectively. We assume that whenever the components of a candidate bigram are aligned to one word in English (2-to-1 alignment), the candidate is more likely to be an LVC. In contrast, the two words of a non-LVC are more likely to be aligned to two distinct words in English (2-to-2 alignment). We thus expect Feature #6 to have higher values for LVCs, and Feature #7 to have high values for non-LVCs. These features are similar to the ones used in Salehi et al. [2011] previous work for the automatic identification of English Verb Particle Constructions

3 Experimental Setup

3.1 Candidate Expressions

Persian LVCs are formed from a limited number of basic verbs, such as *kardan* (‘to make/do’) and *dâdan* (‘to give’), in combination with a noun, an adjective, or a prepositional phrase that precedes the verb.² In this paper, we focus on LVCs of the form noun+verb since according to Doostan and Sanandaj [2001] most of the Persian LVCs are in the form of Noun+Verb. In particular, we focus on LVCs formed from five of

² Persian is an SOV (Subject-Object-Verb) language, hence the co-verbal element of an LVC precedes the verb.

Table 1. Selected basic verbs, and LVC examples

Basic Verb	Meaning	Example LVC	Meaning
<i>zadan</i>	‘to hit’	<i>jâru</i> (‘broom’) + <i>zadan</i>	‘to sweep’
<i>khordan</i>	‘to eat’	<i>shekast</i> (‘failure’) + <i>khordan</i>	‘to fail’
<i>dâdan</i>	‘to give’	<i>tekân</i> (‘motion’) + <i>dâdan</i>	‘to shake’
<i>gozâshtan</i>	‘to put’	<i>ehterâm</i> (‘respect’) + <i>gozâshtan</i>	‘to respect’
<i>gereftan</i>	‘to get’	<i>âtash</i> (‘fire’) + <i>gereftan</i>	‘to burn’

the most common basic verbs in Persian. Table 1 presents these verbs along with an example LVC for each, and their translation in English.³

As our candidate expressions, we first extract each occurrence of any of our selected five verbs along with its preceding word (i.e., a bigram whose second component is a basic verb from our set of five verbs). This is a noisy extraction method, and as a result the first list of bigrams contains a lot of noise, including misspellings. We manually eliminate 98 misspellings (about 4% of the original list), such as *tegân khordan* as a misspelled variant of *tekân khordan*.⁴ The final list of candidates has 2371 expressions, of which 692 are LVCs.

Table 2. Information on the division of our candidate expressions into frequency groups (LOW, MED, and HIGH), evaluation sets (TRAIN, DEV, and TEST), and classes (LVC, NON-LVC)

Data	LOW		MED		HIGH	
	LVC	NON-LVC	LVC	NON-LVC	LVC	NON-LVC
TRAIN	226 (22%)	812 (78%)	144 (48%)	159 (52%)	45 (64%)	25 (36%)
DEV	67 (19%)	283 (81%)	50 (49%)	53 (51%)	15 (60%)	10 (40%)
TEST	74 (22%)	276 (78%)	54 (51%)	51 (49%)	17(63%)	10 (37%)

For evaluation purposes, we divide our candidate expressions into three frequency ranges, i.e., LOW (frequency < 4), MED ($4 \leq$ frequency < 30), and HIGH (frequency \geq 30). Each frequency range group is then divided into three sections for training (TRAIN), development (DEV) and test (TEST). TRAIN includes 60% randomly selected candidates, and the two other parts each contain half of the remaining candidates (20% of the total). Table 2 provides information on the division of our candidate expressions into frequency groups (LOW, MED, and HIGH), evaluation sets (TRAIN, DEV, and TEST), and classes (LVC, NON-LVC).

3.2 Annotation

We asked two native speakers of Persian with sufficient linguistic background to annotate our candidate expressions. The human judges were asked to assign the label 1 to

³ We select these verbs from relevant linguistic studies. Two common verbs are not included: *kardan* (‘to make/do’) and *keshidan* (‘to pull’) because some inflected forms of them are homomorphic with other verbs in written Persian, and *kardan* almost exclusively appears in LVCs.

⁴ Note that the written forms of *g* and *k* are very similar in Persian, and hence such a misspelling is common.

Table 3. AUC for different k in k NN

	$k=1$	$k=3$	$k=5$
LOW	0.62	0.64	0.61
MED	0.71	0.75	0.73
HIGH	0.75	0.78	0.76

LVCs, and the label 0 to all other (non-LVC) combinations (including literal combinations and noise). The observed agreement between our annotators was 95%, and the κ score was 92%. Specifically, the annotators did not agree on the label for 122 candidates (5% of the candidates). We asked a third judge to annotate these expressions, and used her annotations for these 122 items.⁵

3.3 Corpus

We used one of the few freely available Persian corpora, the Tehran English–Persian (TEP) corpus [Pilevar and Faili, 2010].⁶ TEP contains about 600,000 parallel sentences (around 4 million words on each side) extracted from movie subtitles (the text genre is thus conversational). To build word alignments required by our features, we used Giza++ [Och and Ney, 2000].

3.4 Evaluation

Our task is to classify our candidate expressions into two classes: LVC and NON-LVC. We use and compare two classifiers: the k -nearest neighbor (k NN) classifier, and the multilayer perceptron (MLP) classifier. For k NN, we use MATLAB 2006, and for MLP we use Weka 3.6.⁷

4 Results

We first perform experiments on the DEV expressions to find the best k for the k NN classifier. For this, we use the Receiver Operating Characteristic Curve (ROC)⁸, and compute the Area under Curve (AUC)⁹ for $k = 1, 3, 5$, for the LOW, MED, and HIGH expressions in DEV (see Table 3). $k = 3$ produces the best performance, so we set k to 3 in all our experiments. We report the results on TEST, but results on DEV had similar trends.

Table 4 compares the accuracy of our two classifiers (MLP and 3NN) with that of a simple majority baseline — one that assigns the label of the majority class in each frequency group (LOW, MED, and HIGH) within TEST to all members of that set.

⁵ Contact the first author to get our dataset.

⁶ <http://ece.ut.ac.ir/nlp/resources.html>

⁷ <http://www.cs.waikato.ac.nz/ml/weka/>

⁸ ROC plots the true positive vs. the false positive rates.

⁹ For more information on AUC and its desirable properties as a classification performance measure when compared to overall accuracy check [Bradley 1997].

Table 4. %Acc on TEST expressions, for the baseline, and for the two classifiers (MLP and 3NN)

		Majority		
Data	Label	Baseline	MLP	3NN
LOW	(NON-LVC)	78	80.6	76
MED	(NON-LVC)	52	78.1	74.3
HIGH	(LVC)	64	85.2	85.2

Table 5. TEST Performance (R , P , and F) of 3NN and MLP for the LVC and NON-LVC classes**3NN Classifier:**

Data	LVC			NON-LVC		
	R	P	F	R	P	F
LOW	33.8	41.7	37.3	87.3	83.1	85.2
MED	85.2	70.8	77.3	62.8	80	70.3
HIGH	88.2	88.2	88.2	80	80	80

MLP Classifier:

Data	LVC			NON-LVC		
	R	P	F	R	P	F
LOW	16.2	66.7	26.1	97.8	81.3	88.8
MED	94.4	71.8	81.6	60.8	91.2	72.9
HIGH	82.4	93.3	87.5	90.0	75.0	81.8

As the results show, both classifiers outperform the baseline in most cases (shown in boldface): MLP outperforms the baseline in all frequency ranges; 3NN is worse than the baseline only on LOW frequency TEST expressions. Our initial experiments using other classifiers, such as SVM and decision tree did not produce satisfactory results

Table 5 shows the Recall, Precision, and F -score of the 3NN and MLP classifiers on TEST expressions, for the two classes (LVC and NON-LVC). For each frequency group (LOW, MED, and HIGH), the highest F of the two classifications is shown in boldface. These results show that MLP performs better than or comparable to 3NN on five (out of six) cases.

Both classifiers show a very poor performance on on LOW-frequency LVCs (see top and bottom panels in Table 5, first row, left half under LVC). Note that 3NN has a slightly better performance than MLP in identifying LOW-frequency LVCs (an F -score of 37.3 versus 26.1).¹⁰ This poor performance may be partially attributed to the higher proportion of NON-LVCs (78%) in this data set, of which many are likely to be erroneous combinations or noise. Of the seven features we use, only one (Feature #5) is designed to weed out noise. In future, we will need to include more features of this type.

We now focus on the performance of the classifiers on the MED and HIGH-frequency expressions, particularly examining the differences in performance across the two

¹⁰ This result is not in contrast with the overall accuracy of 3NN on LOW expressions in Table 4. Here, too, we can see that 3NN does not perform as well as MLP on NON-LVCs, which represent the majority of items in the LOW frequency group.

classes. In all four cases (two classifiers by two frequency groups), the *F*-score is higher for the LVC class (in three of these cases, *Recall* is higher for LVC; in two cases *Precision* is higher for LVC). This is expected, since most our features are designed so that they capture common properties of LVCs. In addition, our annotation is also separating LVCs from everything else (lumped into the same class, called NON-LVC). It is very possible that our class of NON-LVC expressions is more heterogeneous than the LVCs; the former may contain noise, literal phrases, or even abstract collocations. Thus, a finer-grained distinction among these various semantic classes might be more appropriate [Fazly and Stevenson, 2007].

5 Conclusions

To our knowledge, this is the first study focusing on the automatic extraction of a broad class of Persian light verb constructions (LVCs). Persian is an under-resourced language, and hence we have opted to take advantage of a bilingual corpus. We have thus used and extended several translation-based features that have been introduced for the identification of multiword expressions in other languages. In addition, we have proposed new features that tap into properties of Persian LVCs.

Our classification results (using two different classifiers) show that the features are in general relevant to the task of Persian LVC identification. However, both classifiers had difficulty in identifying low-frequency LVCs. Future work will need to look more closely into incorporating more linguistically-informed features that do not require large amounts of data for accurate estimation. In addition, we intend to examine the effect of making finer-grained semantic distinctions among the expressions, instead of a binary classification into LVC and non-LVC. Other future research directions include examining the effect of each individual feature separately, as well as investigating unsupervised or minimally-supervised techniques for the task. The features we use in this study are mostly language independent. In future, we intend to include features that draw on the properties of Persian LVCs.

References

- Baldwin, T., Villavicencio, A.: Extracting the unextractable: A case study on verb-particles. In: CoNLL 2002 (2002)
- Bradley, A.: The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7) (1997)
- Brown, P., Della Pietra, V., Della Pietra, S., Mercer, R.: The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19(2) (1993)
- Butt, M.: The light verb jungle. In: Workshop on Multi-Verb Constructions (2003)
- Dabir-Moghaddam, M.: Compound verbs in Persian. *Studies in the Linguistic Sciences* 27(2) (1997)
- Doostan, G.K., Sanandaj, I.: N+ v complex predicates in persian. *Structural Aspects of Semantically Complex Verbs*, 277–292 (2001)
- Evert, S., Krenn, B.: Using small random samples for the manual evaluation of statistical association measures. *Computer Speech & Lang.* 19 (2005)

- Fazly, A.: Automatic acquisition of lexical knowledge about multiword predicates. PhD thesis, University of Toronto (2007)
- Fazly, A., Stevenson, S.: Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. In: ACL Wkshp. on MWEs (2007)
- Karimi, S.: Persian complex verbs: Idiomatic or compositional. *Lexicology* 3(1) (1997)
- Karimi-Doostan, G.: Light verb constructions in Persian. PhD thesis, University of Essex (1997)
- Karimi-Doostan, G.: Light verbs and structural case. *Lingua* 115(12) (2005)
- Khanlari, P.: *Tarikh-e zaban-e farsi* [history of the Persian language]. Bonyâd-e Farhang (1973)
- Megerdooimian, K.: A semantic template for light verb constructions. In: 1st Wkshp. on Persian Language and Computers (2004)
- Melamed, I.D.: Measuring semantic entropy. In: ACL-SIGLEX Workshop Tagging Text with Lexical Semantics: Why, What, and How (1997)
- Och, F.J., Ney, H.: Improved statistical alignment models. In: ACL 2000 (2000)
- Pilevar, M.T., Faili, H.: Presenting an automatic movie subtitle translation system. In: 15th Conf. of the Computer Society of Iran (2010) (in Persian)
- Salehi, B., Fazly, A., Jahromi, M.: Extracting non-compositional verb particle constructions using a bilingual corpus. In: Signal and Image Processing and Applications (2011)
- Stevenson, S., Fazly, A., North, R.: Statistical measures of the semi-productivity of light verb constructions. In: ACL Wkshp. on MWEs (2004)
- Villada Moirón, B., Tiedemann, J.: Identifying idiomatic expressions using automatic word alignment. In: EACL Wkshp. on MWEs (2006)
- Villavicencio, A., Kordoni, V., Zhang, Y., Idiart, M., Ramisch, C.: Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In: EMNLP 2007 (2007)

A Cognitive Approach to Word Sense Disambiguation

Sudakshina Dutta and Anupam Basu

Indian Institute of Technology Kharagpur
{sudakshina, anupam}@cse.iitkgp.ernet.in

Abstract. An unsupervised, knowledge-based, parametric approach to Word Sense Disambiguation is proposed based on the well-known cognitive architecture ACT-R. In this work, the target word is disambiguated based on surrounding context words using an accumulator model of memory search and it is realized by incorporating RACE/A with ACT-R 6.0. In this process, a spreading activation network is built following the strategies of Tsatsaronis et al. proposed in [5] using the chunks and their relations in the declarative memory system of ACT-R and the lexical representation has been achieved by integrating WordNet with the cognitive architecture. The resulting Word Sense Disambiguation system is evaluated using the test data set from English Lexical Sample task of Senseval-2 and overall accuracy of the proposed algorithm is 44.74% which outperforms all the participating Word Sense Disambiguation Systems.

Keywords: WSD, Word Sense Disambiguation, RACE/A, Retrieval by Accumulating Evidence in an Architecture, ACT-R, Adaptive Control of Thought-Rational, SAN, Spreading Activation Network.

1 Introduction

Word Sense Disambiguation (WSD) is the process of automatically assigning the correct meaning of a word based on the surrounding context. While the human reader rarely encounters any problem in finding actual meaning of a word in a given context, a computer program faces difficulties as it has neither the knowledge nor the exposure to the different forms of expressions in languages. Research works have been steadily progressed in various directions and various supervised, unsupervised, knowledge-based, data-based algorithms have been proposed for WSD. A few of the WSD algorithms have utilized the concept of spreading activation network theory which was introduced by Quillian in [17], [18] to disambiguate a word based on the surrounding context words. We present a brief overview of the different WSD algorithms and approaches in the next section. In this paper, we have adopted a cognitive approach to address the WSD problem, modeling the cognitive process through the cognitive architecture ACT-R 6.0. An accumulator model of memory search of RACE/A for retrieving facts from declarative memory is utilized for retrieving correct sense

of the target word from the memory system of the cognitive architecture. The cognitive agent uses the spreading activation strategy to simulate the influence of the current context on the retrieval procedure. Maanen et al. [13] found the ballistic model of declarative memory retrieval of ACT-R insufficient to provide accurate prediction of the retrieval procedure of human cognitive system. They proposed a more fine-grained retrieval model of RACE/A to provide a realistic simulation of retrieval procedure of a human cognitive system. In RACE/A, the mental search procedure is described as the accumulation of activation from the previously presented facts to the cognitive system. In this work, the accumulator model of WSD system for disambiguating a word in the presence of surrounding context words using the spread of activation among the associated concepts is realized using ACT-R augmented with RACE/A. WordNet is used to represent the lexicon in the declarative memory of ACT-R. The synset hierarchies of WordNet are mapped to the chunks and the relations among different synsets are mapped to the associative relations between the chunks in the declarative memory of the ACT-R cognitive system. The spreading activation network has been built in the declarative memory using WordNet following the strategies of Tsatsaronis et al. [5]. The proposed WSD algorithm supports the local processing assumption of extended spreading activation theory proposed in [1] although it does not fully obey global assumption about memory and processing. The system is evaluated using English Lexical Sample Task of Senseval-2 and it outperforms all the seven unsupervised approaches which participated in the workshop. The rest of the paper is organized as follows. Section 2 contains the overview of the various related literature. The accumulator model of WSD realized using ACT-R and RACE/A has been described in Section 3, while the implementation of the algorithm has been discussed in Section 4. Section 5 presents the evaluation and the results. Section 6 concludes the paper with some pointers to future work.

2 Literature Survey

2.1 Word Sense Disambiguation Methods

Research work on Word Sense Disambiguation started in the 1940's and a wide range of Word Sense Disambiguation algorithms have been proposed over the years. Some of them follow the supervised approach in which labelled training set is utilized, some of the algorithms follow unsupervised approach which attempts to disambiguate a word without previous training or labelled corpora. In data-based approach, the system only utilizes lexical information conveyed in the passage, but in knowledge-based approach, the algorithm uses the underlying meaning of the text to disambiguate a word. In [23], [26], the authors have published a survey of different approaches of Word Sense Disambiguation systems. In late 1950's, semantic networks were developed to represent the meanings of the words in a sentence. Quillian [16], [17], [18], [19], [20] made semantic networks by hand incorporating dictionary definition of words and used the semantic network for disambiguating a word in a text. As stated in [23], the works in psycholinguistics in 1960's and 1970's established that semantic priming plays

a key role in the disambiguation of senses of a word in human natural language processing system. This concept was applied in simulating a spreading activation model by Anderson in [8], [9]. The idea of human semantic memory of [17], [18] is realized in the spreading activation model of [1] where the activation propagates through concepts and it is gradually weakened as it propagates. The spreading activation model is enhanced with introduction of inhibition among the senses of the nodes proposed in [22]. Lesk [15] proposed an unsupervised, data-based algorithm for disambiguation of word in small phrases. The definition of each sense of a word is compared with the definitions of all the other words in the phrase and the sense with highest overlap with the definitions of the context words is selected as the correct sense of the word in that context. In [28], the author used WordNet in Lesk's algorithm and altered the basic approach of the algorithm. While Lesk's algorithm only compares the glosses of the word to be disambiguated, this algorithm compares the glosses of the related words of that word. But in the WSD system of [28], all the relations of WordNet were not considered. In [12], the author used spreading activation network in their WSD system. But their edge-weighting scheme could not represent the importance of the edge in the spreading activation network. In [29], the author presented an approach based on the use of PageRank algorithm to study the nature of mental lexicon and identify the senses which are more relevant in the context. The method builds a graph to represent all the senses of words in a text and connects pairs of senses with meaningful relations. Tsatsaronis et al. [5] proposed a new method of constructing the spreading activation network and adopted the activation strategy introduced in [7]. For edge weighting, they adopted commonly used tf-idf (term frequency-inverse document frequency) weighting scheme for assigning weights to the edges of the spreading activation network.

In contrast to most of the methods discussed above, we propose a cognitive approach to Word Sense Disambiguation using a well-known cognitive architecture. In the proposed work, the network of synsets and their relations in WordNet are mapped to chunks and their relations in the declarative memory of ACT-R. The methods for construction of the spreading activation network and the edge-weighting scheme proposed in [5] are used to form the network of chunks and associate strength among them in the declarative memory of ACT-R. The proposed word sense disambiguation system is modeled as an accumulator which accumulates until one sense associated with the given word reaches a threshold value and is retrieved. The overall system partially represents the extended spreading activation theory of human semantic processing system of [1].

2.2 WordNet

WordNet, proposed in [6], is a lexical database where nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets). The synsets are connected to each other by different relations like hypernymy/hyponymy, meronymy/holonymy, entailment/causality, attribute terms, similar terms etc. It provides short general definitions for every synsets present in it. In this

implementation, we have used LISP implementation of WordNet which is developed in [2] and it is made from the prolog version of WordNet 2.0.

2.3 ACT-R

Adaptive Control of Thought-rational, proposed in [10], is a cognitive architecture, in which multiple modules are integrated to produce a cognitive architecture which mimics human cognitive system. The chunks in ACT-R represent basic units of information about the outside world and the modules of ACT-R provide the perception and reaction mechanism of the cognitive architecture. There is a central production system, which governs the actions of ACT-R in a particular situation and it interfaces the modules with the buffers. The chunks are stored in the declarative memory. If number of chunks match with the retrieval request, the chunk with highest activation is retrieved from the declarative memory. The activation of a chunk in ACT-R is represented by three components i.e. base-level activation, spreading activation and noise as given in (1).

$$A_i = B_i + \sum_k \sum_j W_{kj} * S_{ji} + \sigma . \quad (1)$$

Here B_i implies the Base Level Activation of the chunk i and represents the likelihood that the chunk can be needed in the near future. The second term is the spreading activation component which signifies the effect of the perceived fact j in the buffer k on the activation of the chunk i . Here W_{kj} is the amount of activation of source j in buffer k and S_{ji} is the strength of association between two chunks j and i . The strength of association between two chunks j and i is assigned using the following equation.

$$S_{ji} = S - \ln(fan_j) . \quad (2)$$

Here S implies maximum associative strength between chunks and fan_j denotes the fan value of the j^{th} chunk in the declarative memory. σ implies noise which is composed of permanent noise associated with each chunk and instantaneous noise computed at the time of a retrieval request. In ACT-R, the retrieval time and the probability of retrieving a chunk which matches the retrieval request are fixed and it is known at the time when the retrieval procedure starts. The retrieval model is referred as ballistic model where the probability of a chunk to be retrieved and the retrieval time required for the chunk are deterministic in nature.

2.4 RACE-A

One of the drawbacks of the retrieval procedure of ACT-R is that it does not provide theory of actual memory retrieval process of human cognitive system. Rather, the ballistic model provides a prediction of retrieval time and probabilities under normal condition. Moreover, in ACT-R 6.0, the chunks in the

buffer spread activation to the chunks in the declarative memory. From (1), it is clearly understood that a chunk i of declarative memory can only receive activation from the chunk residing in the buffers which prevents the propagation of activation among the chunks in the declarative memory of ACT-R 6.0. Hence, in order to simulate the human retrieval process, ACT-R requires some augmentation, which could be conveniently achieved by integrating RACE/A (Retrieval by Accumulating Evidence in an Architecture) with it. In RACE/A, the effect of asynchronous information in the mental search procedure can be modeled. In this architecture, the memory retrieval is modeled using sequential sampling model proposed in [27]. If there is more than one mental representation present in declarative memory, then certain mental representation is discriminated and retrieved if the accumulated activation exceeds a pre-specified boundary. A starting point of accumulation (z), mean drift rate (v) and the match boundary (a or b) specify the parameters of classical diffusion model where the accumulation process starts from the starting point z , continues with drift rate v and reaches match boundary a or b depending on the surrounding evidences. If the starting point is closer to a than b , then the required accumulation to reach a is less than b . One of the drawbacks of the classical diffusion model is that it only accounts for two response options. The other sequential sampling models e.g. the authors of [22] proposed that each memory representation as separate accumulators. However, these models fail to appreciate that retrieval of a fact from declarative memory does not stand alone. RACE/A reconciles the above approaches and in this architecture, the accumulation process is characterized by two equations. The long-term dynamics is expressed by the default base-level activation equation (1) and the short-term dynamics is mediated by spreading activation from the memory-resident chunks using (3). In RACE/A, the spreading activation component of (2) is replaced with the Accumulated Activation component C_i at time instant t which is specified using the following equation.

$$C_i(t) = \alpha * C_i(t-1) + \beta * \sum_j S_{ji} * C_j(t-1) \quad . \quad (3)$$

Here α signifies the decay in the accumulated activation component and it can take value from $[0,1]$. β is the scaling factor which determines the overall accumulation speed. If more than one chunk matches the retrieval criteria, then all the chunks accumulate activation from the other chunks (or from external stimuli) and compete among each other during the entire retrieval time period until any one of them reaches decision boundary. In this process, the chunks of declarative memory spread activation among each other until any one of the competing chunks reach the boundary value. In RACE/A the entire retrieval time period is time sliced based on the update frequency and for each of the time instant t , the accumulated activation of a chunk is updated using (3). During the retrieval time interval, the activations of all the chunks are updated until any one of the competing chunks reaches decision boundary θ . If the activation of i^{th} chunk is A_i and the activation of the competing j^{th} chunk is A_j , then the condition for stopping the accumulation is given below.

$$\frac{e^{A_i}}{\sum_j e^{A_j}} \geq \theta . \quad (4)$$

3 Word Sense Disambiguation system as an Accumulator

In real life, disambiguation of a polysemous word from the discourse context is specified by finding the meaning of the word based on the words surrounding the target words. In the present work, we attempt to model this phenomenon by modelling the cognitive process through ACT-R 6.0 augmented with RACE/A. In RACE/A, the declarative memory search procedure can be modelled as accumulation of activation and the process continues until activation of some fact crosses retrieval threshold. In this work, word sense disambiguation system is implemented as a searching procedure which produces the correct sense of the target word as soon as the activation of some sense exceeds retrieval threshold. The mental lexicon in the long term memory is modeled by integrating WordNet with ACT-R 6.0. In the proposed system, the target word can refer to multiple senses in the WordNet. All the senses (synsets) of the target word along with the senses of the surrounding context words are added to the declarative memory as chunks until the entire network of synsets is connected and forms the spreading activation network in the declarative memory. The activations of the synset chunks of the surrounding context words are increased as they are referred and it is propagated through the spreading activation network. All the competing senses of the target word accumulate activation from the senses of the surrounding words using (3) until any one sense reaches decision boundary as given in (4). As soon as at least one sense of the target word reaches decision boundary, the accumulation procedure is stopped and retrieval procedure starts. The accumulator model can also explain how the latency of finding the correct sense of the target word depends on the nature of the context words. If the target and the context words refer to the same synset, then retrieval of the sense of the target word results in no competition and retrieval occurs immediately as both of them activate the same synset. If the target word and the context word are semantically related, then ratio of (4)(Luce Ratio) becomes lower resulting in competition among the senses and accumulator model takes more time to exceed the decision boundary. If the context and the target word are not related at all, then the retrieval operation does not get affected by the presence of the surrounding words. The above cases correspond to the congruent, related and incongruent condition as explained in [13]. In Fig. 1, the spreading activation process is shown with a context word and the target word. The propagation of activation from the senses of the context word to the senses of the target word is shown with bold line.

4 Implementation

In this section we present the different stages of implementation of the algorithm on ACT-R 6.0.

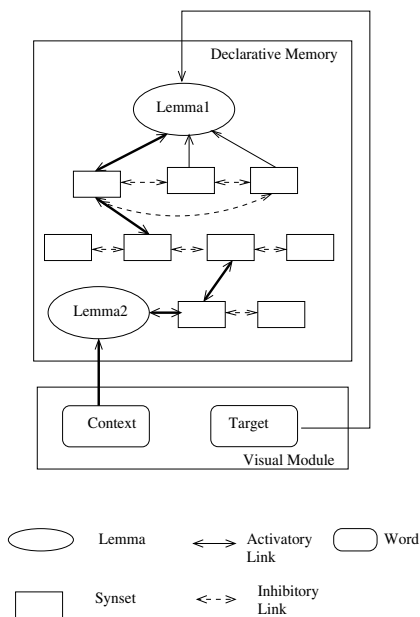


Fig. 1. An example network with one context word and target word. After being reduced to lemmas they started participating in Spreading Activation process.

4.1 Spreading Activation Network Creation

The synsets of the WordNet are mapped to the synset chunks added to the declarative memory and the relations between the synsets are added as the associative relations between the chunks. While executing a single iteration of Word Sense Disambiguation algorithm, the target word and the surrounding context words are visually presented to the cognitive system and are reduced to the base forms (lemmas). The input contains words from n surrounding sentences for which number of WordNet words exceed ten. An example from English Lexical Sample Task can be considered.

Starting from the ancient times, all such studies have influenced every form of art in some way.

In the above example, the words start, ancient, time, all, such, study, influence, form, art, in, some, way can be added as context words for disambiguating the word art. In the 1st iteration of Spreading Activation Network(SAN) construction, all the senses (synsets) of the context and target words are added as chunks to the declarative memory along with the lemmas. The competing synset chunks for a word are mutually associated with negative weighted edge (inhibitory edge). Figure 1 shows the SAN construction method of the word start given in the above example. In iteration 2, all the synset chunks that are connected to the already added chunks using hypernymy/hyponymy, meronymy/holonymy, entailment/causality, synonymy/antonymy, attribute, derivation and similarity

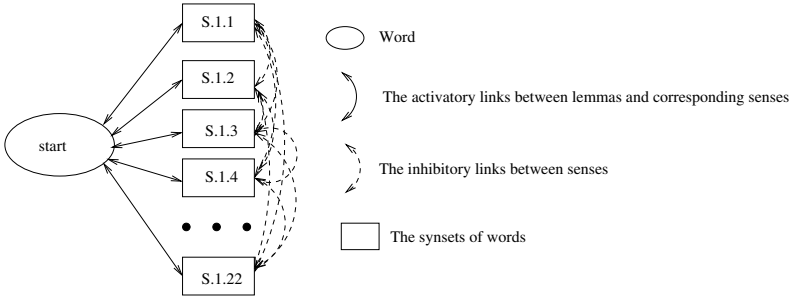


Fig. 2. Iteration 1 of the SAN construction method for the word *start*

relations are added to the declarative memory. The connection between the chunks is both sided and it may cross parts of speech also. This process continues until the network of synset chunks are connected or fifth iteration is over. The value five is chosen as the propagated activation value of the sense reaches a small value (first three digits after decimal point are insignificant) with the parameter settings of this algorithm. This is a major variation from the algorithm of Tsatsaronis et al. proposed in [5] where the disambiguation procedure is aborted if the network is not connected. This variation is incorporated so that the algorithm does not fail to disambiguate even in the case of a sparsely connected network.

4.2 Edge Weight Assignment

In ACT-R 6.0, the strength of association between two chunks j and i is measured using (2) where the fan-out factors of different slots of the chunks are measured. In the current implementation, the strength of association between two chunks is measured in a different way using tf-idf weighting strategy to find the importance of different association relationship i.e. hyponymy, meronymy etc. among the chunks in the entire network. Each of the links (association) between sense chunks is assigned weights proportional to the importance of the link with respect to the entire network. The weighting scheme is taken from [5] for assigning weights to the edges of the SAN. Initially each of the associations (S_{ji}) is assigned +1 if association is activatory and the -1 if it is an inhibitory association. An association between two chunks is inhibitory if the relation is antonymy or the chunks are competing senses of a word. All other associations between the chunks are considered as activatory association between the chunks. After SAN construction, the initial weight assigned on the association between synset chunk j and i is multiplied as follows.

$$S_{ji} = initialweight * ETF(e_{ji}) * INF(e_{ji}) \quad . \quad (5)$$

ETF (Edge Type Frequency) signifies percentage of the outgoing associative relations of a synset chunk, which are of same type as e_{ji} whereas INF(Inverse

Node Frequency) implies fraction of synset chunks in the network, which has at least one outgoing association is of type e_{ji} . The value of ETF and INF are computed using the equations given below.

$$ETF(e_{ji}) = \frac{|\{e_{jm} | type(e_{jm}) = type(e_{ji})\}|}{|\{e_{jm}\}|} . \tag{6}$$

$$INF(e_{ji}) = \log \frac{N + 1}{N_{type}(e_{ji})} . \tag{7}$$

Here N is the total number of synset chunk in the SAN and $N_{type}(e_{ji})$ is the number of synset chunks for which at least one outgoing association is of same type (type of any association can similar to any WordNet relation like hypernymy, meronymy etc.) as e_{ji} .

4.3 Implementation of RACE/A

According to Manen [14], the accumulated activation component of a chunk in RACE/A is represented by starting point or initial activation value of the chunk in the declarative memory, drift equation (3), with which accumulated activation is updated and the decision boundary. The starting point of accumulation is specified using base-level activation of the chunk. If a number of chunks match retrieval request, then all the chunks accumulate activation using the drift equation until at least one of the competing chunks exceeds decision boundary.

In our implementation, all the synset chunks of the surrounding context words are assigned the base-level action value 1.0 as the starting point of accumulation. The synset chunks of the target word are assigned base-level activation value 0.0. The synset chunks associated with the surrounding words propagate activation to the other chunks of the declarative memory during the entire retrieval time period using drift equation. The accumulation procedure stops if any one of the competing chunks of the target word exceeds decision boundary. In this implementation, we have used $1/n$ as the value of decision boundary where n is the number of competing chunks in the declarative memory. Equation 4 is modified and is used in the following form in the current implementation.

$$\frac{e^{A_i}}{\sum_j e^{A_j}} > \frac{1}{n} . \tag{8}$$

The ratio in the left hand side of (8) is called Luce Ratio. The explanation of using $1/n$ as decision boundary is given below. Initially, the competing synset chunks are assigned activation values 0.0. So the Luce Ratio of any competing chunk is $1/n$. As the accumulation process proceeds, some of the competing chunks receive positive activation and some other chunks receive negative or no activation from other chunks and external stimuli. So the denominator of (8) can increase or decrease or can remain at the same value subsequently. As e^z is a monotonically increasing function, the value of e^z will always be more than e^0 , if z is a positive quantity. So, as some of the competing chunks accumulate

positive activation the numerator of (8) increases. At first, suppose $x/y = 1/n$ where $x = e^{A_i}$ and $y = \sum_j e^{A_j}$. After some competing chunk accumulate positive activation, x becomes $x + m$ where m is a positive quantity and y becomes y_1 .

$$y_1 < y \quad y_1 = y - p_1 \text{ where } p_1 > 0$$

$$\frac{x+m}{y-p_1} - \frac{x}{y} = \frac{my+p_1x}{(y-p_1)y} > 0$$

$$y_1 = y \quad \frac{x+m}{y} - \frac{x}{y} = \frac{m}{y} > 0$$

$y_1 > y$ For the maximum activated node

$$y_1 < (x+m)n \text{ and } p_2 = (x+m)n - y_1 > 0$$

$$\frac{x+m}{y_1 t_1} - \frac{x}{y} = \frac{(x+m)}{((x+m)n-p_2)} - \frac{x}{y}$$

$$\frac{t_1}{(t_1 n - p_2)} - \frac{1}{n} = \frac{p_2}{(t_1 n - p_2)n} > 0, \text{ where } t_1 = (x+m)$$

If all the chunks receive negative activation, then the numerator for the maximum negatively activated node becomes $x - m$, where m is a positive quantity and following factors are observed.

$$y_1 > (x-m)n, p_3 = y_1 - (x-m)n > 0$$

$$\frac{x-m}{p_3+(x-m)n} - \frac{1}{n} = \frac{-p_3}{(p_3+(x-m)n)n} > 0$$

So if some of the chunk receives positive activation, then accumulation process stops as decision boundary is exceeded. If all the chunks receive negative activation, then the Luce Ratio of the maximum negatively activated chunk exceeds the decision boundary. After the decision boundary is reached, the maximum activated chunk will be retrieved from declarative memory. If none of the chunk exceeds decision boundary, the retrieval time window expires and the maximum activated chunk will be selected for the retrieval process.

4.4 Word Sense Disambiguation Algorithm

The proposed Word Sense Disambiguation algorithm consists of following steps.
 Step 1: The input sentences are stemmed. The number of sentences should be chosen such that the number of input words surrounding the target word is at least ten. The word to be disambiguated should be marked so that the cognitive system distinguishes it separately.

Step 2: The SAN created using the method described in section 4.1.

Step 3: All the associations (S_{ji}) between the sense chunks are bi-directional and are assigned weights using the methods discussed in section 4.2.

Step 4: If the target word is not a polysemous word and it contains only one sense, then the sense is retrieved immediately. Otherwise, the following steps are performed.

Step 5: All the synset chunks connected to the surrounding context words and the target word are assigned base-level activation value 1.0 and 0.0 respectively. The accumulated-activation for each sense chunk also receives the same value as base-level activation value.

Step 6: The retrieval latency is computed using (9) given below.

$$RT = Fe^{-\tau} \quad . \quad (9)$$

Table 1. Parameter values used in the algorithm

Parameters	Values
α	0.255
β	0.720
Update-frequency	200
F	200
τ	0.0

Here F is the scaling parameter and τ signifies the retrieval threshold. The entire retrieval latency is divided into m intervals where m is the update frequency parameter. In each of the time instant, the accumulated activation of all the chunks of the declarative memory is updated using (3).

Step 8: The accumulation procedure continues until some of the competing synset chunks of the target word exceed decision boundary. If more than one chunk exceeds the boundary, then the synset chunk with maximum activation value is retrieved. The cognitive system announces retrieval failure if none of the chunks exceeds the decision boundary within the retrieval time interval.

Step 9: If the retrieval procedure succeeds, the gloss of the retrieved sense is found from the WordNet and it is reported to the user.

The parameter values used in the implementation are presented in Table 1. The parameters can be tuned for better performance.

5 Evaluation and Results

The proposed algorithm is evaluated using English Lexical Sample Task of Senseval-2. The target words are disambiguated based on the surrounding context words using this dataset. There are a total of 4328 test instances divided among 29 nouns, 29 verbs and 15 adjectives. In the following results, the words followed by the accuracy obtained by our algorithm are listed. The accuracy is measured by dividing the number of correct disambiguated instance by the total number of test instances for a word and is expressed as a fraction. The result of the proposed WSD system is evaluated using the probabilistic scoring mechanism proposed by Melamed and Resnik in [3].

Noun : (art, 0.47), (authority, 0.39), (Bar, 0.27), (Chair, 0.51), (Bum, 0.38), (Child, 0.56), (Holiday, 0.60), (Church, 0.32), (Facility, 0.49), (Lady, 0.77), (Nation, 0.43), (Detention, 0.72), (Fatigue, 0.49), (Spade, 0.55), (Yew, 0.75), (Circuit, 0.32), (Day, 0.37), (Dyke, 0.50), (Feeling, 0.37), (Hearth, 0.50), (Grip, 0.39), (Material, 0.48), (Mouth, 0.40), (Channel, 0.41), (Sense, 0.11), (Post, 0.42), (Restraint, 0.25), (Stress, 0.46), (Nature, 0.44). Thus, the accuracy of nouns is **45.24%**.

Verb : (Begin, 0.59), (Collaborate, 1.00), (Call, 0.22), (Replace, 0.68), (Wander, 0.64), (Treat, 0.33), (Develop, 0.48), (Draw, 0.48), (Dress, 0.56), (Drift, 0.35), (Drive, 0.39), (Face, 0.29), (Find, 0.47), (Keep, 0.41), (Leave, 0.35), (Live, 0.54), (Match, 0.56), (Play, 0.39), (Pull, 0.43), (See, 0.40), (Serve, 0.45), (Strike, 0.35),

(Train, 0.52), (Turn, 0.27), (Use, 0.54), (Wash, 0.32), (Work, 0.47), (Ferret, 1.00), (Carry, 0.41). The accuracy of verbs is **47.90%**.

Adjective : (Blind, 0.30), (Colourless, 0.58), (Faithful, 0.32), (Local, 0.81), (Cool, 0.17), (Fine, 0.18), (Graceful, 0.41), (Fit, 0.30), (Free, 0.35), (Green, 0.57), (Oblique, 0.48), (Solemn, 0.48), (Natural, 0.33), (Vital, 0.37), (Simple, 0.51). The accuracy of adjectives is **41.07%**.

Thus the overall accuracy of the proposed WSD system is **44.74%** evaluated using the full dataset of English Lexical Sample Task of Senseval-2.

5.1 Analysis and Discussion

The proposed algorithm has achieved an accuracy of nearly 45% which outperforms many unsupervised Word Sense Disambiguation algorithm. In addition, our approach compares favorably with other unsupervised Word Sense Disambiguation systems that participated in Senseval-2. There are seven such systems participated in the workshop and the most accurate system reached an accuracy of 40%. The authors of [28] have proposed another unsupervised Word Sense Disambiguation algorithm which reached an accuracy of 32%. Moreover, the algorithm has not encountered any situation where the word cannot be disambiguated. Even for a disconnected network of synsets, the algorithm disambiguates the senses of the word. So there is practically no fail condition of the proposed Word Sense Disambiguation system. In English All Words task, the authors of [29] have reached an accuracy of 50.89% and the authors of [5] have achieved an overall accuracy of 46%.

The idea of human semantic memory proposed in [17], [18] was realized in the spreading activation model proposed by Collins et al. in [1] where the activation propagates through concepts and it is gradually weakened as it propagates. During the process, some of the senses receive activation from several sources. The activation value crosses retrieval threshold and the concept is retrieved according to the Local Processing Assumption of the extended theory proposed in [1]. The spreading activation model is enhanced with introduction of inhibition among the senses of the nodes as given in [22]. The proposed work supports the spreading activation theory of human semantic processing suggested by the above works and it is implemented using ACT-R (augmented with RACE/A). But we have used WordNet instead of the conceptual network of [1] to build the spreading activation network in the declarative memory system of ACT-R. The property comparison strategy is also not utilized. Moreover, the system provides a fine-grained approximation to human retrieval procedure as proposed in [14]. Therefore, it can be said that the proposed system is a partial simulation of human semantic processing occurs during human WSD procedure. There are many ways in which the algorithm can be improved. The context window of the Word Sense Disambiguation system contains only words from the sentences for which the count of the number of WordNet words exceeds ten. If a larger context window is used, the algorithm is expected to work better. The algorithm can also work well with a better tuning of the parameters of RACE equations.

6 Conclusion

In this paper, a cognitive approach to Word Sense Disambiguation has been proposed. The revised ACT-R (incorporated with RACE/A) is used to form an accumulator model of Word Sense Disambiguation system, where a word is disambiguated depending on the context words. The algorithm is evaluated using the test dataset of English Lexical Sample Task of SENSEVAL-2 and it works better than many of the existing unsupervised WSD algorithms. The proposed work is a partial simulation of semantic processing occurs during human Word Sense Disambiguation system. The future work includes setting different parameters in RACE equations and finding out which parameter combination works best for Word Sense Disambiguation System.

References

1. Collins, A.M., Loftus, E.F.: A spreading activation theory of semantic processing. *Psychological Review* 82(6), 407–428 (1975)
2. Emond, B.: WN-LEXICAL: An ACT-R module built from the WordNet Lexical database. In: Proc. of the Seventh International Conference on Cognitive Modelling, Trieste, Italy, pp. 359–360 (2006)
3. Melamed, D., Resnik, P.: Melamed and Resnik's Proposal for Senseval scoring (October 8, 2011), <http://www.cse.unt.edu/rada/senseval/senseval3/scoring/scorescheme.txt>
4. Meyer, D.E., Schvaneveldt, R.E.: Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology* 90(2), 227–234 (1971)
5. Tsatsaronis, G., Vazirgiannis, M., Androutopoulos, I.: Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In: Proc. of IJCAI 2007, pp. 1725–1730 (2007)
6. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: WordNet: An online lexical database. *Int. J. Lexicograph* 3(4), 235–244 (1990)
7. Berger, H., Dittenbach, M., Markl, D.: An adaptive information retrieval system based on associative networks. In: Proc. of the 1st Asia-Pacific Conference on Conceptual Modelling, Dunedin, New Zealand, pp. 27–36 (2004)
8. Anderson, J.R.: *Language, Memory, and Thought*. Lawrence Erlbaum and Associates, Hillsdale (1976)
9. Anderson, J.R.: A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior* 22(3), 261–295 (1983)
10. Anderson, J.R., Lebiere, C.: *The atomic components of thought*. Erlbaum, Mahwah (1998)
11. Anderson, J., Bothell, D.: *Tutorials and Reference Manual of ACT-R* (2011), <http://act-r.psy.cmu.edu/actr6/>
12. Veronis, J., Ide, N.M.: Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In: Proc. of COLING 1990, Helsinki, Finland, pp. 389–394 (1990)
13. Van Maanen, L., Van Rijn, H.: RACE for retrieval: Competitive effects in memory retrieval. In: Proc. of 12th Annual ACT-R Workshop, 12th edn., Trieste (2005)

14. Van Maanen, L.: Context effects on memory retrieval: Theory and applications. PhD thesis, University of Groningen, Groningen (2009)
15. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In: Proc. of SIGDOC (1986)
16. Ross Quillian, M.: A design for an understanding machine. In: Communication Presented at the Colloquium Semantic Problems in Natural Language. Kings College, Cambridge University, Cambridge, United Kingdom (1961)
17. Ross Quillian, M.: A semantic coding technique for mechanical English paraphrasing. Internal memorandum of the Mechanical translation Group. Research Laboratory of Electronics, M. I. T (August 1962)
18. Ross Quillian, M.: Word concepts: A theory and simulation of some basic semantic capabilities. *The Journal of Behavioral Science* 12, 410–430 (1967)
19. Ross Quillian, M.: Semantic memory. In: Minsky, M. (ed.) *Semantic Information Processing*, pp. 227–270. MIT Press (1968)
20. Ross Quillian, M.: The teachable language comprehender: a simulation program and theory of language. *Communication of ACM* 12(8), 459–476 (1969)
21. Byrne, M.D., Anderson, J.R., Qin, Y., Bothell, D., Douglass, S.A., Lebiere, C.: An Integrated theory of the mind. *Psychological Review* 111(4), 1036–1060 (2004)
22. Usher, M., McClelland, J.L.: The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review* 108(3), 550–592 (2001)
23. Ide, N., Veronis, J.: Word Sense Disambiguation: the state of the art. *Computational Linguistics* 24(1), 1–40 (1998)
24. Edmonds, P.: Designing a task for SENSEVAL-2 (2000), <http://www.sle.sharp.co.uk/SENSEVAL2/archive/index.html>
25. Edmonds, P., Cotton, S.: SENSEVAL-2: Overview. In: Proc. of The Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2), Toulouse, pp. 1–6 (2001)
26. Navigli, R.: Word sense disambiguation: a Survey. *ACM Computing Surveys* 41(2), 1–69 (2009)
27. Ratcliff, R., Smith, P.L.: A comparison of sequential sampling models for two-choice reaction time. *Psychological Review* 111(2), 333–367 (2004)
28. Satanjeev, B., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In: Proc. of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, pp. 136–145 (2002)
29. Chklovski, T., Mihalcea, R., Pedersen, T., Puradare, A.: The Senseval-3 multilingual English-Hindi lexical sample task. In: Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, p. 58 (2004)

A graph-Based Approach to WSD Using Relevant Semantic Trees and N-Cliques Model

Yoan Gutiérrez¹, Sonia Vázquez², and Andrés Montoyo²

¹ Department of Informatics
University of Matanzas, Cuba

² Research Group of Language Processing and Information Systems
Department of Software and Computing Systems

University of Alicante, Spain
{yoan.gutierrez}@umcc.cu, @dlsi.ua.es,
{svazquez,montoyo}@dlsi.ua.es

Abstract. In this paper we propose a new graph-based approach to solve semantic ambiguity using a semantic net based on WordNet. Our proposal uses an adaptation of the Clique Partitioning Technique to extract sets of strongly related senses. For that, an initial graph is obtained from senses of WordNet combined with the information of several semantic categories from different resources: WordNet Domains, SUMO and WordNet Affect. In order to obtain the most relevant concepts in a sentence we use the Relevant Semantic Trees method. The evaluation of the results has been conducted using the test data set of Senseval-2 obtaining promising results.

Keywords: Word Sense Disambiguation, Graph-based, N-Cliques, WordNet.

1 Introduction and Motivation

In Natural Language Processing (NLP) one of the main problems is the ambiguity of words. This problem affects different tasks such as: Information Retrieval, Information Extraction, Question Answering, etc. In order to deal with this problem an intermediate task called Word Sense Disambiguation (WSD) has been developed. WSD consists on determining the correct meanings of words according to different contexts. In fact, WSD has been demonstrated to be very useful to improve several NLP tasks such as Parsing, Machine Translation, Information Retrieval or Question Answering.

In WSD we find different kind of systems that can be classified into three main groups [15, 22]: supervised, unsupervised and knowledge-based. The first ones need lots of hand-tagged data in order to achieve enough information for training different types of classifiers. Unsupervised systems use external information from raw unannotated corpora and knowledge-based systems rely on dictionaries, thesauri and lexical knowledge databases without using any corpus evidence. Compared with unsupervised and knowledge-based systems, supervised systems historically have achieved better results in WSD [22]. However, recent results show that enriching

unsupervised and knowledge-based systems with new knowledge sources performs better than supervised systems [1, 25]. Related to knowledge-based systems, different approaches have been addressed to graph-based methods obtaining promising results. We can mention those approaches using structural interconnections such as Structural Semantic Interconnections (SSI) [23] which create structural specifications of the possible senses for each word in a context. Another approach is the proposal of exploring the integration of WordNet¹ (WN) [7] and FrameNet [16] and among others we can mention those using Page-Rank such as Agirre and Soroa [1], Redy et. Al. [26], Soroa et. Al. [28], and Sinha and Mihalcea [27] that used the internal interconnections of Lexical knowledge Base (LKB) from WordNet.

In this paper we present a new graph-based approach for the resolution of semantic ambiguity of words. It takes into account the internal semantic relations of WN and the relations that exist among the synsets of WN and the labels of SUMO² (Suggested Upper Merged Ontology), WordNet Domains³ (WND) and WordNet Affect⁴ (WNA). Our graph-based proposal is an adaptation of the N-Cliques model [17] applying the Clique Partitioning Technique [29] to N distance in order to obtain N-Cliques. Each N-Clique will contain relevant information used to extract the correct sense of each word. In order to build the initial graph our proposal combines the information provided by the synsets of WN with the ten most relevant concepts obtained from WN, SUMO, WND and WNA, using the Relevant Semantic Trees method (RST).

The organization of this paper is as follows: Section 2 describes the resources used. In Section 3 we present our algorithms and methods. Section 4 shows the integration of the resources, algorithms and methods. Section 5 shows the evaluation and a comparative study of the results. Finally, conclusions and further works are detailed.

2 ISR-WN Resource

In this section we describe a semantic network that links different resources: WN, WND, SUMO and WNA. This resource is called ISR-WN [9, 11]. The aim of developing this resource is that we need to extract and relate information from different lexical resources to obtain a multi-conceptual network.

In order to find the right environment to apply our graph-based approach we have analyzed several resources that align different kind of semantic information. Many efforts have been focused on the idea of building semantic networks like MultiWordNet (MWN) [24], EuroWordNet (EWN) [6], Multilingual Central Repository (MCR) [2], etc. For example: MWN is able to align the Italian and English lexical dictionaries conceptualized by Domain labels. The MultiWordNet browser also allows to access to the Spanish, Portuguese, Hebrew, Romanian and Latin WordNets, but these wordnets are not part of the MultiWordNet distribution. EWN was developed to align Dutch, Italian, Spanish, German, French, Czech, English and other lexical dictionaries. MCR integrates into the EWN framework an

¹ <http://www.cogsci.princeton.edu/~wn/>

² <http://www.ontologyportal.org>

³ <http://wndomains.fbk.eu>

⁴ <http://wndomains.fbk.eu/wnaffect.html>

upgraded version of the EWN Top Concept ontology, the MWN Domains, SUMO and hundreds of thousands of new semantic relations and properties automatically acquired from corpora.

ISR-WN, takes into account different kind of labels linked to WN: Level Upper Concepts (SUMO), Domains and Emotion labels. In this work our purpose is to use a semantic network which links different semantic resources aligned to WN. After several tests we decided to apply ISR-WN. Although each resource presented above provides different semantic relations, ISR-WN has the highest quantity of semantic dimensions aligned, so it is a suitable resource to run our algorithm. Using ISR-WN we are able to extract important information from the interrelations of four ontological resources: WN, WND, WNA and SUMO. ISR-WN resource is based on WN1.6 or WN2.0 versions. In the last updated version, Semantic Classes and SentiWordNet were also included but these new dimensions are not taken into account in this work. Furthermore, ISR-WN provides a tool that allows the navigation across internal links. At this point, we can discover the multidimensionality of concepts that exists in each sentence. In order to establish the concepts associated to each sentence we apply Relevant Semantic Trees [10, 12] approach using the provided links of ISR-WN.

Since the Clique Partitioning Technique of our approach needs as input data an initial graph, we use ISR-WN to extract an initial graph from each sentence. Each graph will be based on the minimal path obtained among the Relevant Concepts and the senses of all words in each sentence. In next section we describe the algorithm of our graph-based approach and the method that obtains the Relevant Concepts.

3 Algorithm and Method

An increasing interest on graph-based proposals in the scientific community has motivated the appearance of different techniques applied to WSD. As we have mentioned above, Page Rank technique is an effective performance among the novel graph-based proposals. Although each approach is different, all of them have two important elements in common: the technique used and the network where it will be applied. In this section, we present our proposal of clustering technique and a method to obtain the most relevant concepts of one sentence using the LKB from ISR-WN.

3.1 Clique Partitioning Technique

Taking into account ISR-WN and its useful properties, we propose to use them to represent the analyzed sentences in a conceptual level (in this step we need to obtain a sub-graph as described in section 4.1) also we need to reduce the semantic sub-graphs (clusters/cliques) to identify relevant nodes within the network. Our proposal is to use the Clique Partition Technique to obtain and discard nodes to finally build useful semantic sub-graphs. This algorithm (introduced by Tseng [29]) is able to extract clusters of strongly related nodes from a graph. It is based on building sets of elements from a graph structure using the Clique model.

The Clique model was formally defined by Luce and Perry [18] and they provided this statement: “A Clique is a set of more than two people if they are all mutual friends of one another”. As we can see, this model had its origin in Social Network studies.

In order to understand what is a Clique in terms of graphs definition, we present the following explanation by Cavique et. al. [4]: “Given an undirected graph $G = (V, E)$ where V denotes the set of vertices and E the set of edges, the graph $G_I = (V_I, E_I)$ is called a sub-graph of G if $V_I \in V, E_I \in E$ and for every edge $(v_i, v_j) \in E_I$ the vertices $v_i, v_j \in V_I$. A sub-graph G_I is said to be complete if there is an edge for each pair of vertices”. Each complete sub-graph is also called a Clique.

As we can appreciate the Clique model proposal obtains complete sub-graphs where the maximal distance between the vertices is one edge. Due to the fact that the semantic network where we will apply the Partitioning Technique is integrated by thousand of vertices, the maximal distance between the vertices must be increased. In order to decide which model we should use to modify the original algorithm we have studied different authors [3, 8, 14, 17] with N-Cliques, [3] with K-Plex and [20] with Clubs and Clans.

After studying different models we considered to use N-Cliques model to Partitioning Technique which is very similar to Cliques model. Instead we use N distance among vertices of each complete sub-graph. This partitioning idea was introduces on WSD by Gutiérrez et. al. [14], which was assumed by us. In order to apply this technique, we only have taken into account the creation of one N-Clique and the rest of the complete sub-graph will be Cliques. Our goal is to centralize the highest quantity of semantic information completely on one N-Clique (explained more in detail in [14]). To understand this algorithm we show an example at Fig 1 using $N = 2$ as edges distances.

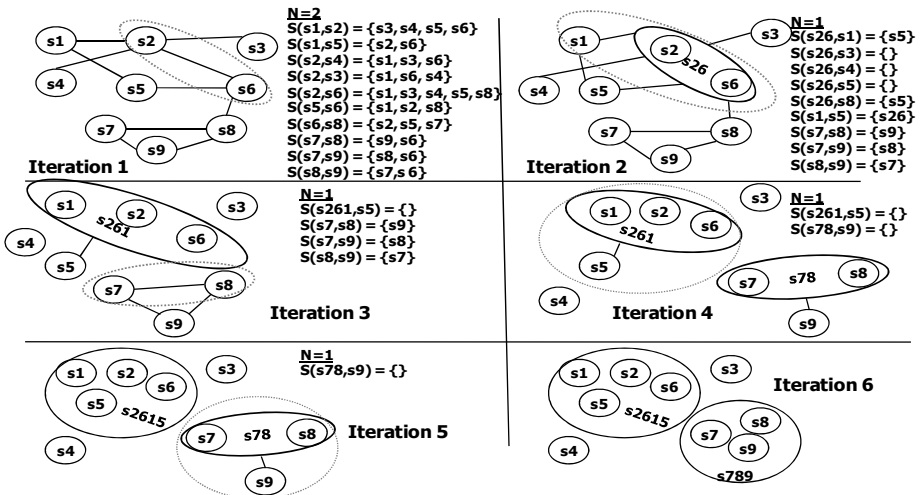


Fig. 1. Example of Clique Partitioning Algorithm to $N = 2$ edges distances

As we can see this heuristic algorithm is able to obtain one set of nodes with maximal distance among all nodes ≤ 2 edges, because $N = 2$. And it continues obtaining Cliques with $N = N - 1$ where $N \geq 1$ for each iteration while $E \neq \emptyset$. As we have described in previous section, the ISR-WN resource is used to apply this algorithm.

3.2 Relevant Semantic Trees (RST)

RST is a method which is able to analyze a sentence and obtain Relevant Semantic Trees from different resources. Therefore, it is possible to get trees that conceptualize the sentences in a multidimensional performance. This approach can be used with different resources mapped to WN.

In order to establish the Concepts associated to each sentence we apply Relevant Semantic Trees [10, 12, 13] using the provided links of ISR-WN.

In this section, we use a fragment of the original RST method with the aim of obtaining Relevant Semantic Trees of the sentences. Notice that this step must be applied for each resource and we have to obtain RST's from the WN taxonomy, SUMO, WND and WNA. Once each sentence is analyzed the Association Ratio (*AR*) value is obtained and related to each concept in the trees. Equation (1) is used to measure and obtain the values of Relevant Concepts:

$$AR(C, s) = \sum_{i=1}^n AR(C, w_i) \tag{1}$$

Where

$$AR(C, w) = P(C, w) * \log_2 \frac{P(C, w)}{P(C)} \tag{2}$$

In each equation *C* is a concept (could be a label from WN, WND, WNA or SUMO, according to the loaded dimension (resource)); *s* is a sentence or a set of words (*w*); *w_i* is the *i*-th word (*w*) of the sentence *s*; *P(C, w)* is joint probability distribution; and *P(C)* is marginal probability.

The first stage is to Pre-process the sentence to obtain all lemmas. For instance, in the sentence “But it is unfair to dump on teachers as distinct from the educational establishment.” the lemmas are: [be, unfair, dump, teacher, distinct, educational, establishment].

Next, each lemma is searched through ISR-WN resource and it is correlated with concepts of WND (the dimension used in this example). Table 1 shows the results after applying Equation (1) over the sentence.

Table 1. Initial Vector of Domain

<i>AR</i>	<i>Domain</i>	<i>AR</i>	<i>Domain</i>	<i>AR</i>	<i>Domain</i>
0.90	Pedagogy	0.36	Politics	0.36	Quality
0.90	Administration	0.36	Environment	0.36	Psychoanalysis
0.36	Buildings	0.36	Commerce	0.36	Economy

After obtaining the Initial Concept Vector of Domains we apply the Equation (3) in order to build the Relevant Semantic Tree related to the sentence.

$$AR(PC, s) = AR(ChC, s) - ND(IC, PC) \tag{3}$$

Where:

$$ND(IC, PC) = \frac{MP(IC, PC)}{TD} \tag{4}$$

Where $AR(PC, s)$ represents the AR value of Parent Concept (PC) related to the sentence s ; $AR(ChC, s)$ is the AR value calculated with Equation (1) in case of Child Concept (ChC) was included in the Initial Vector, otherwise is calculated with the Equation (3); ND is a Normalized Distance; IC is the Initial Concept from we have to add the ancestors; TD is Depth of the hierarchic tree of the resource to use; and MP is Minimal Path.

For example, Fig. 2 shows a part of the WND hierarchy. To build the RST from the hierarchy line: “Administration”, “Social_Science” and “Root_Domain” we need some iterations. In the first iteration, IC is “Administration”, ChC is “Administration” and PC is “Social_Science”. In the second, IC is “Administration”, ChC is “Social_Science” and PC is “Root_Domain”. Applying the Equation (3), the algorithm to decide which parent concept will be added to the vector is showed here:

```

if (AR(PC,s) > 0 ){
    if ( PC had not been added to vector)
        PC is added to the vector with AR(PC,s) value;
    else PC value = PC value + AR(PC,s) value; }

```

This bottom-up process is applied for each Concept of the Initial Vector to add each Relevant Parent to the vector. After reproducing the process to each Concept of the Initial Vector, the RST is built. As a result, Table 2 is obtained. This vector represents the domain tree associated to the sentence such as Fig 2 shows. As we can see, the Relevant Semantic Tree of domains in Fig 2 has associated a color intensity related to the AR value of each domain. The more intense the color is the more related AR is.

Table 2. Final Domain Vector based on WND

<i>AR</i>	<i>Domain</i>	<i>AR</i>	<i>Domain</i>	<i>AR</i>	<i>Domain</i>
1.63	Social_Science	0.36	Economy	0.36	Environment
0.90	Administration	0.36	Quality	0.11	Factotum
0.90	Pedagogy	0.36	Politics	0.11	Psychology
0.80	Root_Domain	0.36	Buildings	0.11	Architecture
0.36	Psychoanalysis	0.36	Commerce	0.11	Pure_Science

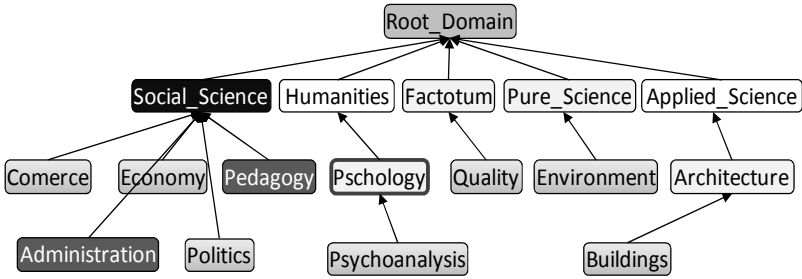


Fig. 2. Relevant Semantic Tree from WND

As result, for each resource included (WN, WND, WNA and SUMO) an RST is obtained. Next, it is explained the WSD global method.

4 Global Method Structure

Once each constituent of our proposal has been described we present our method. It consists of three steps: 1. Creating an initial semantic graph. 2. Applying N-Cliques Partitioning Technique. 3. Selecting the correct senses. Everything begins when a sentence is introduced. After that, the lemmas of the words are obtained and next the three steps are applied. Following, we describe the three steps.

4.1 Creating an Initial Semantic Graph

The aim of this step consists of building a semantic graph from all senses of the words in each sentence. The connections among these senses are established using Breath First Search (BFS) over ISR-WN, between all senses vs all the most relevant Concepts (using Relevant Semantic Trees based on ISR-WN). The process of building the initial graph (Fig 3) guarantees that the graph built will be more centralized and the diameter between the corners will be the shortest.

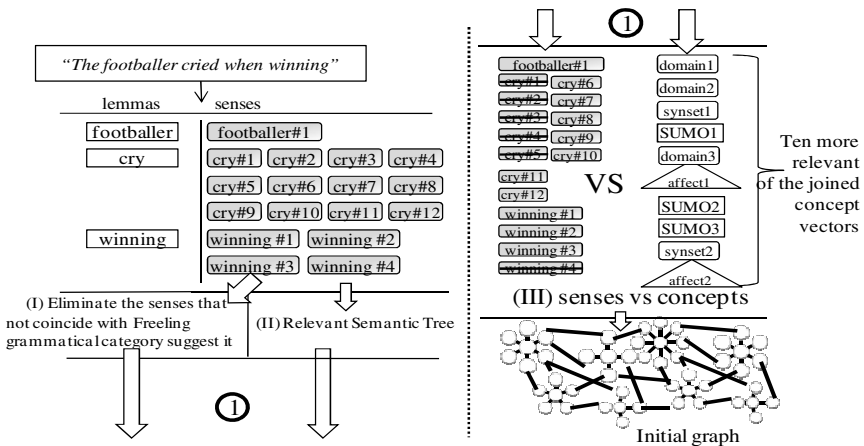


Fig. 3. Initial graph creation with all senses vs the ten most relevant concepts

These are the steps to build the initial graph:

- I. First, we get all the senses from the lemmas of the sentence and next we have to discard the senses that not match with the grammatical category suggested by Freeling⁵ Pos-Tagger.
- II. Next, we apply the RST method (see section 3.2) to obtain the Relevant Semantic Trees and then select the ten⁶ most Relevant Concepts among all Trees.
- III. Finally, we have to obtain the shortest path among all senses according to the Pos (I) and all the selected concepts (II); and then create the Initial Graph avoiding repeated nodes.

It is important to remark that the initial graph is composed by synsets (senses), Domains, Emotion labels and SUMO categories.

4.2 Applying N-Cliques Partitioning Technique

Once we have obtained the initial graph we apply N-Cliques Partitioning Technique, the N-Cliques will be composed by all the elements that appeared in the initial graph.

4.3 Selecting the Correct Senses

After obtaining the N-Cliques, to select the correct senses it is necessary to sort the N-Cliques by quantity of nodes. Later, for each lemma of the sentence it is checked if the N-Cliques contains its corresponding synset. If the synset does not exist in the first N-Clique we check next, and so on. The synset selected in this process will be considered as the right sense for each lemma. In case of finding two or more synsets for one lemma in a N-Clique, we select as correct the synset that the Freeling tool returns as the most frequent in the sentence where it appears. If the lemma to disambiguate is the verb “be” we always select the Most Frequent Sense for this lemma (the explanation of this proceeding can be followed in Gutiérrez et. al. [14]).

5 Evaluation

Our method has been evaluated using the “English All Words” task corpus of Senseval-2 [5] competition. The analysis has been divided into eight experiments that have been analyzed in detail to evaluate the influence of different combinations of resources. The experimental distance used in the Partitioning Technique was $N = 3$. This distance is more effective and faster than other upper distances. However, using minor distances produces worst results. The experiments are described next applying the proposal with a graph composed by:

1. WN, WND, WNA and SUMO, using RST of Domains.
2. WN, WND, WNA and SUMO, using RST of SUMO.
3. WN, WND, WNA and SUMO, using RST of Affects.

⁵ <http://nlp.lsi.upc.edu/freeling/>

⁶ After conducting several experiments the ten most relevant concepts have demonstrated to be the best choice.

4. WN, WND, WNA and SUMO, using RST of WN taxonomy.
5. WN, WND, WNA and SUMO, using RST of Domains and SUMO.
6. WN, WND, WNA and SUMO, using RST of Domains, SUMO, Affects and WN taxonomy.
7. WN and SUMO, using RST of SUMO.
8. WN and WND, using RST of Domains.

5.1 Analysis of the Behavior of Different Grammatical Categories

Each experiment has been analyzed in order to find out the behavior of each grammatical category. We can see in Table 3 that the adverbs disambiguation obtains the highest accuracy, reaching 67%. On the other hand, we can notice that verbs and adjectives have a low accuracy, respectively. After analyzing the results, our method can be considered powerful to disambiguate adverbs. Nouns and adjectives reached around 49% and 35% respectively. Finally, verbs obtain the worst results around 24% because they are the most polysemous⁷.

5.2 General Analysis

In this section we have conducted a general analysis to detect if the integration of WordNet resources helps the WSD task. Moreover, we are interested in knowing if the selection of Relevant Concepts has influence on the results obtained.

Such as Table 3 shows almost all the experiments that integrate all Dimensions improved the results of the Experiments 7 and 8, with the exception of Experiment 6. This indicates that the applied Technique to partition the contextual graph (initial graph) has effective performance using more information. The accuracy was computed using the same measures as in Senseval competition. As baseline we assumed the original proposal by Gutiérrez et al. [14] (N-Cliques+RV), which used Reuter Vector method [19] proposed by Magnini et. al. to obtain the Relevant Concepts. The main difference is using RST instead the Reuter Vector to build the initial graph.

Table 3. Accuracy of the experiments: Precision (P) and Recall (R)

	<i>Total (P)</i>	<i>Total (R)</i>	<i>Noun (R)</i>	<i>Adj(R)</i>	<i>Verb(R)</i>	<i>Adv(R)</i>
Baseline N-Cliques+RV	0.444	0.433	0.489	0.359	0.239	0.646
Exp 1	0.436	0.426	0.490	0.353	0.231	0.639
Exp 2	0.416	0.407	0.452	0.336	0.232	0.649
Exp 3	0.515	0.413	0.467	0.323	0.243	0.646
Exp 4	0.414	0.405	0.454	0.325	0.224	0.657
Exp 5	0.414	0.405	0.451	0.333	0.228	0.652
Exp 6	0.405	0.397	0.450	0.330	0.228	0.652
Exp 7	0.425	0.402	0.483	0.345	0.197	0.585
Exp 8	0.406	0.398	0.445	0.315	0.215	0.672

⁷ Later of an analysis over WordNet 1.6 and 2.0, the polysemy averages for each grammatical category respectively are the next: verbs (≈ 2.138 , ≈ 2.179), adjectives (≈ 1.481 , ≈ 1.447), adverbs (≈ 1.249 , ≈ 1.246) and nouns (≈ 1.231 , ≈ 1.236).

The most relevant experiment was that use all mentioned resources (creating the initial graph with RST of Domains) where the recall obtained reached 42.6%. That indicates that WND is a semantically more influential resource than the others, when all semantic dimensions conform the knowledge base. This result could locate our proposal in the 11th place of Senseval-2 ranking. It is important to remark that using WNA RST on ISR-WN improved all precision results obtained by our system. Moreover, the usage of the affective dimension could be more effective if the evaluated context was related to emotions.

5.3 Comparison with Previous WSD Graph-Based Approaches

Previously, we mentioned some relevant graph-based approaches. To measure and compare them with ours we have selected those evaluated over “English All Words” task corpus from Senseval-2.

Firstly, we start analyzing the results obtained by the original proposal N-Cliques+RV compared with ours. In N-Cliques+RV all the remaining process is very similar to our proposal, the main difference is related to replace the Reuter Vector by RST to build the initial graph. The best results obtained by the original proposal are around 43.3% and ours reached 42.6%. However, accuracy obtained by adverbs of our approach is 67.2%, which increased the previous proposal on 2.3%. Our conclusion is that the two proposals are very similar, due to the fact that both proposals have not obtained significant differences in their results.

In the other hand, the very popular Page-Rank method has been used on different tasks such as Classification of Documents or Web Page Indexing and it has been used again by different authors Agirre and Soroa [1], Redy et. Al. [26], Soroa et. Al. [28] and Sinha and Mihalcea [27] to solve WSD. At this time, Page Rank has been used to determine the centrality of structural lexical networks. In these proposals cited above, context is used to build sub-graphs similarly to our approach. Then, to disambiguate each word it is chosen the most weighted sense. The difference with our proposal is that they are able to assign weights to the labels ((nodes) senses) using the semantic relations of WordNet. This important element will be considered of inclusion on new proposals as future work.

Essentially, both proposals of Agirre and Soroa and Sinha and Mihalcea, executed over Senseval-2 data set, conducted the same process with one difference; Mihalcea applied an experimental phase, first with six different similarity measures assigned to the edges and then applied four different measures of centrality. Despite of applying these measures Mihalcea’s results did not improve Agirre’s. These approaches obtained respectively 58.6% and 56.37% of recall. In comparison with ours, we do not improve these results obtaining a 42.6% of recall. The reason of the lower recall obtained is due to our proposal non-apply weight techniques to sort senses by different scores.

6 Conclusions and Further Works

In this paper we propose a new graph-based method to solve Word Sense Disambiguation that uses the internal semantic relations from ISR-WN (Integration of Semantic Resource based on WordNet). Our proposal uses an adaptation of Clique

Partition Technique using Relevant Semantic Trees in order to identify sets of strongly related senses. With this approach we are able to obtain many densely connected sub-graphs where each sub-graph will contain a series of proposed senses. In case of obtaining only one sense per word, it will be selected as the right sense into the sub-graph. However, if there are located several senses for the same word in a sub-graph, it will be selected the Most Frequent Sense as right sense.

The evaluation of our approach was conducted using Senseval-2 corpus, obtaining propitious results. To improve the selection criterion of right senses we could compute the sum of weights (frequencies, page rank or similarity measures values).

After analyzing several approaches that took into account structural semantic networks, we discovered that it is very important to assign weights to nodes in order to distinguish the priority of the senses in the clusters of lexical structures. To do that, we propose as future work applying Page-Rank to the Cliques obtained in our approach, to analyze and evaluate if the results can be improved. Moreover, we want to evaluate Page-Rank over ISR-WN without applying Cliques. The aim of this experiment is to determine if the ISR-WN resource works better than other resources when using Page-Rank. Besides, in order to enrich the knowledge base we plan to include into ISR-WN gloss relations from eXtended WordNet [21].

Acknowledgments. This paper has been supported partially by Ministerio de Ciencia e Innovación - Spanish Government (grant no. TIN2009-13391-C04-01), and Conselleria d'Educación - Generalitat Valenciana (grant no. PROMETEO/2009/119, ACOMP/2010/288 and ACOMP/2011/001).

References

1. Agirre, E., Soroa, A.: Personalizing PageRank for Word Sense Disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), Athens, Greece (2009)
2. Atserias, J., Villarejo, L., Rigau, G., Agirre, E., Carroll, J., Magnini, B., Vossen, P.: The MEANING Multilingual Central Repository. In: Proceedings of the Second International Global WordNet Conference (GWC 2004), Brno, Czech Republic (2004)
3. Balasundaram, B., Butenko, S., Hicks, I.V., Sachdeva, S.: Clique relaxations in Social Network Analysis: The Maximun k-plex Problem (2006)
4. Cavique, L., Mendes, A.B., Santos, J.M.A.: An Algorithm to Discover the k-Clique Cover in Networks. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS, vol. 5816, pp. 363–373. Springer, Heidelberg (2009)
5. Cotton, S., Edmonds, P., Kilgarriff, A., Palmer, M.: English All word. In: SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems. Association for Computational Linguistics, Toulouse, Toulouse (2001)
6. Dorr, B.J., Castellón, M.A.M.: Spanish EuroWordNet and LCS-Based Interlingual MT. In: AMTA/SIG-IL First Workshop on Interlinguas, San Diego, CA (1997)
7. Fellbaum, C.: WordNet. An Electronic Lexical Database. The MIT Press, University of Cambridge (1998)
8. Friedkin, N.E.: Structural Cohesion and Equivalence Explanations of Social Homogeneity. *Sociological Methods & Research* 12, 235–261 (1984)

9. Gutiérrez, Y., Fernández, A., Montoyo, A., Vázquez, S.: Integration of semantic resources based on WordNet. In: XXVI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural, SEPLN 2010, vol. 45, pp. 161–168. Universidad Politécnica de Valencia, Valencia (2010)
10. 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, Uppsala (2010)
11. Gutiérrez, Y., Fernández, A., Montoyo, A., Vázquez, S.: Enriching the Integration of Semantic Resources based on WordNet. *Procesamiento del Lenguaje Natural* 47, 249–257 (2011)
12. Gutiérrez, Y., Vázquez, S., Montoyo, A.: Improving WSD using ISR-WN with Relevant Semantic Trees and SemCor Senses Frequency. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, RANLP 2011 Organising Committee, Hissar, Bulgaria, pp. 233–239 (2011)
13. Gutiérrez, Y., Vázquez, S., Montoyo, A.: Sentiment Classification Using Semantic Features Extracted from WordNet-based Resources. In: Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2011), pp. 139–145. Association for Computational Linguistics, Portland (2011)
14. Gutiérrez, Y., Vázquez, S., Montoyo, A.: Word Sense Disambiguation: A Graph-Based Approach Using N-Cliques Partitioning Technique. In: Muñoz, R., Montoyo, A., Métais, E. (eds.) NLDB 2011. LNCS, vol. 6716, pp. 112–124. Springer, Heidelberg (2011)
15. Ide, N., Véronis, J.: Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics* 24, 2–40 (1998)
16. Laparra, E., Rigau, G., Cuadros, M.: Exploring the integration of WordNet and FrameNet. In: Proceedings of the 5th Global WordNet Conference (GWC 2010), Mumbai, India (2010)
17. Luce, R.D.: Connectivity and generalized cliques in sociometric group structure. *Psychometrika* 15, 159–190 (1950)
18. Luce, R.D., Perry, A.D.: A Method of Matrix Analysis of Group Structure. *Psychometrie* 14, 95–116 (1949)
19. Magnini, B., Strapparava, C., Pezzulo, G., Gliozzo, A.: Comparing Ontology-Based and Corpus-Based Domain Annotations in WordNet. In: Proceedings of the First International WordNet Conference, Mysore, India, pp. 21–25 (2002)
20. Mokken, R.J.: *Cliques, Clubs and Clans*, vol. 13. Elsevier Scientific Publishing Company, Amsterdam (1979)
21. Moldovan, D.I., Rus, V.: Explaining Answers with Extended WordNet. *ACL* (2001)
22. Navigli, R.: Word sense disambiguation: A survey. *ACM Comput. Surv.* 41, 10:11–10:69 (2009)
23. Navigli, R., Velardi, P.: Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 27 (2005)
24. Pianta, E., Bentivogli, L., Girardi, C.: MultiWordNet. Developing an aligned multilingual database. In: Proceedings of the 1st International WordNet Conference, Mysore, India, pp. 293–302 (2002)
25. Ponzetto, S.P., Navigli, R.: Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems. In: *ACL*, pp. 1522–1531 (2010)

26. Reddy, S., Inumella, A., McCarthy, D., Stevenson, M.: IIITH: Domain Specific Word Sense Disambiguation. In: Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Uppsala (2010)
27. Sinha, R., Mihalcea, R.: Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA (2007)
28. Soroa, A., Agirre, E., de Lacalle, O.L., Bosma, W., Vossen, P., Monachini, M., Lo, J., Hsieh, S.-K.: Kyoto: An Integrated System for Specific Domain WSD. In: Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics, Uppsala (2010)
29. Tseng, C.-J., Siewiorek, D.P.: Automated Synthesis of Data Paths in Digital Systems. *IEEE Trans. on CAD of Integrated Circuits and Systems* 5, 379–395 (1986)

Using Wiktionary to Improve Lexical Disambiguation in Multiple Languages

Kiem-Hieu Nguyen and Cheol-Young Ock

School of Electrical Engineering
University of Ulsan

93, Daehakro, Nam-gu, Ulsan 680-749, Korea
{hieunk, okcy}@ulsan.ac.kr

Abstract. This paper proposes using linguistic knowledge from Wiktionary to improve lexical disambiguation in multiple languages, focusing on part-of-speech tagging in selected languages with various characteristics including English, Vietnamese, and Korean. Dictionaries and subsumption networks are first automatically extracted from Wiktionary. These linguistic resources are then used to enrich the feature set of training examples. A first-order discriminative model is learned on training data using Hidden Markov-Support Vector Machines. The proposed method is competitive with related contemporary works in the three languages. In English, our tagger achieves 96.37% token accuracy on the Brown corpus, with an error reduction of 2.74% over the baseline.

Keywords: Wiktionary, collaborative dictionary, lexical disambiguation, part-of-speech tagging, supervised learning, discriminative model.

1 Introduction

Lexical ambiguity, either syntactic or semantic, is a common characteristic of natural language. This paper addresses syntactic disambiguation in the form of part-of-speech (POS) tagging, i.e., mapping every word in a text to its appropriate syntactic category based on its context. POS tagging is essential for high-level problems like syntactic parsing, chunking, and word sense disambiguation. State-of-art POS taggers in English have exceeded the human performance (about 97.3% token accuracy), e.g., by using machine learning methods with rich and carefully studied features [1], or with a large amount of untagged texts [2]. However, because about 20% of errors are, in fact, due to linguistic issues, there is still more ground for improvement using linguistic knowledge [3].

Since its launch in 2002 as a multilingual collaborative dictionary, Wiktionary has been growing rapidly¹. It has gained more and more attention of the NLP community. For instance, Wiktionary was mined for semantic analysis, information retrieval, semantic relatedness calculation, and ontology construction [4-7]. To our knowledge,

¹ The English Wiktionary snapshot on 27th, August, 2011, contains 2,803,610 articles.

there has been no study in using Wiktionary for lexical disambiguation, including POS tagging.

The main contribution of this paper is the use of linguistic knowledge in Wiktionary to improve lexical disambiguation in multiple languages. The paper focuses on POS tagging in selected languages with various characteristics including English, Korean, and Vietnamese. Because the proposed method is language-independent, it can be applied to any language in Wiktionary.

The paper is organized as follows. Section 1 introduces our motivation. Section 2 surveys related works on Wiktionary mining and POS tagging in some major languages. Section 3 explains in details our method, particularly the use of linguistic knowledge in Wiktionary. The experimental results are discussed in Section 4. The paper ends with conclusions and future works.

2 Related Works

Wikipedia mining has been recently an active research area [8]. However, Wiktionary, another Wikimedia project, has not been paid as much attention as its sibling. Wikipedia and Wiktionary are both multilingual but are different in their contents. The former contains a huge number of conceptual entities and semantic relations. Therefore, it facilitates and attracts any kind of semantic mining (concept mining, relationship extraction, or ontology construction). The latter, in contrast, is more likely a lexicon containing word senses and lexical relations (WordNet [9]) or a dictionary with various kinds of linguistic categories (e.g., lexical, semantic, topical, grammatical, and etymological categories).

Most of previous studies have considered Wiktionary as a lexicon in English. Zesch et al. [6] showed that Wiktionary is superior to Wikipedia and WordNet for computing semantic relatedness on a benchmark test data of word pairs. Navarro et al. [10] extracted a synonymy network from Wiktionary. Meyer and Gurevych [7,11] induced an ontology from Wiktionary and reported an accuracy 66.1% of alignment between Wiktionary word senses and WordNet synsets. There has been no extrinsic evaluation proving that Wiktionary helps improve an existing NLP task. This work, for the first time, addresses Wiktionary as a dictionary and uses its linguistic knowledge to improve POS tagging.

Since the landmark rule-based Brill tagger [12], machine learning has been dominant in POS tagging in forms of supervised, unsupervised or semi-supervised learning [13]. The performance in English has recently exceeded the human inter-annotator agreement of 97% [1,2]. POS tagging in languages such as Chinese, Korean, and Vietnamese is more difficult because of the need of word segmentation or morphological analysis to resolve word boundary ambiguity [14,15]. Because the performances of word segmentation and morphological analysis have not achieved 100% (and it is definitely hard to achieve), it takes much time and effort to build large treebanks in these languages. Methods involving untagged texts (e.g., semi-supervised and unsupervised methods) also suffer from word segmentation and morphological analysis errors. While a joint model of word segmentation (or morphological analysis) and POS tagging [16] is a reasonable solution, it unavoidably increases system complexities.

For consistency, this work simplifies the differences between the investigated languages. Based on the availability of annotated data, we ignore the need of word segmentation in Vietnamese and morphological analysis in Korean both on training and test examples. Therefore, the contribution of linguistic knowledge from Wiktionary could be emphasized.

3 POS Tagging Problems

3.1 POS Tagging in Korean

A sentence in Korean text is composed of *ejoels* (어절) separated by spaces. An *ejeol* is vaguely equivalent to a word-phrase. An *ejeol* might be a word itself like '실내' (indoor) or a composition of content words and functional words like '실내용품' (interior supply). Whenever necessary, an *ejeol* is decomposed into content words and functional words by morphological analysis as '실내+용품+을' (words in the same *ejeol* are separated by '+').

This word composition consequently divides a POS tagset in Korean into two separate groups, one for content words, and the other for functional words [17,18]. E.g., given a sentence *엠마누엘 옹가로 / 의상서 실내 장식품으로... 디자인 세계 넓혀*. The sentence is morphologically analyzed as *엠마누엘 옹가로 / 의상+서 실내 장식품 + 으로 + ... 디자인 세계 넓혀 + 어*, and jointly morphologically analyzed and POS tagged as *엠마누엘/NNP 옹가로/NNP //SP 의상/NNG + 서/JKB 실내/NNG 장식품/NNG + 으로/JKB + .../SE 디자인/NNG 세계/NNG 넓혀/VV + 어/EC*.

3.2 POS Tagging in Vietnamese

Word compositions in Vietnamese and Korean are somewhat opposite. A word may be composed of a single syllable (*tiếng*) like 'hoa' (flower) and 'học' (learn), or multiple syllables like 'máy tính' (computer) and 'nhà khoa học' (scientist). There is no explicit boundary between words in a Vietnamese text. Word segmentation is the task to automatically resolve word boundary ambiguity based on its context [19]. E.g., given a phrase where six syllables (three words) are separated by spaces, *xử lý ngôn ngữ tự nhiên* (natural language processing), it is then segmented into three words as *xử_lý ngôn_ngữ tự_nhiên* (syllables of the same word are grouped together by '_'), and POS tagged as *xử_lý/V ngôn_ngữ/N tự_nhiên/A*.

The tagset of Vietnamese treebank contains only coarse-grained tags [20]. While it is sufficient for our purpose, fine-grained POS annotated corpora will certainly benefit the Vietnamese NLP community.

3.3 Basic Concepts

This section describes the concepts used to name linguistic resources extracted from Wiktionary (Fig.1).

Wiktionary category network. A network contains nodes as categories and directed links as subsumption relations between categories. A node can be any kind of

categories such as Wiktionary administration, part-of-speech, domain, and etymology. A Wiktionary category network is roughly a forest of sub-networks like the Wiktionary administration and domain category networks.

Domain category network. A sub-network of the Wiktionary category network contains nodes as domain categories and directed links as sub-domain relations between domains. A domain category network is roughly an hierarchy except that some domains might have more than one super-domains (e.g., *communism* is both *economics* and *forms of government*).

Lexical category network. A sub-network of the Wiktionary category network containing nodes as lexical categories, i.e., all kinds of category except domain and Wiktionary administration, and subsumption between lexical categories. A lexical category network is roughly a forest of sub-networks like the grammatical and etymological category networks).

Lexical dictionary. Each entry of a lexical dictionary contains a word and all the lexical categories that it belongs to. This dictionary is derived from the categorization of words into categories in Wiktionary. While word categorization in Wiktionary is basically flat partitioning, the structural information in the lexical category network could be embedded into the lexical dictionary by simply adding the categories having subsumption relations with those in the word entry.

Domain dictionary. Each entry of a lexical dictionary contains a word and all the domain categories that it belongs to. It can be extended similarly to lexical dictionary using a domain category network.

3.4 Knowledge Acquisition from Wiktionary

As a multilingual dictionary, Wiktionary contains the translations of English word senses into other languages and vice versa. As a monolingual dictionary, besides the information about coarse-grain parts-of-speech, pronunciations, word senses, and examples, Wiktionary contains rich lexical and semantic information (Table 1).

Table 1. Wiktionary statistics. #word is the size of vocabulary (e.g., English words in the English Wiktionary). #lexicon is the number of lexical categories. #domain is the number of domain categories.

Language	Snapshot	#word	#lexicon	#domain
English	2011-08-27	389,941	2,412	1,162
Korean	2011-10-15	57,352	957	126
Vietnamese	2011-08-23	29,630	38	219

Wiktionary categories are divided into three groups: administrator, lexicon, and domain. The categories in *administrator* facilitate the maintenance of Wiktionary content. They are article templates, user information, article status, etc. These categories, hence, do not provide useful information for lexical disambiguation. The categories in *lexicon* provide the most useful information. Categories such as etymology, POS categorizations, personal names, numbers, and plural nouns are obviously important clues for POS tagging. The role of other categories such as prefixes, suffixes, infixes, and pronunciation for POS tagging, however, cannot be concluded. Since the number of *lexical categories* (about 2,400 categories in English)

is too large for the scope of our work to be manually selected, we use all kinds of lexical categories for consistency between the investigated languages.

Categories in *domain* define the domains of vocabulary. For polysemous words, different word senses might belong to different domains. Normally, semantic ambiguity must be resolved first to use the domains of selected word senses. This makes an egg-and-chicken problem because word sense disambiguation (WSD), in turn, necessitates POS tagging as preprocessing. Although it is not mandatory, POS information is an important feature for both supervised and knowledge-based WSD [21,22]. In this work, word senses are ignored. A word, hence, belongs to all the categories as described in its entry in the *domain dictionary*.

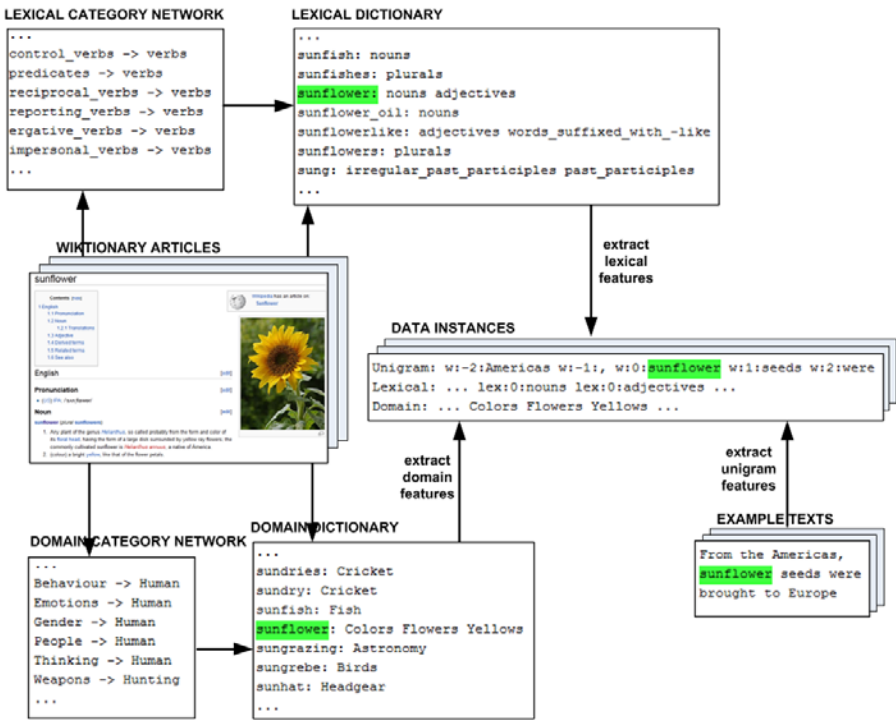


Fig. 1. Using linguistic knowledge from Wiktionary as learning features for 'sunflower'

Lexical (or domain) categories from a Wiktionary category network are extracted based on the fact that sub-networks in a Wiktionary category network are roughly hierarchical. This is done by starting from the root category and running breadth-first-search (BFS) to extract the accompanied sub-network. In the English Wiktionary, lexical and domain category titles start with the prefixes 'English' and 'en:', respectively. The extraction is hence done automatically by pattern matching. In the Vietnamese Wiktionary, lexical category titles end with the suffix 'tiếng Việt' (Vietnamese). However, domain category titles do not have a specific prefix or suffix. The root of domain category network is *Mục từ theo chuyên ngành* (vocabulary by domain). We use it as a seed category and run the algorithm. In the Korean

Wiktionary, the root of Wiktionary category network is **한글어** (Korean). All the top domain and lexical categories are its sub-categories. To extract only domain categories, we use **한글어** and 41 top domain categories as seeds and run BFS².

3.5 Hidden Markov-Support Vector Machines

Hidden Markov-Support Vector Machines [23] (HM-SVM) is both theoretically and empirically attractive for sequence labeling. The HM part of HM-SVM formulates Hidden Markov Models with high-order dependency between associated labels. The SVMs part indicates a discriminative learning technique based on a margin maximization principle. In comparison with other discriminative techniques, HM-SVM is capable of learning non-linear models via kernel functions.

HM-SVM is superior to Hidden Markov Models and Conditional Random Fields for named entity recognition and POS tagging in small scale experiments [23]. In this work, we investigate HM-SVM on large datasets. In addition, many Wiktionary-based features are used besides words in the context. These features inevitably contain redundant or irrelevant ones that make the learning of SVMs more difficult.

4 Experimental Results and Discussions

4.1 Experimental Setups

Tagging results are evaluated using the token tagging accuracy as measure:

$$token_accuracy = \frac{\#correct_predicted_tokens}{\#total_predicted_tokens} \quad (1)$$

Parameters for HM-SVM³ are chosen by default except C (i.e. smoothing factor between empirical and structural errors for SVMs): Linear kernel, and $epsilon = 0.1$ for SVMs; First-order transition and emission matrices for HMMs. C was optimized using the Vietnamese treebank [20] (80% training data and 20% development data with the baseline feature set *unigrams [-2,2]*). $C = 10^2$ achieves the best accuracy of 91.19% within the values in $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$.

Table 2. The POS annotated datasets in English, Korean, and Vietnamese

Dataset	Language	#sent	#token	#tag
Brown corpus	English	57,304	1,161,192	87
Sejong corpus	Korean	39,980	1,130,658	43
Vietnamese Treebank	Vietnamese	10,000	220,420	18

For evaluating, POS annotated datasets in English, Korean, and Vietnamese [17,20,24] (Table 2) are randomly divided into 80% training and 20% test data.

² The resources are available for download at

<http://nlplab.ulsan.ac.kr/hieunk/resources/cicling2012.zip>

³ http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

4.2 Feature Set

The *baseline* feature set is *unigram* containing position-sensitive unigrams (words) in a window $[-2,2]$. The *lexicon* contains position-sensitive features as lexical categories of words. The *domain* contains position-insensitive domain categories of words. The *lexicon (domain) + structure* is an extension of *lexicon (domain)* with parent categories from the lexical (domain) category network (Fig. 1).

We assume that word length is a strong indicator for lexical disambiguation, hence feature weight as word length instead of binary values. E.g, the *unigram* features of 'sunflower' in the sentence “*From the America, sunflower seeds were brought to Europe*” would be $\{w:-2:America/7, w:-1:/1, w:0:sunflower/9, w:1:seeds/5, w:2:were/4\}$ instead of $\{w:-2:America/1, w:-1:/1, w:0:sunflower/1, w:1:seeds/1, w:2:were/1\}$. We empirically verify this assumption by comparing binary vs. word length-based *unigram* feature sets on the Vietnamese dataset. It reveals that the word length-based feature set outperforms by 4.47% over the binary one (91.19% vs. 86.62%). In the future, we will verify this assumption on learning techniques other than HM-SVM.

4.3 Experimental Results

Lexical information stably improves performances in all the languages. The best feature set, i.e. *Unigram+Lexicon+Structure*, uses lexical and structural information in the lexical dictionaries and lexical category network. Domain information does not improve performances in Korean and Vietnamese. However, it improves the performance of English by 0.57%. The combination of lexical and domain information does not make any improvement (Table 3).

The highest accuracy is reported in Korean. The accuracy in Vietnamese is slightly lower, partially because the training data is not as big as in the other two languages. The best improvement over the baseline is achieved in English, by 2.74%. In Korean, Wiktionary only helps increase accuracy by 0.75%.

The results in English are close to the human performance. In Korean, our tagger is superior to the best reported results on the Sejong corpus (96.11%) [25]. In Vietnamese, the tagger is closed to the best reported accuracy (94.1%) [15]. It should be noticed that our taggers achieve such results with a language-independent feature set and little machine learning optimization: The baseline feature set, i.e., *unigram*, is very simple; A large number of lexical features are used without feature selection; Non-linear kernels are not used because of high computational complexity.

4.4 Error Analysis

Using domain information without WSD is not theoretically supported. However, an improvement in English by 0.57% is reported (Table 3). A reasonable though not very strong explanation is that domain information in English is very rich while that in Korean and Vietnamese is limited (Table 1). For future works, we should verify whether this improvement is only an artifact of data or not.

Table 3. Comparison of Wiktionary-based feature sets and the baseline feature set (The best performed results are displayed in bold face)

Feature set	English	Korean	Vietnamese
Unigram (baseline)	93.3	95.62	91.19
Unigram + Domain	93.74	95.62	91.27
Unigram + Domain + Structure	93.88	95.60	91.16
Unigram + Lexicon	95.94	96.31	92.13
Unigram + Lexicon + Structure	96.05	96.35	92.08
Unigram + Domain + Lexicon	96.01	96.35	92.26
Unigram + Domain + Lexicon + Structure	96.03	96.37	92.08

Table 4. Error analysis of *Unigram+Lexicon (U+L)* vs. *Unigram (U)*. *U+L:true U:false* indicates the improvement of *U+L* over *U*. *U+L:false U:true* indicates the cases in which *U+L* makes false predictions even though *U* can make correct predictions. *U+L:false U:true* is the case when both two feature sets cannot correctly predict. A pair *tag/number* indicates the tag and the percentage of improvement (or error) it contributes.

Prediction case	English	Korean	Vietnamese
U+L:true U:false	NP/15.62	NNP/35.33	NP/24.38
	NNS/13.31	NNG/8.91	V/24.32
	RB/10.31	SL/8.78	N/18.92
	JJ/9.37	SN/6.21	A/13.51
	Other/51.39	Other/40.77	Other/18.87
U+L:false U:true	NN/21.95	NNG/47.89	N/54.29
	NP/16.31	NNP/14.56	V/
	JJ/11.89	JX/8.59	A/
	VBD/7.70	VV/5.24	E/
	Other/42.15	Other/23.72	Other/
U+L:false U:false	NP/17.67	NNP/42.08	NP/39.02
	JJ/8.28	SL/19.38	N/14.63
	NP\$/7.70	SW/7.75	V/14.63
	NPS/5.94	SH/7.20	C/4.88
	Other/60.41	Other/23.59	Other/26.84

Performances in the three languages are not directly comparable because of the inconsistency between POS tagsets and feature sets. Firstly, POS tagsets are language-dependent. They have different size and scale, i.e., coarse-grained and fine-grained scale. Secondly, lexical categories contain redundant and irrelevant ones: 38 lexical categories in Vietnamese improves by 1.07%; 957 lexical categories in Korean improves by 0.73%; and 2,412 lexical categories in English improves by 2.74% (Table 1, 3). A manual selection by linguists or at least an automatic feature selection could yield more improvement.

To emphasize the effect of lexical information, the tagging results of *Unigram* and *Unigram+Lexicon* are compared in details (Table 4, 5).

The *proper noun* tag contributes the most to improvement among the tags because of *personal name, abbreviations, acronyms* and *trademark* categories in Wiktionary ($U+L:true$ $U:false$). The rest of improvement mainly results from syntactic categories in Wiktionary. For instance, according to Wiktionary, 'sunflower' can only be tagged as a noun or an adjective (Fig. 1). If 'sunflower' only occurs in training examples as a noun, the *unigram* model will probably tag any occurrence of 'sunflower' as a noun in training examples. The *unigram+lexicon* model will give less probability to a noun and increase the chance for an adjective. If 'sunflower' is an unknown word, i.e. a word not existing in training examples, *unigram* has no clue for tagging, hence tends to select the most frequent tag in the training corpus. In contrast, *unigram+lexicon* dramatically reduces the candidates from all the tagset to only a noun and a word.

Reversely, the *noun* tag contributes the most to failure among the tags ($U+L:false$ $U:true$). This implies that the *noun* tag is better recognized by using only context words rather than by using additional lexical knowledge. Finally, for the case that both *unigram* and *unigram+lexicon* are failed, even Wiktionary is very helpful for the *proper noun* tag, it is still the most challenging tag ($U+L:false$ $U:false$).

The improvement is also effected by language specific characteristics. The English Wiktionary categorizes plurals into *English plurals*. As a result, the *plural nouns* tag holds a significant proportion, 13.31%, of improvement in English. The use of Korean Wiktionary is, meanwhile, limited because the lexical information of verb cannot be retrieved. In Korean, an infinitive verb is composed of a stem and a suffix '다'. Verb stems combine with functional words to form *eojeols* in normal texts. However, only infinitive verbs are described in Wiktionary. For instance, '증가하다' (to augment) is in Wiktionary but its stem '증가하' is not. In this work, we deal with this specific morphological variance by heuristically looking up for ~하다 instead of ~하 on Wiktionary. In the future, we will find a solution to deal with more general morphological variance cases.

5 Conclusions

For further improvement of POS tagging, our future works will focus on the multilingual aspect, i.e., translation information. Because the POS tagsets are quite different in the Brown corpus and Penn treebank [26], we will evaluate our method on the Penn treebank and compare our results with state-of-art results of contemporary works.

Table 5. Language-dependent POS tags occurring in the error analysis

English		Korean		Vietnamese	
NP	Proper noun	NNP	Proper noun	NP	Proper noun
NN\$	Plural noun	NNG	Noun	V	Verb
RB	Adverb	SL	Abbreviation	N	Noun
JJ	Adjective	SN	Number	A	Adjective
NN	Noun	JX	Noun suffix	E	Preposition
VBD	Verb, past tense	VV	Verb, present	C	Conjunction
NP\$	Possessive proper noun	SW	Measure unit		
NPS	Plural proper noun	SH	Hanja		

In this work, we successfully use linguistic knowledge from Wiktionary to improve POS tagging in multiple languages, including English, Korean, and Vietnamese. Because supervised learning frameworks for POS tagging and other lexical disambiguation problems are similar, we believe that problems such as WSD, text chunking, and diacritic restoration will also be benefited. Last but not least, other languages, in particular those having more than one million articles in Wiktionary such as Chinese and French, are worth investigating.

Acknowledgement. This work is partially supported by a scholarship of the Institute of Information Technology Advancement (IITA), Korean Ministry of Information and Communication from March, 2008 to March, 2012. The first author has received a tuition fellowship of PhD course from the University of Ulsan. The authors would like to thank the anonymous reviewers for their constructive comments and suggestions on the paper.

References

1. 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, NAACL 2003, vol. 1, pp. 173–180. Association for Computational Linguistics, Stroudsburg (2003)
2. Spoustova, D.J., Hajic, J., Raab, J., Spousta, M.: Semi-supervised training for the averaged perceptron pos tagger. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2009, pp. 763–771. Association for Computational Linguistics, Stroudsburg (2009)
3. Manning, C.D.: Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In: Gelbukh, A.F. (ed.) CILing 2011, Part I. LNCS, vol. 6608, pp. 171–189. Springer, Heidelberg (2011)
4. Chesley, P., Vincent, B., Xu, L., Srihari, R.: Using verbs and adjectives to automatically classify blog sentiment. In: AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pp. 27–29 (2006)
5. Müller, C., Gurevych, I.: Using Wikipedia and Wiktionary in Domain-Specific Information Retrieval. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 219–226. Springer, Heidelberg (2009)
6. Zesch, T., Muller, C., Gurevych, I.: Using wiktionary for computing semantic relatedness. In: Proceedings of the 23rd National Conference on Artificial Intelligence, AAAI 2008, vol. 2, pp. 861–866. AAAI Press (2008)
7. Meyer, C.M., Gurevych, I.: Ontowiktionary - constructing an ontology from the collaborative online dictionary wiktionary. In: Pazienza, M.T., Stellato, A. (eds.) Semi-Automatic Ontology Development: Processes and Resources. IGI Global, Hershey (2011) (to appear)
8. Medelyan, O., Milne, D., Legg, C., Witten, I.H.: Mining meaning from wikipedia. *International Journal of Human-Computer Studies* 67, 716–754 (2009)
9. Miller, G.A.: Wordnet: a lexical database for english. *Commun. ACM* 38, 39–41 (1995)
10. Navarro, E., Sajous, F., Gaume, B., Prevot, L., Hsieh, S., Kuo, I., Magistry, P., Huang, C.R.: Wiktionary and NLP: Improving synonymy networks. In: Proceedings of the 2009 ACL-IJCNLP Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources, pp. 19–27. Association for Computational Linguistics, Suntec (2009)

11. Meyer, C.M., Gurevych, I.: What psycholinguists know about chemistry: Aligning wiktionary and wordnet for increased domain coverage. In: Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 883–892. Asian Federation of Natural Language Processing, Chiang Mai (2011)
12. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. *Computational Linguistics* 21, 543–565 (1995)
13. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, pp. 1–8. Association for Computational Linguistics (2002)
14. Lee, G.G., Lee, J.H., Cha, J.: Syllable-pattern-based unknown-morpheme segmentation and estimation for hybrid part-of-speech tagging of korean. *Comput. Linguist.* 28, 53–70 (2002)
15. Nguyen, L.M., Xuan, B.N., Viet, C.N., Nhat, M.P.Q., Shimazu, A.: A semi-supervised learning method for vietnamese Part-of-Speech tagging. In: International Conference on Knowledge and Systems Engineering, pp. 141–146. IEEE Computer Society, Los Alamitos (2010)
16. Jiang, W., Huang, L., Liu, Q., Lu, Y.: A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In: Proceedings of ACL-2008: HLT, pp. 897–904. Association for Computational Linguistics, Columbus (2008)
17. Kim, H.: Korean national corpus in the 21st century sejong project (2003)
18. Han, C., Han, N., Ko, E., Palmer, M.: Penn Korean Treebank: Development and Evaluation. In: Proceedings of the Pacific Asian Conference of Language and Computation (2002)
19. Pham, D.D., Tran, G.B., Pham, S.B.: A hybrid approach to vietnamese word segmentation using part of speech tags. In: Proceedings of the 2009 International Conference on Knowledge and Systems Engineering, KSE 2009, pp. 154–161. IEEE Computer Society, Washington, DC (2009)
20. Nguyen, P.T., Vu, X.L., Nguyen, T.M.H., Nguyen, V.H., Le, H.P.: Building a large syntactically-annotated corpus of vietnamese. In: Proceedings of the Third Linguistic Annotation Workshop, pp. 182–185. Association for Computational Linguistics, Suntec (2009)
21. Nguyen, K.H., Ock, C.Y.: Margin perceptron for word sense disambiguation. In: Proceedings of the 2010 Symposium on Information and Communication Technology, SoICT 2010, pp. 64–70. ACM, New York (2010)
22. Nguyen, K.H., Ock, C.Y.: Word sense disambiguation as a traveling salesman problem. *Artificial Intelligence Review* (2011) (online first)
23. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden markov support vector machines. In: Fawcett, T., Mishra, N. (eds.) *ICML*, pp. 3–10. AAAI Press (2003)
24. Kucera, H., Francis, W.N.: *Computational analysis of present-day American English*. Brown University Press, Providence (1967)
25. Ahn, Y.M., Seo, Y.H.: Korean part-of-speech tagging using disambiguation rules for ambiguous word and statistical information. In: Proceedings of the 2007 International Conference on Convergence Information Technology, ICCIT 2007, pp. 1598–1601. IEEE Computer Society, Washington, DC (2007)
26. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.* 19, 313–330 (1993)

Two Stages Based Organization Name Disambiguity

Shu Zhang¹, Jianwei Wu², Dequan Zheng², Yao Meng¹, Yingju Xia¹,
and Hao Yu¹

¹ Fujitsu Research and Development Center

Dong Si Huan Zhong Rd, Chaoyang District, Beijing and 0086, China

{zhangshu, mengyao, yjxia, yu}@cn.fujitsu.com

² School of Computer Science and Technology, Harbin Institute of Technology

No.92, Xidazhi Street, Harbin 150001, China

{jwwu, dqzheng}@mtlab.hit.edu.cn

Abstract. With the rapid growth of user generated media, Twitter has become an important information resource where users share fresh information on any subject. Pursuing on the problem of finding related tweets to a given organization, we propose two stages based organization name disambiguity. Insufficient information and the diversity of organizations are two key problems for this task. We induce multiple types of features to enrich the information of organization to solve the problem of insufficient information. The relationships between tweets and organization, the relationships among tweets are mined in two stages to solve the diversity of organization. Furthermore, we probe the distribution of organization names' ambiguity and its influence to different classifiers. Our experimental results on WePS-3 prove the proposed methods are effective and promising in performing this task.

Keywords: Twitter, name disambiguity, online reputation management.

1 Introduction

With the spread of the World Wide Web, more and more people express their opinions on almost anything in web sites such as forums, blogs and community websites. Twitter is an online social networking and microblogging service, which rapidly gained worldwide popularity, with 200 million users as of 2011¹, generating over 200 million tweets and handling over 1.6 billion search queries per day². It becomes an important information resource. How to analyze sharing information have received considerable attention in research community. There are some researches such as opinion mining, online reputation management, which focus on monitoring user generated media. One of the essential things of these researches is first to get the information which is related to the studied entity, such as product, company, or certain event. For example, a popular brand has a requirement that monitor relevant

¹ <http://www.bbc.co.uk/news/business-12889048>

² <http://blog.twitter.com/2011/08/your-world-more-connected.html>

tweets per day, it is important to filter out the spurious tweets when brand name is ambiguous.

This paper focuses on finding related tweets to a given organization. Assuming that tweets are retrieved by the organization name query, the task is to decide whether each organization name found in each tweet represents the target organization or not. This is a challenging task due to the potential organization name ambiguity. For example, the name of company “Apple” which has a separate meaning refers to one kind of fruit. Filtering spurious name matches is important to effectively detect and analyze relevant information about the organization from the users.

There are two problems: insufficient information and the diversity of organizations. Twitter differs from the traditional user generated media. It allows users to generate each message with no more than 140 characters. Therefore, each tweet contains a very small textual context for disambiguation. Aim to process any organization names but not one or some given organization names, the organization names in training data are different from those in test data. This leads that we could not train a classifier to a certain organization. Furthermore, these organizations may refer to different domains, which makes this task more challenging. To overcome these problems, we propose two stages based organization name disambiguity method. The major contributions of our approach are as follows:

Induce multiple types of features to enrich the information of organization to solve the problem of insufficient information.

Combine supervised and semi-supervised methods in two stages to solve the diversity of organization.

Probe the distribution of organization names’ ambiguity and its influence to different classifiers.

The remainder of the paper is organized as follows: Section 2 describes the related work on name disambiguity. Section 3 gives problem description. Section 4 presents supervised methods to classify tweets. Section 5 introduces semi-supervised method to classify the tweets and two stages strategy. Section 6 gives the experiments and results. Finally section 7 summarizes this paper.

2 Related Work

Online social networks such as Twitter have attracted much interest from the research community. With little information contained in each tweets, it is a challenge for monitoring and analyzing them. WePS-3 Online Reputation Management³ held in 2010, aimed to identify tweets which are related to a given company. It provides standard training and test dataset that enable researchers to carry out and evaluate their methods[1].

For this task, the research of [2] shows the best performance in the evaluation campaign. They adopt support vector machines (SVM) classifier with external resources, including Wordnet, metadata profile, category profile, Google set, and user feedback. To overcome the problem of tweets containing little context information,

³ <http://nlp.uned.es/weps/>

they create several profiles with external resources as a model for each company. Yoshida et al. classify organization names into “organization-like names” or “general-word-like names” [3]. They categorize each query in the first stage, and categorize each tweet in the second stage using the rules customized for each class of queries. Kalmar adopts bootstrapping method to classify the tweets [4]. The research of [5] shows the named entities in tweets are appropriate for certain company names.

Perez-Tellez et al. propose term expansion to the ambiguous words and words which highly co-occur with it [6].

Focus on identifying relevant tweets for social TV, Dan et al. propose a bootstrapping algorithm utilizing a small manually labeled dataset, and a large dataset of unlabeled messages [7].

Our work is different from theirs: supervised method is first adopted to utilize the training data to train a generic classifier. Then semi-supervised method is induced in second stage to mine the relationship between each tweet in test data. Our method aims to utilize both training and test data to improve the accuracy of the performance. Furthermore, we probe an effective way to combine supervised and semi-supervised method to solve organization name disambiguity.

3 Problem Description

Given a set of tweets and an organization name, the goal is to decide if each tweet in the set talks about this organization.

The input information per tweet contains: the tweet identifier, the entity name, the query used to retrieve the tweet, the author identifier and the tweet content. For each organization in the dataset, it gives the organization name and its homepage URL.

The output per tweet is True or False tag corresponding to related or non-related with the given organization. Table 1 shows the examples of tweet disambiguation for the company “Cadillac”.

Table 1. Examples of tweet disambiguation for the company “Cadillac”

	Tweet content	Tag	Description
1	On Sale: 2004 Hotwheels Crank Itz 3/5 Cadillac Escalade	TRUE	about car
2	Update: Cadillac CTS-V vs BMW M5 Performance Testing.....	TRUE	about car
3	#nowwatching cadillac records while I’m finishing my paper	FALSE	a movie
4founded in 1701 by the Frenchman Antoine de la Mothe Cadillac	FALSE	adventurer

In Table 1., it shows the word “Cadillac” in four tweets, which have three different meanings in different context. This task wants to find “Cadillac” with the meaning referring to a certain car brand, therefore no.1 and no.2 tweets are tagged as “TRUE” and others are tagged as “FALSE”.

Fig.1 shows the distribution of ambiguity across company names on WePS-3 training data. That is the ratio of related tweets in each tweet set retrieved by the

organization name query. The ratio ranges from 0 to nearly 1, which shows a great variability of ambiguity. There is a question whether the ratio of the related tweets has any influence on the performance of disambiguity.

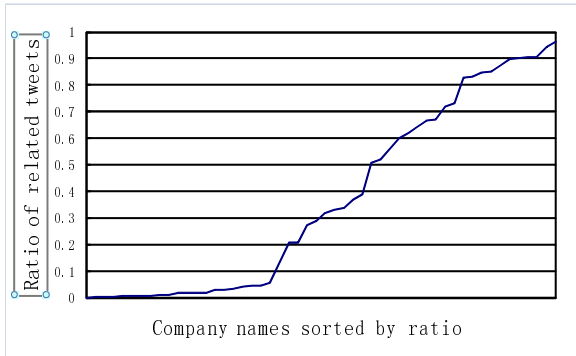


Fig. 1. Ambiguity distribution on WePS-3 training dataset

4 Supervised Classifier

The task is to judge that each tweet in the set is related to the organization or not, therefore we treat it as a classification task. With the training data, we aim to train a classifier to classify different organization names with generic features. It includes three parts: features extraction, representation and classifier selection.

4.1 Feature Extraction

In order to train a general classifier, the selected features should not be too general with the preference to tag tweets as “True”, or too narrow with the preference to tag tweets as “False”. Features could be extracted from two parts: tweets and organization. Here, we aim to expand organization information with external resources. Therefore, we introduce related homepage, related Wikipedia page, and GoogleSet in the following way to get information to represent the organization.

Homepage

It is natural to regard that the organization's web site is indicative to represent the organization itself. Firstly, we crawl through web pages from the homepage in maximum depth of 2. Then all the crawled web pages are preprocessed, including removing HTML tags, filtering stop words, and stemming. Finally, all unigrams and bigrams are chose to represent the organization.

However, some homepages are edited by javascripts or even flash, from which no valuable text could be extracted. At present, we discard these homepages.

Wikipedia page

As a well organized and freely available knowledge, Wikipedia provide some high quality information for some entity. Because lexical ambiguity exists, we utilize

Wikipedia disambiguation page⁴, which provides some candidates for a given entity name. If the wiki-webpage of an entity candidate contains the organization's homepage URL, we believe that this webpage is related to the organization. However, not all the entities could find related wiki-webpage for the limited coverage of wikipedia or homepage URL mismatch. If we can get wiki-webpage, the preprocess step is the same as the homepage mentioned before.

Metadata

Meta tags in HTML page provide high quality keywords to represent its webpage, they are strong indicators to represent organization. However, only a fraction of webpages have this information. The following is an example of meta tags from the "Cadillac company" related webpage:

```
<META NAME="keywords" CONTENT="Cadillac ,CADILLAC, Cadillac, GM,
General Motors Japan General Motors, XLR, SRX, CTS, STS, SEVILLE, servile,
DEVILLE, deville"
```

However, only a fraction of webpages have this information.

GoogleSet

GoogleSet provides similar words with the query words. We utilize this function to enrich organization information with related words. For example, input the organization name "Yale University", associated words "Stanford", "Columbia" are returned. This kind of information is useful, it gives latent semantic category information of the organization at some extent.

URL

URL in homepage or wiki- webpage is also a strong indicator. If the tweet contains the same URL with homepage or wiki-webpage, it is more possible to be related to the organization.

Corresponding to the above types of features organization, we extract unigrams, bigrams words, and URL from tweets as features.

4.2 Representation

The representation of tweets corresponding to given organization is shown as follows:

$$Vector(T_i, O_k) = \{F_1, F_2, \dots, F_n\} \quad (1)$$

Here, T_i is a tweet, O_k is the corresponding organization, F_j is one type of the features described in the above section, and F_j is a set of keywords, such as unigrams, bigrams, or urls. For each feature F_j , the value is computed as follows.

$$Value(T_i, F_j) = \sum_{t_m \in T_i \wedge t_m \in F_j} Wt_m \quad (2)$$

Wt_m is weight of keyword t_m , which could be computed by tf*idf or just given {0,1} value. t_m is the co-occurrence keywords between F_i and tweet T_i . This part is similar with the work of [2].

⁴ [http://en.wikipedia.org/wiki/xxx_\(disambiguation\)](http://en.wikipedia.org/wiki/xxx_(disambiguation)).

4.3 Classifiers

In this stage, we adopt three classical supervised methods to classifying the tweets. They are Maximum Entropy classifier, Support Vector Machine, and Naive Bayes Classifier. We want to testify their performance on this task.

Maximum Entropy Classifier

The classifier is to classify tweets as True or False with the given feature vector. We aim to train a Maximum Entropy Classifier for this task. The principle of Maximum Entropy Model is that the model should maximize entropy, or "uncertainty" with satisfying all the constraints. This is a straightforward idea that just model what is known, and just keep uniform what is unknown. Here, we utilize all features describe above in this classification task. NLTK⁵ tool is used to implement Maximum Entropy Classifier.

Support Vector Machine

Support Vector Machine (SVM) is a popular machine learning approach. Based on the structural risk minimization of statistical learning theory, SVM finds an maximum-margin hyperplane to separate the training examples into two classes. Due to maximum-margin preventing over-fitting in high-dimensional data, SVM usually achieves good performance on a range of tasks.

We use LIBSVM⁶ toolkit to achieve the classification result. RBF kernel function is used and all the other parameters are set to their default values.

Naive Bayes Classifier

The Naive Bayes Classifier is based on Bayesian theorem. Though it is simplicial, Naive Bayes Classifier has been proved very effective for text categorization. We use the Naive Bayes Classifier provided by the NLTK toolkit.

5 Semi-supervised Classifier

Focus on processing any organization names but not one or some given organization names, the organization names in training data are different from those in test data. However, supervised classifier trained on training data is a generic classifier, it is not very effective to a certain organization name in the test data. In order to utilize or mine specific information for a certain organization, we adopt one of the classic semi-supervised methods Label Propagation (LP) to classify tweets. In this stage, we aim to utilize the relationship among tweets in each organization name related tweet set.

5.1 Label Propagation Algorithm

LP algorithm has achieved good performance in many applications, such as noun phrase anaphoricity in coreference resolution [8], word sense disambiguation[9] and relation extraction [10]. Compared with Bootstrapping algorithm, which is based on a local consistency assumption, LP algorithm is based on a global consistency assumption., and can effectively capture the natural clustering structure in both the labeled and unlabeled data to smooth the labeling function.

⁵ <http://www.nltk.org/>

⁶ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Label Propagation Algorithm is propagating labels from the labeled vertices (served as seeds) to all the unlabeled ones through the weighed edges in a graph [11]. Larger edge weight will make propagate easier, which means that if the similarity of two nodes is high, they tend to have the same label.

Label distributions are spread across a graph $G=\{V,E,W\}$, where V is the set of n nodes, E is a set of m edges and W is an $n*n$ matrix of weights with W_{ij} as the weight of edge (i, j) .

The seeds selection and graph construction are important for Label Propagation Algorithm. Seeds selection is based on the first stage supervised method. Supervised classifier gives a confidence value for each tweet which is classified to True or False. Therefore, we choose N ($N=5$) True and False tweets tagged by supervised classifier as seeds according to its confidence value.

Each tweet is treated as node, the edge is constructed if two tweets have co-occurrence words, its weight is computed by Cosine similarity. Here, we make use of JUNTO Label Propagation toolkit⁷.

5.2 Two Stages Strategy

The target is to combine supervised and semi-supervised methods to mine both training and test data for this specific task. In this paper, we try to select some organizations tweet set to be classified by semi-supervised classification, not all organizations. The selection process is based on the ratio of “True” tweets in the given organization test set tagged by supervised classifier.

$$Ratio(O_k) = Num(True) / Num(O_k) \quad (3)$$

Here, O_k is the organization, $Num(True)$ is number of tweets tagged as True by supervised classifier, $Num(O_k)$ is the number of tweets for the given organization. If $Ratio(O_k)$ is higher than the threshold, semi-supervised classifier is applied to classify the tweets once more for organization O_k . The other organizations tweets set do not need to be classified by LP. The ratio is set at 0.5 experientially.

6 Experiments

We have conducted experiments on the WePS-3 task 2 data. The training data contain about 50 organizations with about 400 tweets for each organization. The test data also contain about 50 organizations. There is no intersection between training and test data.

The task is to classify the tweets related or non-related to the given organization, it belongs to classification task. Therefore, we measure the performance by accuracy, precision, recall and F-measure.

First, we testify the performance of difference methods on this performance. Table 2 shows their performance.

⁷ <http://code.google.com/p/junto/>

Table 2. Performance on organization name disambiguity

	ACC	P+	R+	F+	P-	R-	F-
ME	0.6952	0.6362	0.4284	0.4139	0.6249	0.8109	0.6503
NB	0.7158	0.6351	0.4649	0.4535	0.6390	0.7853	0.6503
SVM	0.6675	0.6882	0.3380	0.3318	0.6166	0.8479	0.6496
LP	0.4399	0.4366	0.9973	0.5280	0.7709	0.0169	0.0314
ME+LP	0.7160	0.6334	0.4620	0.4298	0.6176	0.7566	0.6291
NB+LP	0.7469	0.6257	0.5225	0.4789	0.6674	0.6855	0.6058

Table 2 gives the performance of ME, NB and SVM classifier. F+ is the F-measure of tweets tagged as True. Three supervised methods perform better for tweets tagged as False (F+) than that of True (F+). Performance of SVM classifier is a little lower, this may be caused by the generic features chosen. Only LP algorithm performs worse, however it has a good recall for tweets tagged as True. There is a phenomenon that supervised classifiers tend to tag tweets as False, on the other hand semi-supervised classifier tends to tag tweets as True. Therefore the combination of these two methods is a feasible way to make up for their shortcoming. The performance shows that the two stages methods have high values on ACC, it is a balance between True and False tweets. LP algorithm need seed selection, SVM classifier does not give the confidence value of each tweet classification result, therefore there is no experiments combining SVM with LP.

Compared our systems with WePS participant system, our systems performance are less than the system of [2]. Its accuracy value is 0.83. Its system introduces manually constructed UserFeedback profile. With only homepage as features, its F-measure of related tweets is about 0.3. Different from theirs, our systems are all automatically.

Second, we analyze the performance of supervised methods on different ambiguity distribution of tweets. Table 3 gives the results on ACC value.

Table 3. Performance on different organization name ambiguity

	[0~0.2)	[0.2~0.8)	[0.8~1.0]
ME	0.8708	0.6126	0.5619
NB	0.8658	0.6375	0.6173
SVM	0.8791	0.5789	0.4850

Table 3 shows that the performance of supervised classifiers on different ratio of related tweets. The ratio of related tweets in each tweet set retrieved by the organization name query really has effect on the performance of classifiers. Supervised classifiers have better performance in the range of [0~0.2) when the tweets set has more unrelated tweets. This proves that the two stages strategy is feasible to utilize supervised methods classify the set whose has more unrelated tweets.

7 Conclusion

In this paper, we probe into the problem of finding related tweets to a given organization. This task is more challenging caused by insufficient information and the diversity of organizations. Induce multiple types of features to enrich the information of organization to solve the problem of insufficient information. Combine supervised and semi-supervised methods in two stages to classify the tweets to solve the diversity of organization. Our experimental results on WePS-3 are primary and encouraging, they prove the proposed techniques are effective in performing the task.

There is still a gap needed to be filled by further improving these techniques. For example, probe the way of the combination of supervised and semi-supervised methods.

References

1. Amigó, E., Artiles, J., Gonzalo, J., Spina, D., Liu, B., Corujo, A.: WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In: 3rd Web People Search Evaluation Workshop (2010)
2. Yerva, S.R., Miklós, Z., Aberer, K.: It was Easy, when Apples and Blackberries were only Fruits. In: 3rd Web People Search Evaluation Workshop (2010)
3. Yoshida, M., Matsushima, S., Ono, S., Sato, I., Nakagawa, H.: ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In: 3rd Web People Search Evaluation Workshop (2010)
4. Kalmar, P.: Bootstrapping Websites for Classification of Organization Names on Twitter. In: 3rd Web People Search Evaluation Workshop (2010)
5. García-Cumbreras, M.A., García-Vega, M., Martínez-Santiago, F., Peréa-Ortega, J.M.: SINAI at WePS-3: Online Reputation Management. In: 3rd Web People Search Evaluation Workshop (2010)
6. Perez-Tellez, F., Pinto, D., Cardiff, J., Rosso, P.: On the Difficulty of Clustering Microblog Texts for Online Reputation Management. In: 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT (2011)
7. Dan, O., Feng, J., Davison, B.D.: A Bootstrapping Approach to Identifying Relevant Tweets for Social TV. In: 5th International AAAI Conference Weblogs and Social Media (2011)
8. Zhou, G.D., Kong, F.: Global Learning of Noun Phrase Anaphoricity in Coreference Resolution via Label Propagation. In: Empirical Methods in Natural Language Processing, pp. 978–986 (2009)
9. Niu, Z.Y., Ji, D.H., Tan, C.T.: Word Sense Disambiguation Using Label Propagation Based Semi-Supervised Learning. In: 43rd Annual Meeting on Association for Computational Linguistics, pp. 395–402 (2005)
10. Chen, J.X., Ji, D.H., Tan, C.T., Niu, Z.Y.: Relation Extraction Using Label Propagation Based Semi-supervised Learning. In: 21st International Conference on Computational Linguistics and 44th Annual Meeting on Association for Computational Linguistics, pp. 129–136 (2006)
11. Zhu, X., Ghahramani, Z.: Learning from Labeled and Unlabeled Data with Label Propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)

Optimizing CRF-Based Model for Proper Name Recognition in Polish Texts

Michał Marcińczuk and Maciej Janicki

Wrocław University of Technology, Wrocław, Poland
michal.marcinczuk@pwr.wroc.pl, macjan@o2.pl

Abstract. In this paper we present several optimizations introduced to Conditional Random Fields-based model for proper names recognition in Polish running texts. The proposed optimizations refer to word-level segmentation problems, gazetteers incompleteness, problem of unambiguous generalization features, feature construction and selection, and finally recognition of common proper names on the basis of external sources of knowledge. The problem of proper name recognition is limited to recognition of person first names and surnames, names of countries, cities and roads. The evaluation is performed in two ways: a single domain evaluation using 10-fold cross validation on a Corpus of Stock Exchange Reports and a cross-domain evaluation on a Corpus of Economic News. An additional corpus of Wikipedia articles, namely InfiKorp is used in the feature selection. Finally, we evaluate three configurations of proposed modifications. The top configuration improved the final result from 94.53% to 95.65% of F-measure for single domain and from 70.86% to 79.63% for cross-domain evaluation.

Keywords: Named Entity Recognition, Proper Name Recognition, Machine Learning, Conditional Random Fields, Gazetteers, Classifier Ensemble, Polish.

1 Introduction

Recognition of proper names consists of identification and categorization of multiword expressions in text that are unique (to some extent) names of real or fictional entities. Effective recognition of proper names is very important in many tasks from the natural language processing (NLP) domain, i.e. information extraction from medical documentation [1], text anonymization [2], machine translation [3] or forensic linguistics [4].

Proper names are subpart of named entities and the task of Proper Name Recognition (PNR) is commonly referred as a task of Named Entity Recognition (NER). Named entities also refer to numerical expressions (date, time, numbers, etc.), definite descriptions and in some cases also to noun phrases [5].

The recent successful approaches to NER are based on application of Conditional Random Fields (CRF), e.g. a method for sequence tagging. This method has been already applied to English [6], Polish [7,8], Bulgarian [9], Arabic [10]

and many other languages. The advantage of CRF over generative models like Hidden Markov Models (HMM) is that CRF can make use of additional features attached to a sequence of words. Construction of new features and selection of the best subset of all possible features must be done in order to obtain optimal results.

CRF ([14]) is undirected graph-based model trained to maximize a conditional probability $Pr(y|x)$, where y is a linear sequence of labels from a fixed set, and x is a sequence of words and their corresponding features. The label set contains $2 * n + 1$ distinct labels, where n is the number of PN categories. For every PN category P there are two labels: **B- P** (beginning) and **I- P** (intermediate); and one additional label **O** (out of entity). CRF has to estimate a labelling Y from the observation sequence X .

In this paper we present modifications to the existing model for proper names recognition presented in [8]. The model is trained to recognize 5 common categories of proper names in Polish texts, i.e. person first names and surnames, names of countries, cities and roads. In Section 2 we describe the evaluation process, corpora used in the evaluation and baseline for that corpora. In Section 3 we define and evaluate the modifications and in the Section 4 we evaluate the generality of the modifications.

2 Evaluation and Baseline

In the evaluation process we used three corpora: Corpus of Stock Exchange Reports (CSER) and Corpus of Economic News (CEN) from [8] and a part of InfiKorp¹ containing Wikipedia articles (IKW). The obtained results were compared to the results reported in [8]. 10-fold cross validation on CSER was used to evaluate and to select the best modifications. In addition, the feature selection was performed on IKW in order to shorten the processing time. To evaluate the generality of the selected modifications we used CEN to perform a cross-domain evaluation — the model was trained on CSER and tested on CEN.

The baseline model (described in [8]) was a CRF model utilizing 38 types of features (19 orthographic, 5 morphological, 10 gazetteer- and 4 wordnet-based features). For every feature a window of 2 preceding, a current and 2 following tokens was used. The gazetteer-based features included 5 gazetteers of proper names gathered semi-automatically from the Internet and 5 gazetteers of keywords obtained from wordnet for Polish. CRF++² package was used to train the models. The configuration of CRF++ consists of the cut-off threshold for the features set to 5 and hyper-parameter set to 1.

Comparing to [8], some corrections were introduced to CSER and the baseline experiments were repeated. The corrections consist mainly reduction of inconsistency in the annotation — verification of names which were annotated as

¹ The corpus is being developed within SyNaT project, financed by NCBiR NrU.: SP/I/1/77065/10 (<http://www.synat.pl>)

² CRF++ home page: <http://crfpp.sourceforge.net/>

Table 1. Baseline results for proper names recognition

Corpus	Precision	Recall	F-measure
<i>10-fold Cross Validation on CSER</i>			
CSER from [8] (B1.1)	96.20%	90.00%	92.53%
CSER after correction (B1.2)	96.79%	92.38%	94.53%
<i>Cross-domain Evaluation on CEN trained on CSER</i>			
CSER from [8] (B2.1)	92.88%	53.29%	67.72%
CSER after correction (B2.2)	92.02%	57.61%	70.86%

different categories of names. The original (reported in [8]) and repeated baselines are presented in Table 1. The correction of CSER improved the results by 2% of F-measure for CSER and by 3% of F-measure for cross-domain evaluation on CEN.

3 Model Modifications

3.1 Text Segmentation

In [11] the authors report that errors in word-level and sentence-level segmentation have negative impact on proper name recognition. Authors used *TaKIPI* to tokenize the text and pointed out two common errors: missing segmentation of words linked by a hyphen and abbreviations attached to names. In both cases a fine-grained tokenization might fix the problem. Such tokenization is provided by *maca* — a tool for text tokenization and morphological analysis [12]. We have tagged the corpora again with *maca* and repeated the evaluation. The fine-grained tokenization improved slightly the recall by 0.3% and F-measure to 94.66%. In the following experiments we used corpora tokenized with *maca* instead *TaKIPI*.

3.2 Gazetteers Extension

Gazetteers are a very helpful source of general knowledge when processing documents related to real world, like our corpora. We have used gazetteers from [8] (i.e., gazetteers of first names, surnames, names of countries, cities and roads). The gazetteers contain many nominal name forms, but many inflected forms are missing. To fill this gap we applied two procedures to gather the inflected forms:

Table 2. Evaluation of different word-level segmentations

Tokenizer	Precision	Recall	F-measure
TaKIPI (B1.2)	96.79%	92.38%	94.53%
maca (B1.3)	96.74%	92.68%	94.66%

Table 3. Comparison of *base* and *extended* gazetteers

Gazetteers	<i>First names</i>	<i>Surnames</i>	<i>Countries</i>	<i>Cities</i>	<i>Roads</i>	TOTAL
Base	22 435	371 379	1 867	77 873	40 859	514 413
Extended	46 351	371 379	4 086	152 543	62 106	636 465

1. **Extraction from large text corpora.** We have taken a large corpora of texts from Internet and tagged them with TaKIPI with Guesser (Guesser tries to guess a base form and a morphological tag for *out of vocabulary* words). Then, we have extracted all pairs of word forms with their base form, and grouped them by the base forms. Next, for every group we checked whether the given base form was present in the gazetteers. If so, then we added all word forms from that group to the gazetteer as inflected forms of the given name. This procedure was applied only to single-word names.
2. **Extraction from dictionary of inflections *sjp-odm***³. The *sjp-odm* dictionary contains ca. 190 000 base forms with their inflections. We have selected all words starting from a capital letter. Then, for every base form we obtained PN categories, to which given word was assigned in our gazetteers (i.e., first name, surname, country, city and road). Finally, we have selected words, that were unambiguous among PN categories and their inflected forms to the gazetteers. This way we have extended all categories except surnames. The manual verification showed that many of the new inflections were in fact inflections of other categories.

By applying these two procedures we have gathered about 125 000 new inflected proper name forms. The detailed results are presented in Table 3. We are aware that this procedure is not optimal and many incorrect inflected forms might have been added to the gazetteers. We did not evaluate the automatic extension itself, but we evaluated the impact of the extension on PNR. The extended gazetteers increased the recall from 92.68% to 93.84% and keep the same level of precision (see Table 4). In the following experiments we used extended gazetteers.

Table 4. Evaluation of extended gazetteers

Gazetteers	Precision	Recall	F-measure
Base (B1.3)	96.74%	92.68%	94.66%
Extended (B1.4)	96.75%	93.84%	95.27%

3.3 Features Modification

Gazetteer-based Features. We have observed a common error of dividing long names into several shorter ones. For example, a country name

³ Available at <http://www.sjp.pl/slownik/odmiany/> on GPL, LGPL or CC SA.

“Stany Zjednoczone Ameryki Północnej” (Eng. *United States of North America*) was recognized as two separate names: “Stany Zjednoczone” (Eng. *United States*) and “Ameryki Północnej” (Eng. *of North America*). The problem was caused by the way how the gazetteer-based features were encoded.

The basic concept of gazetteer-based features was following (cf. [8]): a single gazetteer-based feature is associated with one gazetteer. If a sequence of tokens is found in the gazetteer, then the first word in the sequence is set as B and the other as I. If word is not a part of any gazetteer entry it is set to 0. If there are two nested sequences found in the gazetteer both of them are marked — first, the longer one and then the nested one.

For the above example, the nested sequences present in the gazetteer resulted in the following sequence of values B I B I. The sequence of values could be interpreted as two consecutive or nested proper names, so the interpretation was ambiguous. We decided to encode only the longest proper names and do not encode the nested names. The modification fixed the problem and the results were improved to 95.44% of F-measure (see Table 5).

Table 5. Evaluation of feature modifications on CSER

Configuration	Precision	Recall	F-measure
Baseline (B1.4)	96.79%	92.38%	94.53%
Gazetteer-based Features	96.92%	94.01%	95.44%
Wordnet-based Features	96.70%	93.79%	95.23%
Both modifications (B1.5)	96.83%	94.12%	95.46%

Wordnet-base Features. The basic concept of wordnet-base features (cf. [8]) was to use word synonym and hyperonyms as word features. The features were generated on the basis of plWordnet — a wordnet for Polish. As there is no tool for word sense disambiguation for Polish the ambiguous words had to be solved somehow. The partial solution was to gather all potential hyperonyms (or synonyms) for all senses and choose the first word in alphabetical order.

Using this concept we observed many wrong generalizations for ambiguous words. For example, the Polish word “członek”, which means both *member* and *limb*, was generalized to “part of the body” in a phrase “Członek Rady Nadzorczej” (Eng. “Supervisory Board Member”). We decided to use the generalization only for unambiguous words, i.e. words, for which every sense of the word can be transformed to the same synonym or hyperonym. In other cases, the generalization left the word unchanged. The modification fixed the problem but also reduced the number of generalized words. However, the result was improved to 95.23% of F-measure (see Table 5).

Both Features. Combination of changes introduced to gazetteer-based and wordnet-base features also improved the results, but only slightly better than

Table 6. Evaluation of new features

Configuration	Precision	Recall	F-measure
Baseline (B1.5)	96.83%	94.12%	95.46%
Road #1	96.83%	94.10%	95.45%
City #1	96.73%	94.05%	95.38%
Person #1	96.86%	94.15%	95.48%
Person #2	96.76%	94.12%	95.42%

gazetteer-based feature modification. The final result for both changes was 95.46%. In the following experiments we used both modifications.

3.4 Feature Construction

In the next step we tried to construct new features for CRF. The feature templates were based on rules created for proper names recognition described in [13]. We have selected those rules that were present in CSER corpus. Every rule was encoded as a combination of existing features. We have evaluated 4 templates (features):

- Road#1: `road_prefix[-1]/road_nam[0]` — road name after a road prefix,
- City#1: `city_nam[0]/pattern[1]/pattern[2]/country_nam[3]` — city name followed by some punctuation marks and a country name,
- Person#1: `base[-1]/first_nam[0]/last_nam[1]/base[2]/last_nam[3]` — person name after a certain word containing first name, surname and maiden name,
- Person#2: `base[-2]/first_nam[-1]/pattern[0]/base[1]` — person name between certain words with known first name and unknown surname.

Introduction of new features does not have noticeable impact on the results (see Table 6). Only the *Person #1* rule insignificantly improved the results by 0.02% of F-measure. The low impact might be caused by the fact that the rules operate on narrow context and CRF might have already learned the close context patterns.

3.5 Feature Selection

To measure the features relevance we have calculated Information Gain (IG) for every feature used by CRF model. The total number of features was 172. The range of IG values was from 0.46 for `orth+0` to 0.000009 for `pesron_suffix-2` (see Table 7 — we presented only the top and the bottom values of IG).

We grouped the bottom features according to IG into three groups: (1) $< 10^{-5}$, (2) $< 10^{-4}$ and (3) $< 10^{-3}$. Then we evaluated every group on IKW and CSER by removing the features from the set of all features. The results are presented in Table 8. For IKW the feature reduction does not improve the

Table 7. Information Gain for top and bottom features

Top 10 features		Bottom features	
IG	Feature	IG	Feature
0.45929656	orth+0	(3) 0.00094167	road_prefix+2
0.45623956	base+0	0.00072422	person_prefix+2
0.44374859	prefix-4+0	0.00060709	country_prefix-1
0.42121654	suffix-4+0	0.00059445	person_noun+2
0.40814769	prefix-3+0	↓ 0.00035950	road_prefix+1
0.35655636	suffix-3+0	(2) 0.00003794	country_prefix+2
0.30563528	prefix-2+0	0.00001289	person_suffix+1
0.28395843	orth-1	0.00001239	person_suffix+0
0.26671994	orth-2	↓ 0.00001046	person_suffix-1
0.26541514	base-1	(1) 0.00000948	person_suffix+2
		0.00000925	person_suffix-2

results at all. On CSER only for the second group of features we noticed a small improvement by 0.05% of F-measure. We observed that the first two groups of features completely removes the `person_suffix` feature. The feature indicates words that can appear immediately after person name but in CSER the words do not appear and the feature is useless.

Table 8. Evaluation of feature selection using IG measure on IKW and CSER

Configuration	Features	IK			CSER		
		P	R	F	P	R	F
Baseline	172	73.51%	53.10%	61.66%	96.83%	94.12%	95.46%
Without (1)	170	73.58%	52.56%	61.32%	96.83%	94.08%	95.43%
Without (2)	166	73.76%	52.29%	61.20%	96.88%	94.17%	95.51%
Without (3)	161	73.96%	52.83%	61.64%	96.83%	94.08%	95.43%

3.6 Feature Reduction

In this approach, feature selection is done top-down. The reduction is done as follows: we start with a rich set of features, each of them taken in a -2:-1:0:1:2 window — for current token, two preceding and two next. Then, for each feature separately, we perform a 4-fold cross validation on the IKW with this feature limited to a window -1:0:1, 0 (for current token only) and completely deleted, while other features are left without changes. We compare results for each feature in each window and perform the reduction, that makes the best result. Then we use the obtained configuration as base for the next iteration. This procedure was repeated as long as any of the possible reductions improve the results.

Table 9. Feature reduction results per iteration in IKW

No	Feature	Reduction	Precision	Recall	F-measure
0	—	—	73.51%	53.10%	61.66%
1	number	delete	75.66%	54.45%	63.32%
2	case	0	77.69%	54.45%	64.03%
3	ctag	delete	77.78%	54.72%	64.24%
4	class	-1:0:1	78.46%	54.99%	64.66%
5	suffix-2	delete	78.33%	55.53%	64.98%
6	pattern	0	78.79%	56.06%	65.51%
7	city_nam	0	78.95%	56.60%	65.93%
8	suffix-3	0	78.60%	57.41%	66.36%
9	base	0	79.18%	57.41%	66.56%
10	hyp1	delete	79.48%	57.41%	66.67%

Table 10. Feature reduction verification on the CSER corpus

No	Feature	Reduction	Precision	Recall	F-measure
Baseline (B1.5)			96.83%	94.12%	95.46%
3	ctag	delete	96.84%	94.29%	95.55%
6	pattern	0	96.72%	94.46%	95.58%
10	hyp1	delete	96.67%	94.36%	95.50%

Table 9 shows results of this process. For each iteration, it shows which feature was selected to be reduced, which reduction (to which window) was performed and what were the evaluation results after the reduction. The first row (“zero” iteration) shows results before the whole process.

After the reduction process performed on IKW, we used a 10-fold cross validation on the CSER corpus to verify the effects of feature reduction. We have observed slight improvement. Table 10 shows results before first and after last iteration, as well as for the reductions with the best precision and F-measure, respectively.

3.7 Post Processing

Unambiguous Gazetteer Look-up is used to recognize known and unambiguous names — names that are present in the gazetteers and are assigned to only one category. This approach is motivated by an observation that there are many inflected forms in the gazetteers that are unambiguous among categories. Table 11 presents the results for the chunker itself. We selected the most unambiguous categories, i.e. names of countries and cities, and applied to the CRF model. The results for chunker combined with CRF is presented in Table 12.

Table 11. Evaluation of unambiguous dictionary chunker

	<i>first name</i>	<i>surname</i>	<i>country</i>	<i>city</i>	<i>road</i>	<i>Total</i>
Precision	56.63%	63.52%	98.97%	93.22%	42.46%	77.15%
Recall	32.13%	14.68%	60.97%	61.91%	54.04%	48.58%
F-measure	41.00%	23.85%	75.46%	74.41%	47.56%	59.62%

Table 12. Evaluation of post-processing of CRF

Annotation	Precision	Recall	F-measure
Baseline (B1.5)	96.83%	94.12%	95.46%
CRF + dictionary chunker	95.26%	95.74%	95.50%
CRF + heuristic chunker	96.66%	94.50%	95.57%

Pattern Matching is based on a set of heuristics constructed on the basis of grammars for named entity recognition presented in [13]. The heuristics annotate names that are present in the gazetteers and appear in a defined context (for example *road name* after *road prefix*, or *person first name* and *surname* after *person noun*). The evaluation of the heuristics itself is presented in Table 13. First names, surnames and city names obtained very high precision over 95% but very low recall. Only for road names we got relatively lower precision below 80%. The analysis of the results showed that many false positives were caused by partial matching of long road names. In most cases those names are correctly recognized by the statistical model. Thus, we decided to make union of chunks recognized by the statistical model and heuristic chunker, and discard all nested annotations of the same type. The combined chunkers obtained F-measure 95.57% comparing to 95.46% for baseline (see Table 12). The small improvement shows that most of the names are already recognized by the statistical model.

Table 13. Evaluation of heuristic chunker

	<i>first name</i>	<i>surname</i>	<i>country</i>	<i>city</i>	<i>road</i>	<i>Total</i>
Precision	96.15%	95.24%	0.00%	99.56%	86.62%	86.62%
Recall	3.62%	2.91%	0.00%	11.26%	73.99%	13.26%
F-measure	6.97%	5.64%	0.00%	20.23%	75.81%	22.99%

3.8 Final Configuration

Finally, we evaluated the combination of proposed modifications. The base configuration for all tested combinations included: (a) tokenization done with *maca* (see Section 3.1), (b) extended gazetteers (see Section 3.2) and (c) modified

Table 14. Evaluation of configurations (10-fold CV on CSER)

Annotation	Precision	Recall	F-measure
Baseline (B1.5)	96.83%	94.12%	95.46%
CRF & chunkers	95.08%	96.09%	95.58%
CRF #1	95.02%	96.28%	95.65%
CRF #2	95.08%	96.07%	95.57%

gazetteer-based and wordnet-based features (see Section 3.3). The evaluation followed 10-fold cross validation on CSER. We have tested three configurations:

CRF & chunkers includes only gazetteer-based and heuristic-based chunkers (all features remain as in the base configuration).

CRF #1 includes those modifications which obtained F-measure above the baseline. This includes: (a) new feature Person#1, (b) feature reduction for iteration 6 and (c) gazetteer-based and heuristic-based chunkers.

CRF #2 includes those modifications which obtained precision above the baseline combined with chunkers. If the sets of chunks recognized by high-precision CRF and chunkers are disjoint, the final results might be improved. This configuration includes: (a) new feature Person#1, (b) feature reduction for iteration 3, (c) gazetteer-based and heuristic-based chunkers and (d) wordnet-based features are discarded.

The results of final configurations evaluated on a single domain are presented in Table 14. Comparing F-measure, all proposed configurations performed better than the baseline configuration. The best improvement was obtained for CRF #1, i.e. combination of F-measure-oriented optimizations, that improved the F-measure from 95.46% to 95.65%.

4 Cross-Domain Evaluation

To verify the generality of introduced changes and final configurations proposed in Section 3.8 we evaluated them on another corpus, namely CEN — the model was trained on whole CSER and evaluated on CEN. Comparing F-measure, all proposed configurations performed better than the baseline configuration. The highest improvement was obtained for CRF #2 that was optimized on precision, which was slightly better than CRF #1.

In the cross-domain evaluation we observed the same tendency as in the cross-validation on a single domain — high increase of recall and small decrease in precision. Comparing the results on the level of PN categories we observed improvement for every category (see Table 16). The highest improvement was obtained for city names — almost 15%, as most of the applied improvements affected this recognition of city names.

Table 15. Cross domain evaluation on CEN

Configuration	Precision	Recall	F-measure
Baseline (B2.2)	92.02%	57.61%	70.86%
CRF & chunkers	91.32%	69.86%	79.16%
CRF #1	91.55%	70.32%	79.54%
CRF #2	91.44%	70.53%	79.63%

Table 16. Detailed comparison of F-measure for the baseline (B2.2) and the best configuration (CRF #2)

	<i>first name</i>	<i>surname</i>	<i>country</i>	<i>city</i>	<i>road</i>	<i>Total</i>
Baseline (B1.5)	76.14%	61.76%	80.20%	58.68%	18.18%	70.86%
CRF #2	84.13%	66.51%	90.42%	73.27%	28.95%	79.63%

5 Summary

In the paper we presented several optimizations applied to the CRF-based model for proper names recognition in Polish texts. The optimization includes: fine-grained text segmentation on word level, extension of gazetteers, modification of gazetteer-based and wordnet-based features, construction of new features, reduction of existing features and application of external sources as post processing. The fine-grained tokenization improved the recognition of names that were incorrectly treated as parts of longer segments. The automatic extension of gazetteers improved recognition of common, inflected names. The modification of gazetteer-based features improved the recognition of long proper names that were incorrectly recognized as a sequence of shorter names. The modification of wordnet-based features that are used to generalize the text reduced the noise in the single-domain evaluation but does not improve the results in the cross-domain evaluation due to low generalization impact. Most of the new features constructed on the basis of grammars for named entity recognition does not improve the results significantly, as the CRF model might already have learned these patterns. The feature selection has better impact in the cross-domain evaluation than in the single-domain evaluation.

The combination of proposed optimization improved the final result in the single-domain evaluation from 94.54% to 95.63% of F-measure. The generality of the selected optimization was tested on another corpora, on which the result was improved from 70.86% to 79.63% of F-measure.

References

1. Mykowiecka, A., Kupść, A., Marciniak, M., Piskorski, J.: Resources for Information Extraction from Polish texts. In: Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, (LTC 2007), Poznań, Poland, October 5-7 (2007)

2. Graliński, F., Jassem, K., Marcińczuk, M.: An Environment for Named Entity Recognition and Translation. In: Màrquez, L., Somers, H. (eds.) *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, Barcelona, Spain, pp. 88–95 (2009)
3. Graliński, F., Jassem, K., Marcińczuk, M., Wawrzyniak, P.: Named Entity Recognition in Machine Anonymization. In: Kłopotek, M.A., Przepiorkowski, A., Wierchoń, A.T., Trojanowski, K. (eds.) *Recent Advances in Intelligent Information Systems*, pp. 247–260. Academic Publishing House Exit (2009)
4. Marcińczuk, M., Zaśko-Zielińska, M., Piasecki, M.: Structure Annotation in the Polish Corpus of Suicide Notes. In: Habernal, I., Matoušek, V. (eds.) *TSD 2011*. LNCS, vol. 6836, pp. 419–426. Springer, Heidelberg (2011)
5. ACE (Automatic Content Extraction) English Annotation Guidelines for Entities. Linguistic Data Consortium, LDC (2008)
6. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: *Seventh Conference on Natural Language Learning*, CoNLL (2003)
7. Mykowiecka, A., Waszczuk, J.: Semantic Annotation of City Transportation Information Dialogues Using CRF Method. In: Matoušek, V., Mautner, P. (eds.) *TSD 2009*. LNCS, vol. 5729, pp. 411–418. Springer, Heidelberg (2009), doi:10.1007/978-3-642-04208-9_56
8. Marcińczuk, M., Stanek, M., Piasecki, M., Musiał, A.: Rich Set of Features for Proper Name Recognition in Polish Texts. In: *Proc. of the S&IIS 2011*, Poland (2011)
9. Georgiev, G., Nakov, P., Ganchev, K., Osenova, P., Simov, K.: Feature-Rich Named Entity Recognition for Bulgarian Using Conditional Random Fields. In: *Proceedings of the International Conference RANLP 2009*, pp. 113–117. Association for Computational Linguistics, Borovets (2009)
10. Benajiba, Y., Rosso, P.: Arabic Named Entity Recognition using Conditional Random Fields. In: *Proc. Workshop on HLT & NLP with in the Arabic World* (2008)
11. Marcińczuk, M., Piasecki, M.: Statistical Proper Name Recognition in Polish Economic Texts. In: *Control and Cybernetics* (2011)
12. Radziszewski, A., Śniatowski, T.: Maca: a configurable tool to integrate Polish morphological data. In: *Proceedings of Free RBMT 2011*, Barcelona, Spain (2011)
13. Piskorski, J.: Extraction of Polish named entities. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, LREC 2004 (ELR 2004), pp. 313–316. ACL, Prague (2004)
14. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: *Proceedings of the 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, NAACL 2003, vol. 1, pp. 134–141. Association for Computational Linguistics, Stroudsburg (2003)

Methods of Estimating the Number of Clusters for Person Cross Document Coreference Task

Octavian Popescu and Roberto Zanolì

{popescu, zanolì}@fbk.eu

Abstract. Knowing the number of different individuals carrying the same name may improve the overall accuracy of a Person Cross Document Coreference System, which processes large corpora and clusters the name mentions according to the individuals carrying them. In this paper we present a series of methods of estimating this number. In particular, an estimation method based on name perplexity, which brings a large improvement over the baseline given by the gap statistics, is instrumental in reaching accurate clustering results because not only it can predict the number of clusters with a very good confidence, but also it can indicate what type of clustering method works best for each particular name.

Keywords: cross document coreference, k-estimation, (p, γ) statistics, gap statistics.

1 Introduction

Usually, in large collections of documents, like corpora made out of pieces of news from a newspaper, web pages etc., many persons are mentioned. The majority of these mentions are ambiguous, because the same name, or same phrases denoting a person, could refer to different individuals and moreover, the same individual could be mentioned under different names. The Person Cross Document (PCDC) task is the Natural Language Processing (NLP) task of identifying all the mentions to persons in a given corpus and to cluster these mentions according to the individual they refer to (Grishman 1994).

A possible application of a PCDC system may be a searching engine whose response to a name query is the list of different individuals carrying that name with links to the most informative documents for each individual. To exemplify, let us consider the result returned by the most common search engines for a query like “George W. Bush”. The answers consist of pages that refer mostly to the 44th USA president, at least in the top ranked 300 pages. A careful examination reveals that there are other individuals carrying this name, for example a remarkable engineer. However, even the response to a query like “George W. Bush engineer” is a set of pages referring to the 44th president of USA in which the word engineer appears in a more or less surprising ways. In the same vein, if one looks for the address of an individual named “Michael Jackson”, other than the pop star, one finds out that the search engines are of little help, as the responses refer overwhelmingly to the pop star.

Another problem, somehow representing the opposite of the one above, appears when the same individual is involved in many different things, like in the case of a politician who is also active in sport or the entertainment business, for example. In this case, it is difficult to realize that there is just one person mentioned in all the documents, as, in general, the documents tend to keep the topics separate.

For the above application of PCDC to search engines which clearly mark different individuals, it is critical to correctly compute the number of different individuals carrying the same name. We call this number the *k*-number, somehow alluding to the *K*-neighborhood clustering method. The experience accumulated in PCDC contests, WePS-1, WePS-2 (Artiles et al. 2007) has shown that while the PCDC systems are rather good at determining all the mentions referring to the most prominent individual, in general they lack the ability to correctly determine the *k*-number (see "Related Work Section"). However, the *k*-number is important for many reasons. To begin with, the clustering algorithms cannot accurately resolve it (see the "Related Work" Section). The second reason comes from the fact that knowing how many individuals carry the same name implies knowing the prior probability of two name mentions to corefer. Therefore, the need for coreference evidence can be dynamically adjusted and different schemes of clustering can be applied for each name. In some cases, this strategy may lead to a huge improvement in the global accuracy of a PCDC system (see the "Experiments" Section). The third reason comes from the point of view of an end user of a search engine that is likely to assign different subjective values to the clustering errors. An user interested in searching for a specific person is likely to appreciate an output in which all the mentioned individuals are listed over an output in which all the mentions have been resolved but some of the individuals have been utterly confounded.

In this paper we present various methods of estimating the *k*-number and we analyze the influence of knowing the *k*-number for a PCDC system. We present the following types of methods: (i) corpus independent methods based on external resource and (ii) corpus dependent methods, based on statistics run on name mention population. We present the experiments and the results related to the confidence and tolerance intervals, and the performances obtained compared against the baseline which is given by the gap statistics.

The paper is structured as follows: in Section 2, "Related Work" we review the relevant literature; in Section 3, "Name, Perplexity, Complexity" we establish the connection between name distribution and the *k*-number; in Section 4, "k-number Estimation" we present the methods of estimating the *k*-number; in Section 5, "Experiments" we present the experiments we run in order to compare the *k*-number estimation method. The paper ends with the "Conclusion and Further Research" section.

2 Related Work

The dominant working paradigm in the PCDC system engineering community is the vector space model. The context is represented as a vector and the decision to coreference two name mentions is made according to the their vector closeness in a vectorial space computed according to a given metrics and a threshold, usually empirically chosen.

In Baga&Baldwin (1998) the vectors are made up using all the words in the same sentence as the target name with a tf/idf weighting. The results reported are based on a corpus containing just one name, "John Smith". In a paper extending their work (Gooi&Allen 2002), it has been noted empirically that the accuracy of the results varies significantly from name to name. Indeed, considering just the information found in the same sentence as the name itself, a PCDC system obtains a good score for "John Smith". This happens because the prior probability of coreference of any two "John Smiths" mentions is low, because this is a very common name and none of the "John Smith" has an overwhelming number of mentions. But for other types of names the same system is not accurate. If it considers, for instance, "Barack Obama", the same system obtains a very low recall, as the probability of any two "Barack Obama" mentions to corefer is very high and the relevant coreference context is found very often beyond the sentence level.

A more informative set of vectors can be obtained by using bigrams with correlation statistics (Pederson et al. 2007). This work also considered the behavior of the clustering module when the k-number is given as an input parameter. They noted empirically that the accuracy of the system dropped when the k-number was known to the system. A typical case responsible for this outcome is the case in which an individual is mentioned very often, leaving few mentions for other individuals; when the number of clusters is passed in the input, the clusters representing the persons who are rarely mentioned are wrongly enriched. A subsequent paper (Pederson&Kulkarni 2009) presented a method of estimating the stop condition for the clustering algorithm. Considering also gap statistics, (Tibshirani 2001), three more criteria were proposed: PK1, PK2, PK3. These statistics, as well as gap statistics, are proved to work correctly in the case of a uniform distribution among the clusters. However, as it will be shown in Section 3, the name mentions distribution is very skewed.

In Popescu&Magnini (2007) the information revealed using the name mentions distribution is used for a PCDC system. In Popescu (2009), a method based on ordered statistics is presented which is proved to work well for skewed distribution. This work shows that the accuracy of a PCDC system actually improves using k-number estimates when the clustering methods are applied differentially, according to the prior coreference probability that the k-number indicates.

In (Ng et al. 2001), (Sanguinetti et al. 2005) the contexts are clustered using a variant of the spectral-clustering algorithm. The contexts are represented in a graph structure and the graph matrix is normalized. Their systems make use of the k-number which is dynamically determined by starting with a low rank matrix determined by the eigenvectors with the greatest values. The rank of the matrix is increased in an iterative process in order to find the desired number of clusters. If the linear separation of data along the initial eigenvectors covers all the data, then the algorithm stops. If a "fake" cluster, having the center at the origin, contains at least a point, it means that the present number of entities do not correctly account for the data set. Consequently, the rank of the matrix is upgraded and the clustering is repeated.

Advanced clustering techniques for PCDC have been described also by Han et al. 2005, Chen 2006. However, these techniques rely on the precise analysis of the context, which is a time consuming process. It has been also noted that, in spite of

deep analysis, the relevant coreference context is hard to find (Vu 2007). It has been estimated that as much as 60% of the total of pieces of news from a regular news corpus lack the proper information which allows the correct coreference (Popescu&Magnini 2007).

The work reported here is complementary to the above approaches. The methods of k-number estimation we present can be implemented in a full fledged PCDC system. We compute the confidence and tolerance intervals for the proposed statistics.

3 Name, Perplexity, Complexity

The k-number associated to a name is the number of different persons that name refers to in a given corpus. The corpus we use in this paper is Adige500k (Magnini et al. 2006). The set of names we use in this paper is made of 105 names. The news containing these names are part of the CRIPCO corpus (Bentivogli et. al. 2007).

The k-number is a random variable. Its distribution is rather skewed, and the variance from the average is very high. Most of the name mentions in a corpus refer to just one entity, but there are names that refer to more than 10 different persons. However, the percentage of names with an ambiguity greater than 3 is low. In Figure 1, we sketch the Lorenz curve. The average k-number is 3.25. However, the test set was chosen on ambiguity criteria, and the expected percentage of the 1-number names is 86%.

As Figure 1 shows the k-number distribution is very skewed and the variance from the mean is high. All the 105 names considered here are two tokens names. For three or more token names the dispersion around the average is quite low. In general, the three or more token names refer to a unique individual. However, the one token names are considerably more ambiguous and the dispersion is very high.

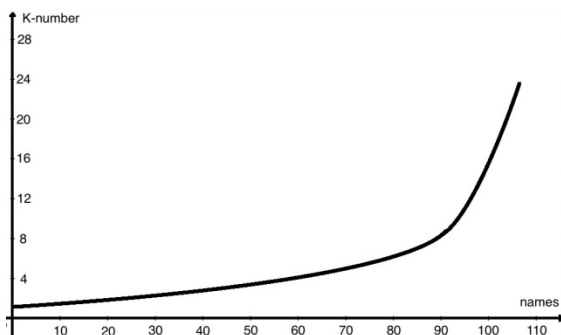


Fig. 1. k-number Distribution in CRIPCO

The number of mentions per name follows the same type of very skew distributions. In Table 1 we list the name mentions distributions: on the first column the number of different names, on the second column the interval of frequency these name belong to, on the third column the total number of occurrences in the corpus and

on the forth column the average number of the mentions for a name in the respective frequency class, which is the ratio between the second and the third column.

Table 1. Name/Mentions Distribution in Adige500k

frequency	name	mentions	average
1	317,245	317,245	1
2 – 5	166,029	467,560	2.81
6 – 20	61,570	634,309	10.3
21 – 100	25,651	1,090,836	42.52
101 – 1000	7,750	2,053,994	265.03
1001 – 2000	425	569,627	1340.29
2001 – 4000	157	422,585	2691.62
4001 – 5000	17	73,860	4344.7
5001 – 31091	22	190,373	8653.31

Comparing the first and the last rows of Table 1, we see that there are 22 names, representing 0,00025% of the names, which are mentioned as much as a third of the time the other 317, 245 names, representing 65% of the names, have been mentioned.

The k-number gives an important information regarding the coreference, namely the prior probability of coreference. There is a direct relationship between the prior coreference probability and the quantity of information required for deciding the coreference. The analysis carried out before shows that in many cases the coreference can be decided with a little similarity requirement, while in other cases the similarity threshold must be set up to a high value. Different type of clusterings may be used for different types of names in order to achieve a high accuracy level. A system that relies on only one type of clustering may experience great variance in the accuracy according to the corpora it processes. In the WePS contest, it has been noted that some variances existed between the results of the same clustering system obtained on the training vs. test corpora (Lefever 2007, Popescu 2007 among others).

We define the perplexity of a name as being the ratio between the k-number and the number of mentions of that name in the corpus. As the number of mentions or a given name and a corpus is a constant, there is a linear relationship between the perplexity and the k-number, therefore an estimation of the perplexity may be converted into an estimation of the k-number and vice versa. In Section 4 we develop a full method of estimating the k-number on the basis of an estimation of the name perplexity.

There is a direct relationship between ambiguity, perplexity and the complexity of the PCDC tasks. Intuitively, the larger the perplexity, the harder the PCDC task. However, this is not the case, because a high perplexity may actually simplify the task. In fact, considering the Gini's mean difference we can see how the name mentions are distributed relative to the individuals. Let X_1, X_2, \dots, X_n be the individuals that are named with the same name and let S be the set of the mentions of this name S_1, S_2, \dots, S_n . The Gini's mean difference is a measure of the spread of the information in the set:

$$\sum_{j=2}^n \sum_i \frac{|S_i - S_j|}{n(n-1)} = G \quad (1)$$

Equation (1) takes values between 0 and 1. The uniform distribution makes the Gini's factor null. In this case, a coreference algorithm which considers little context evidence works best. On the other hand, a value close to 1 indicates that there is one individual that has been mentioned. In this case, an algorithm requiring strong evidence for non coreference works best. A Gini's factor which is well inside the [0,1] interval shows that there are some individuals that have the majority of the mentions, while some others share more or less the same tiny amount from the rest of the mentions. This case is directly related to two phenomena manifested in a vectorial space, namely superposition and masking (Hastie et. al 2001, Popescu&Magnini 2007). The negative impact on the accuracy may be big if these phenomena remain unaddressed. The perplexity of a name mitigates between the two extremes and may indicate which way is safer: either considering a uniform distribution or a very skew one.

4 Name, Perplexity, Complexity

In this Section we present two different approaches to k-number estimation. The first class is represented by corpus independent methods. The k-number is estimated using an external repository, such as Wikipedia, Google or the telephone book. The great advantage of the corpus independent approach is that the estimation is computationally light. The disadvantages are related to the fact that the method is not completely generalizable, not all the persons in a corpus are in Wikipedia and there is no universal phone book – even a local newspaper has a significant number of mentions to alien names. The second class is represented by corpus dependent methods. The first dependent method uses the name perplexity. The names are grouped in a few categories according to their perplexity and an individual and a group estimation for the k-number is computed. The second dependent method uses gap statistics. We only sketch the gap statistics method because this method is fully presented elsewhere (see the “Related Work” Section). We use gap statistics as a baseline. We compare the results obtained by all these methods individually and in combinations.

Corpus Independent Methods

Telephone Book

A computationally costless solution is to use the phone book to estimate the ambiguity of a person. It is considered that the number of different persons carrying a name is exactly the number of persons with different phone numbers carrying the same name (Bentivogli et. al. 2007).

While this method computes the name ambiguity at the global level in linear time, it cannot be scaled. The ambiguity of a name in a corpus varies according to corpus size. It does not consider the alien names which represent roughly 7% of the names in Adige500k.

Wikipedia

Another computationally costless solution is to consider the number of different persons sharing a name equal to the number of different persons carrying that name and appearing in Italian Wikipedia pages.

Corpus Dependent Methods

Gap Statistics

In order to perform Gap statistics (Tibshirani et al., 2001) we used SenseClusters, which is a freely available system that clusters similar contexts (Pedersen et al. 2006). It creates a sample of reference data that represents the observed data as if it had no meaningful clusters in it and was simply made up of noise. The criterion function of the reference data is then compared to that of the observed data, in order to identify the value of the k-number in the observed data that is least like to be simply noise, and therefore represents the best clustering of the data.

Perplexity Estimates

The name perplexity is defined as the ratio between the k-number and the number of mentions of that name in the corpus. We are interested in finding a method of estimating the name perplexity using the corpus name distribution. In particular we are interested in one and two token names. This is because for three or more token names the perplexity is 1 in 99,6% of the cases (the name by itself is a relevant context for reference).

The method of estimating the perplexity, (and the k-number) of one or two tokens names is a two step method: (1) estimate the perplexity for one token and (2) use the perplexity of one token names to estimate the perplexity of two token names.

An estimate of the name perplexity of the one-token names is the size of the different one-token names with which it forms a complete name in the corpus. For example for the first name “John” the estimation of its perplexity is the size of the one-token last names it combines with in forming a name, like “Smith, Travolta, Kennedy” etc. The bigger the size of its complementary names, the higher is its name perplexity. In Table 2 we present these figures.

Table 2. Interval Perplexity One Token Estimates in Adige500k

occurrences	average estimated perplexity
1-5	4.13
6-20	8.34
21-100	17.44
101-1,000	68.54
1,000-5,000	683.95
5,000-31,091	478.23

We categorize the one-token names into perplexity classes. For convenience we use five classes, “very low” (VL), “low” (L), “medium”, (M) “high” (H) and “very high” (VH). However, the names do not behave alike. Apart from the skew distribution analyzed in Section 3, the names have different perplexity classes

according to whether they are first or last names. The perplexity of the first names is two times bigger than the perplexity of the last names. Consequently we keep the perplexity classes divided between first and last names respectively (the names that can function either way have two perplexities respectively). In Table 3a and 3b we list a representative one token name for each perplexity category. On the first column the names are listed, on the second column the computed perplexity (P), on the third column the number of occurrences as one-token name (O), on the fourth the number of occurrences in a two-token name (T) and on the last column the computed perplexity class (PC).

In Table 3a and 3b we list a representative one token name for each perplexity category. On the first column the names are listed, on the second column the computed perplexity (P), on the third column the number of occurrences as one-token name (O), on the fourth the number of occurrences in a two-token name (T) and on the last column the computed perplexity class (PC).

Table 3a., 3b. First, Last Name Representatives

Name	P	O	T	PC	Name	P	O	T	PC
Varena	5	10	85	VL	Dellai	7	31091	10722	VL
Camilla	276	664	1731	L	Ruini	15	554	203	L
Romano	14	886	6414	M	Prodi	52	9184	3382	M
Lorenzo	2088	2167	2198	H	Parolari	171	1,619	2207	H
Paolo	5255	4001	51244	VH	Rossi	753	7506	8356	VH

To each perplexity class we associate an interval which has the highest probability to contain the k-number for most of the names in that class. The constraints observed by these intervals are: (i) the intervals of any two perplexity class do not overlap and (ii) the percentage of the names from each of the perplexity class which have a k-number inside the associated interval is the highest possible.

To compute the borderline between two consecutive categories we employed an ordered statistics technique, namely the (p,γ) technique. The technique allows us to assert that the associated interval contains the k-number for at least 100p percentage of the names in the respective perplexity class with γ confidence. The interval between the lowest and respectively the largest k-number of the names within a perplexity class represent a 100γ percent tolerance interval for at least 100p percent of these names.

In order to determine a partition of the names into perplexity classes we start from a random sample of names, let's say X, where x_i represents the k-number of the ith name chosen, and consider its ordered sample pair Y (Y has the same elements like X, just that they are ordered). We start with a set of perplexity intervals: $(\xi_0, \xi_1]$, $(\xi_1, \xi_2]$, ..., $(\xi_4, \xi_5]$, corresponding to VL, L, M, H and VH categories. For each $(\xi_i, \xi_{i+1}]$ we want to know what percentage of the population enters in this interval (p) and which is the confidence that the names inside this interval have the k-number between ξ_0 and ξ_1 respectively.

For a given number ξ , the percentage of the name population having the name perplexity at most ξ is determined by finding the smallest Y_i greater than ξ . Therefore the percent of the population entering in $(\xi_i, \xi_{i+1}]$, p , is given by finding Y_i, Y_j such that $Y_i < (\xi_i, \xi_{i+1}] \leq Y_j$. The confidence is computed using Equation 2:

$$\gamma = P(F(Y_j) - F(Y_i) \geq p) = \int_p^1 \frac{\Gamma(n+1)}{\Gamma(j-i)} \Gamma(n-j+i+1) x^{j-i-1} (1-x)^{n-j+i} dx \tag{2}$$

For example, Suppose that $(\xi_0, \xi_1] = (0,2]$. Thus we are interested in finding p , the percentage of the name population such that we can be γ sure that at least p names have a perplexity between 1 and 2 inclusive. We take a random sample of $n = 30$ and suppose the smallest index i_1 such that $Y_i \geq 3$ for all $i > i_1$ is 17. We want to compute the confidence γ that at least $p = 60\%$ of the name population has the name perplexity within $(0,2]$:

$$\gamma = 1 - \int_0^{0.6} \frac{30!}{16!15!} x^{15} (1-x)^{14} dx \geq .965... \tag{3}$$

In practice we want to have optimal values for p and γ ; a large p implies a small γ and vice-versa. The optimality is determined by the accuracy of the PCDC system: we want to have the largest possible percentage of names into each partition such that our confidence that the names inside each partition have the same perplexity. In the next section we present the figures for confidence and tolerance we obtained for the partition in the 5 classes. However, the number of partitions could be variable, according to the possibility of control (p,γ) values, which ultimately, is a property of the population.

Table 4a. and 4b. Name Perplexity: first (second column) and last name (third column)

perplexity class	percentage	percentage
very high (VH)	5.3%	1.8%
high (H)	8.7%	3.36%
medium (M)	20.9%	17.51%
low (L)	27.6%	20.31%
very low (VL)	37.5%	57.02%

The perplexity of the two token names is deduced from the perplexity of the one token names. The general rule is that a two token names is assigned to the perplexity category which is immediately below the perplexity category of the one token name which has the minimal perplexity class. For example, the estimated perplexity of the name “Lorenzo Dellai”, which has “Lorenzo” in the H(igh) perplexity class and has “Dellai” in the V(ery)L(ow) perplexity class, belongs to the VL perplexity class (see Table 4a, 4b). If both one token names belong to the top perplexity names in their categories, then the two token name belongs to the minimum of their perplexity

classes, rather than the one category below. For example “Paolo Rossi” belongs to the V(ery)H(igh) perplexity class because both “Paolo” and “Rossi” are among the top perplexity names in their category , VH.

However, a more detailed heuristics, which takes into account not only the category of the one token name, but also how much the perplexity of that name departs from the average inside the class, can be used. In fact, it turns out that the best estimation results for two token names are obtained by considering separately the extreme. Therefore, the general rule is the one presented in the above paragraph; but if the perplexity of one token name is in the first or last quarter then the perplexity category is the average of the perplexity of the two names, first and last respectively.

Comparative Overview

In Table 5 we present comparatively the characteristics of these methods. The parameters we use to evaluate each methods are: LD-language dependent, whether the method could be applied to any language, ER-external resource, whether the method requires additional resources, SC- scalability, whether the method can account for different corpora, CSYN-corpus synchronization, whether the corpus information justifies the results, EXP-whether the method allows the treatment of exception, such as foreign names, CC-computational cost, the computational burden brought by the respective method.

Table 5. k-number Estimation Methods Overview

	TelePhone Book	Wiki pedia	Gap statistics	Corpus Perplexity
LD	√			
ER	√	√		
SC			√	√
CSYN			√	√
EXP		√	√	√
CC	O(1)	O(1)	O(n ²)	O(n)

5 Experiments

The skewness of the distribution of the name/mention ratio is influenced by two factors: 1) whether the name is a common one, many different people in the population are carrying it and 2) whether there is one person that is famous and it makes it frequently in the news.

An evaluation corpus should take into account the skewness of the distribution in order to ensure a competitive evaluation of the results. The CRIPCO resource is compiled from the Adige500k, considering 209 different names, 709 different entities, and more than 43,700 newspaper articles. The partition of the gold standard in 15 classes, representing the different levels of entity fame and name ambiguity, allows for a more informative evaluation and analysis of systems performances. The experiments and the results reported in this section are carried out, and respectively

obtained, by using half of the CRIPCO corpus, the one which is the part currently distributed, containing 105 names. In Table 5 we present the strata values for this half of the CRIPCO. The strata used are A – ambiguous name, NA – not ambiguous name; F – famous person(s); NF – not famous person.

Table 6. Strata Values CRIPCO Corpus

	A	A	NA	NA
	F	NF	F	NF
Strata value	33	22	42	8

A point estimation experiment was conducted. The Telephone book, Wikipedia methods can produce only a point estimation. Also the Gap method returns the best estimation, which is a point. By default, the perplexity method outputs an interval, not a point. However, we chose the best guess inside the interval considering a normal distribution around the median of the interval and considering that a 2σ variation corresponds to the extreme values. The results, in terms of the BCUBED formula, are presented in Table 7.

Table 7. Point Estimation Accuracy

	A	A	NA	NA	total
	F	NF	F	NF	
TelephoneB	.094	0	.581	.375	.295
Perplexity	.031	.045	0.907	0.75	0.467
Wikipedia	.031	0	.698	.5	.333
Gap	.125	.182	.465	.125	.276

Table 8. Combined Point Estimation Accuracy

	A	A	NA	NA	Total
	F	NF	F	NF	
G+TpB	.094	.182	.837	.5	.448
G+ Perpht	0.66	0.45	1	1	.0572
G+ Wiki	.031	.045	.860	.625	0.419
G+Wiki+TpB	.031	.091	.953	.875	0.486

The perplexity estimation is sensibly more accurate than the rest. However, important differences exist between all these methods. In particular, gap statistics scores consistently several times better when ambiguous names are considered. In order to take advantage of the differences, we combined the above methods with the Gap statistics one. The results are presented in Table 8 (G= gap statistics,

TpB=phonebook, Wiki= Wikipedia, Perpxt= perplexity). The method of combining these methods is the following: if a name appears in Wikipedia, then use the number of Wikipedia pages; if not, check the phone book, and use the number present there. If the name is not in the phone book, then use 1, as a default value. The estimation is passed to the Gap statistics as an upper threshold.

The Gap method, in combination with any other methods, loses some points for the ambiguity class, but it also gains points for the non-ambiguous class. All in all, combining the methods tends to uniform the results toward the upper individual value.

The results suggest that, due to the inherent problems related to the vectorial model, the gap statistics cannot accurately predict the k-number. However, an interpolation with other methods which brings additional information from corpus, such as the perplexity method, is likely to produce a jump in the accuracy.

A second experiment we carried was to determine the accuracy of interval estimation. The names in the LM-coref have the following distribution, where on the first column the k-number is listed, on the second column the number of names having that k-number, on the third column the percentage from the whole population (there are 105 names see Table 6), on the third and fourth columns the mass distribution, number and percentage.

Table 9. k-number Distribution in CRIPCO

k-number	#	%	<	%
1	51	48.57	51	48.57
2	14	13.34	65	61.9
[3-7]	29	27.61	94	89.52
[8-13]	8	7.61	102	97.14
[14-24]	5	2.85	105	100

The names with k-number bigger than 10 have one occurrence each. The names having the k-number bigger than 12, which is the median, represent only 3.86%. The partition specified in Table 4 leads to the following estimation results, in spite of this skew distribution:

Table 10. Interval Estimation Accuracy

interval	Accuracy(p)	Tolerance(γ)
(0-1]	94.11	82.35
[1-2]	100	73.84
(1,2]	81	78.57
[3-7]	78.94	71.48
(8-13]	80.00	83.34
[9-10]	66	100
(13,24]	100	100

Table 11 shows the distribution of the k-number in the perplexity classes, as we can see the borders are clearly defined. The Box Plots in Figure 2 show the outliers. The outliers for low/high perplexity classes are not significant. The highest variance is found in [3-7].

Table 11. k-number Distribution in Perplexity Classes

k-number	VL	L	M	H	VH
1	41	8	1	1	0
2	3	11	0	0	0
[3-7]	1	3	21	3	0
[8-13]	0	1	0	7	0
[14-24]	0	1	0	1	1

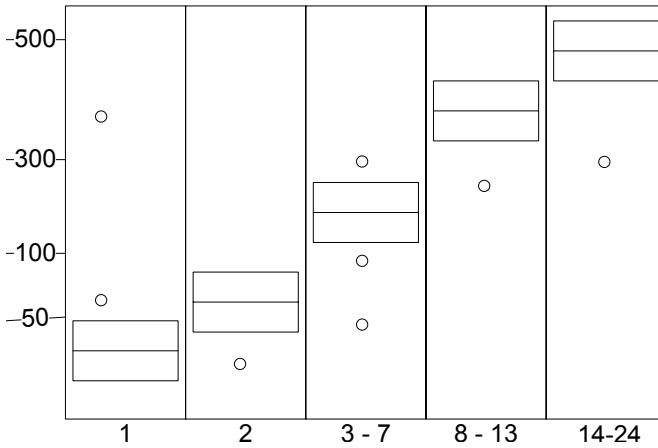


Fig. 2. BoxPlots for k-numbers vs. Perplexity

6 Conclusion and Further Research

Without further adjustments, a vectorial model cannot resolve the problem of considering too much or too little contextual evidence in order to obtain a good precision for “John Smith”, which is a frequent name having no famous individual carrying it, and simultaneously a good recall for “Barack Obama”, which is very infrequent name with a famous individual carrying it. The estimation of the k-number is instrumental in overcoming this problem. The results presented show that a good estimate of the k-number could be obtained by relying on the name distribution in the corpus for computing the name perplexity. An interpolation between perplexity and gap statistics works best.

In the future we will implement a clustering method with dynamic parameters, such that the similarity threshold may be adequately changed for each name individually according to the perplexity number.

Another direction of research for us is to generalize the mechanism of dividing a given population in perplexity classes, such that the method may be used in other NLP tasks. Instead of using a static heuristics for determining the perplexity classes we think it is possible to automatically learn the rules that lead to the optimal partition. As noted in Section 4, the best results are obtained by employing a rule based heuristics which takes into account the variation inside the perplexity class. The form of rules is crucial and the challenge is to automatically learn it.

References

1. Artiles, J., Gonzalo, J., Sekine, S.: Establishing a benchmark for the Web People Search Task: The Semeval WePS Track. In: Proceedings of Semeval 2007, Prague (2007)
2. Bagga, A., Baldwin, B.: Entity-based Cross-Document Co-referencing using the Vector Space Model. In: Proceedings of the 17th International Conference on Computational Linguistics (1998)
3. Bartalesi, V., Sprugnoli, R.: EVALITA 2009: Description and Results of the Local Entity Detection and Recognition (LEDR) task. In: Proceeding of IAIA (2009)
4. Bartalesi, V., Sprugnoli, R.: EVALITA 2007: Description and Results of the TERN Task. In: Proceedings of EVALITA 2007. IA (2007)
5. Bentivogli, L., Girardi, C., Pianta, E.: Creating a Gold Standard for Person Cross-Document Coreference Resolution in Italian News. In: LREC Workshop on Resources and Evaluation for Identity Matching, Entity Resolution and Entity Management. European Language Resources Association (2008)
6. Gooi, C., Allan, J.: Cross-Document Coreference on a Large Scale Corpus (2004)
7. Hastie, T., Tibshirani, R., Friedman, R.: Elements of Statistical Learning. Springer Press, Heidelberg (2001)
8. Grishman, R.: Whither Written Language Evaluation? In: Proceedings Human Language Technology Workshop, San Mateo, pp. 120–125 (1994)
9. Hardy, H., Shimizu, N., Strzalkowski, T., Ting, L., Wise, G., Zhang, W.: Cross-Document Summarization by Concept Classification. In: Proceedings of SIGIR 2002, Tampere, Finland, August 11-15 (2002)
10. Lefever, E., Hoste, V., Timur, F.: AUG: A Combined Classification and Clustering Approach for Web People Disambiguation. In: Proceedings of SemEval 2007, Prague (2007)
11. Magnini, B., Speranza, M., Negri, M., Romano, L., Sprugnoli, R.: I-CAB – the Italian Content Annotation Bank. LREC (2006)
12. Ng, V.: Shallow Semantics for Coreference Resolution. In: Proceedings of IJCAI (2007)
13. Pedersen, T., Kulkarni, A.: Automatic cluster stopping with criterion functions and the Gap Statistics. In: Proceedings of the Demo Session of HLT/NAACL (2006)
14. Pedersen, T., Kulkarni, A.: Unsupervised Discrimination of Person Names in Web Contexts. In: Gelbukh, A. (ed.) CILing 2007. LNCS, vol. 4394, pp. 299–310. Springer, Heidelberg (2007)
15. Pianta, E., Girardi, C., Zanoli, R.: The TextPro tool suite. In: 6th International Conference on Language Resources and Evaluation (2008)
16. Popescu, O., Magnini, B.: Inferring Coreference Among Person Names in a Large Corpus of News Collection. In: Basili, R., Pazienza, M.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 362–373. Springer, Heidelberg (2007)

17. Popescu, O., Magnini, B.: Iterative Person Coreference Using Name Frequency Estimates. In: Proceedings of LTC, Poznan (2007)
18. Popescu, O.: Name Perplexity for Cross Document Coreference. In: Proceedings of EMNLP, Singapore (2009)
19. Sanguinetti, G., Laidler, J., Lawrence N.: Automatic Determination of the Number of Clusters using Spectral Algorithms. In: Proceeding of IEEE Machine Learning for Signal Processing (2005)
20. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the Gap statistic. *Journal of the Royal Statistics Society* (2001)
21. Vu, Q., Masada, T., Takasu, A., Adachi, J.: Using a Knowledge Base to Disambiguate Personal Name in Web Search Results. In: Proceedings of SAC 2007, Seoul, Korea (2007)
22. Zanoli, R., Pianta, E., Giuliano, C.: Named Entity Recognition through Redundancy Driven Classifiers. In: EVALITA 2009 Workshop on Evaluation of NLP Tools for Italian (2009)

Coreference Resolution Using Tree CRFs

Vijay Sundar Ram R. and Sobha Lalitha Devi

AU-KBC Research Centre
MIT Campus of Anna University, Chennai-600044
sobha@au-kbc.org

Abstract. Coreference resolution is the task of identifying which noun phrases or mentions refer to the same real-world entity in a text or a dialogue. This is an essential task in many of the NLP applications such as information extraction, question answering system, summarization, machine translation and in information retrieval systems. Coreference Resolution is traditionally considered as pairwise classification problem and different classification techniques are used to make a local classification decision. We are using Tree-CRF for this task. With Tree-CRF we make a joint prediction of the anaphor and the antecedent. Tree-based Reparameterization (TRP) for approximate inference is used for the parameter learning. TRP performs an exact computation over the spanning trees of a full graph. This helps in learning the long distance dependency. The approximate inference methodology does a better convergence. We have used the parsed tree from the OntoNotes, released for CoNLL shared task 2011. We derive features from the parse tree. We have used the different genre data for the experiments. The results are encouraging.

Keywords: Coreference chains, treeCRFs, pronominal resolution.

1 Introduction

Coreference resolution is required in most of the natural language processing applications such as Information Extraction, Question Answering system, similarity analysis, summarization etc. Coreference resolution is defined as the task of identifying which noun phrases (NPs) or mentions refer to the same real-world entity in the text or dialogue [20]. Coreference resolution task involves resolving the anaphoric pronouns and the anaphoric noun phrases. There are many different approaches applied for this task. Here we use a conditional random fields (CRFs) based graphical approach to solve the task, we use tree CRFs for pronominal resolution and linear CRFs for non pronominal resolution.

Research in coreference resolution task are diversified into various directions, to obtain an improved coreference chain, such as different approaches for generating coreference chain, different metrics for evaluation of the coreference chain, identification of non-referential pronouns and development of coreference systems, enabling to add new features and algorithms to test and enhance this research. This anaphoric resolution task started in 1980s and resolving pronominal anaphors were mostly concentrated. One of the early works is Hobb's non naive approach, which

relies on semantic information [8]. Carter with Wilkas' common sense inference theory came up with a system [3]. Carbonell and Brown's introduced an approach of combining the multiple knowledge system [2]. The initial approaches, where broadly classified as knowledge poor and rich approach.

Syntax based approach by Hobb (naive approach), centering theory based approaches [10,11] and factor/indicator based approach such as Lappin and Leass' method of identifying the antecedent using a set of salience factors and weights associated to it [12]. This approach requires deep syntactic analysis. Ruslan Mitkov introduced two approaches based on a set of indicators, MOA (Mitkov's Original Approach) and MARS (Mitkov's Anaphora Resolution System) [17]. These indicators return a value based on certain aspects of the context in which the anaphor and the possible antecedent can occur. The return values range from -1 to 2. MOA does not make use of syntactic analysis, whereas MARS system makes use of shallow dependency analysis.

There are many different works done for coreference chain detection task using Machine learning (ML) technique. These ML approaches vary with the learning algorithms used and in the presentation of the training and testing files. Dagan and Itai did an unsupervised approach to find the antecedents of the pronoun, where they used statistically collected co-occurrence words from Hansa corpus [5]. Cohn came up with rule learner based approach for this task [4]. Aone and Bennett [1], McCarty and Lahnert [16], Soon et. al [25] and Ng and Cardia [19] have used decision tree based approach for this task. These works varied with the number of features chosen for classification. Cardie and Wagstaff did coreference resolution cluster based approach. Ng and Cardia did coreference resolution task in two steps, determining the anaphoricity of all types of mentions and using this for built coreference chain using classifier techniques. Daelmas and Van den Bosch (2005) used a memory based learning (TiMBL) and rule learners such as RIPPER are used. There are also works based on neural network based techniques like voted perceptron and statistical learner like SVM [20]. Hoste did a coreference resolution system for Dutch using TiMBL, where feature optimization and feature selection is discussed [7]. Marta Recasen did a detailed study of features used in TiMBL for pairwise classification in Spanish [23]. There are few works in Conditional Random Fields. McCallam and Wener, used CRFs for finding the antecedents for the anaphors [28]. Li et al used CRFs for resolving Chinese personal pronouns [6]. Sobha et al used linear CRFs and tree CRFs for pronominal resolution and compared the performance [15].

Coreference resolution task got boosted with CoNLL 2011 shared task "Modeling Unrestricted Coreference in OnoNotes". In this task, there were 18 different systems participated. The systems were built with various machine learning, rule based and hybrid approaches [21]. Sobha et al has used conditional random fields in building noun phrase anaphora resolution [14].

To enhance the anaphoric detection there were works in identifying the non-anaphoric pronouns and entities. Bergsma did a distributional method based approach for detection non anaphoric 'It'. Vieira and Poesio did a rule-based approach for identifying the non-anaphoric definite descriptions [20].

There are few publicly available Coreference systems, where we can experiment with changing the features and the algorithms. The first available system was JavaRAP, java implementation of Lapin and Leass (1994) algorithm. This is mainly for pronominal resolution. GuiTAR is an anaphora resolution system designed in a

modular way. This tool has the implementation of MARS (Mitkov's Anaphora Resolution System). This tool was developed as a part of segmentation and summarization research [22]. BART is a modular toolkit for developing coreference application. This has a coreference resolution implemented using Soon et. al., features. This tool kit has a built in maximum entropy learner [30]. Reconcile provides facilities for experimental evaluation. It has various learning algorithms, available in Weka toolkit for classification task, Single-link, Best First, Most recent First for clustering and B^3 , MUC and CEAF scoring algorithms [26].

The next section is on our approach to this task, where we have explained about the pronominal resolution and non-pronominal resolution. The third section describes the experiments and evaluation and the paper ends with the conclusion.

2 Our Approach

In the present work we come up with a Coreference Resolution system using Conditional Random Fields. We have divided the task into pronominal resolution and non-pronominal resolution. Tree CRFs is used for pronoun resolution, linear CRFs is used for non-pronominal resolution and heuristic based approach for generation of chain.

2.1 Pronominal Resolution

Pronominal resolution is the task of identifying the referent of a pronoun called antecedent. We have used Tree CRFs based graphical technique for resolving the pronoun.

2.1.1 TreeCRFs

When general graphs are used instead of linear chains in CRFs, it is called as General CRFs or Tree CRFs. In this more general factor graphs and approximate inference algorithms are used [28].

This is used in applications, where multiple labeling is required to be done simultaneously. Here a joint prediction is done on multiple output labels. Sutton et al has described this task by doing POS and chunking task simultaneously [28]. Tree CRFs uses a tree based reparameterization (TRP) algorithm which includes Belief propagation (BP) is used for parameter learning task. This approximate inference algorithm helps in providing a new conceptual view of a large class iterative algorithms for computing approximate marginals in graph with cycles. This performs exact computations over spanning trees of the full graph. In TRP, it updates suffices to transmit information globally throughout the graph, it might be expected to have better convergence properties than the purely local BP updates [31].

There are few works where Tree CRFs is used in Semantic role tagging tasks. Shilpa Arora et al have used in Japanese Semantic Role labeling [24], Cohn and Philip Blunsom did Semantic Role labeling for English using Tree CRF and Erwan Moreau used treeCRFs for a developing a generic tool for semantic role tagging [18].

Here we try to make a joint prediction of the anaphor and the antecedent. As the exact computation over the spanning tree of the full graph is done, this helps in learning long distance relation. The Tree CRFs forms a clique between two nodes and tries to predict the labels of the two nodes based on the node feature, context features and the features based edge between the two nodes

When a cliques $C=\{y_c, x_c\}$ are given, CRFs define the conditional probability of a state assignment given the observation set.

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \phi(x_c, y_c)$$

$$\phi(x_c, y_c) = \exp \sum_k \lambda_k f_k(x_c, y_c)$$

Here $\Phi(x_c, y_c)$ is a potential function. And $Z(x)$ is the partition function.

Features used in Tree CRFs learning:

The OnoNotes provided in the CoNLL shared task 2011 is used in this work. We have used the phrase structure tree provided in the OnoNotes, to come up with tree based features. The performance of the ML technique is based on the features used in the training and testing. The features used in Tree CRFs are classified into three 1, Node Features, 2, Sibling and Parent Nodes based Features and the edge clique features. The features used in the Tree CRFs are as follows

Node Features - These are features on each individual nodes of the graph. These can be considered as the basic features.

1. IsSubject – whether the Node’s syntactic category is a subject
2. IsObject– whether node’s syntactic category is an object
3. IsIndirect-Object - whether node’s syntactic category is an indirect object
4. IsPcomp - whether node’s syntactic category is a complement of PP
5. IsSbar – whether the node is in Sbar
6. IsSbarPronoun – whether the node is in Sbar and a pronoun
7. IsNE – whether node is a NE

Sibling and Parent node Features – These are the features on each individual node with respect to the other surrounding nodes

1. IsSubjParentNodeRoot - whether the node which is a subject, and its parent node is a root
2. IsNPrightSibVP – whether node’s right sibling a VP
3. IsNPleftSibVP – whether node’s left sibling a VP
4. IsNPrightSibPP – whether node’s right sibling a PP
5. IsNPParentPP – whether node’s parent node a PP
6. IsPronounParentNodeRoot – whether the node is a Pronoun and the node’s parent node a root.
7. IsPronounParentNodePP – whether the node is a pronoun and the node’s parent node a PP.

8. `IsSbarRightPronoun` – whether the node is a pronoun and the pronoun node's right sibling is in Sbar

Edge Clique Features – These are the features taken over the output random variables that factor over the tree (over the edge cliques). These are joint features

1. `IsAnaAnteNEMatch` – whether Anaphor's and Antecedent's NE category match
2. `IsbothPronoun` - Are both Anaphor and Antecedent pronouns
3. `IsAnaAntePronounMatch` – If both Anaphor and Antecedent are pronouns, do they match
4. `IsbothInCurrentSbar` – are both Anaphor and Antecedent inside current Sbar
5. `IsAnteInAnaParentSbar` – Is Antecedent in Anaphor's parent Sbar.
6. `AnaAnteCliqueDistance` – The edge distance between the anaphor and the antecedent. The edge distance is the tree depth level.

This tree CRFs has more advantage over linear CRFs. In Linear-chain CRF, edge cliques are the edges between two adjacent nodes where as for Tree CRFs, edge cliques are edges between parent and child nodes. So we used the parse structure from Ontonotes to model the Tree CRFs. Consider the following sentences

“Mr. Peterson also says he doesn't consider himself a financial planner anymore.”

In the above sentence there are two anaphoric pronouns, ‘he’ and ‘himself’. Here ‘himself’ refers to ‘he’ and ‘he’ refers to ‘Mr. Peterson’. The tree representation given in Ontonotes for the above sentence is given in figure 1.

```
(TOP (S (NP-SBJ (NNP Mr.)
              (NNP Peterson))
        (ADVP (RB also))
        (VP (VBZ says)
            (SBAR (-NONE- 0)
                (S (NP-SBJ (PRP he))
                    (VP (VBZ does)
                        (RB n't)
                        (VP (VB consider)
                            (S (NP-SBJ (PRP himself))
                                (NP-PRD (DT a)
                                    (JJ financial)
                                    (NN planner))))
                            (ADVP-TMP (RB anymore))))))
                (. .)))
```

Fig. 1. Tree representation of the sentence

The tree features that are true for the two antecedent-anaphor pairs (Mr. Peterson – he and he – himself) are presented in table 1.

Table 1. Features, which are true for the above antecedent anaphor pairs

Mr. Peterson - he	he - himself
Mr. Peterson	he
Node Features:	Node Features:
IsSubject	IsSubject
IsSubjParentNodeRoot	
he	himself
Node Features:	Node Features:
IsSubject	IsSubject
IsNleftSibVP	IsNrightSibVP
	IsNleftSibVP
Sibling and Patent node Features:	Sibling and Patent node Features:
IsNrightSibVP	IsNrightSibVP
Edge Clique Features:	Edge Clique Features
IsAnaAnteNEMatch	IsAnaAntePronounMatch
IsAnaAntePronounMatch	IsbothInCurrentSbar
	IsbothPronoun

We used GRMM toolkit for Tree CRFs [27].

2.2 Non-pronominal Resolution

Non-pronominal resolution or noun phrase resolution is the task of identifying the all NPs / mentions that refers to the same entity in a text or dialogue. This includes named entities, definite descriptions that refer each other. To identify the non-pronominal referent and anaphors, we used linear CRFs [13]. Though Tree CRFs learns better, when there are different constraints and ambiguous data, it performs poor for this task, as this resolution majorly depend on the word match, acronym as shown by Soon et al [25].

Features Used in Linear CRFs:

As the noun phrase resolution is mostly dependent on the word similarity, the features are chosen properly for the algorithm to learn the similarities.

He features used are

- 1, head noun similarity (in the mention only the head noun is reconsidered for word match)
- 2, Person, number gender
- 3, Acronyms
- 4, Position of the words

We used CRF++ toolkit for linear CRFs [29].

2.3 Chain Generation

The coreference chains are generated using heuristic rules. The coreferring pairs are obtained from the treeCRFs (pronominal pairs) and the coreferring NP pairs identified

by the linear CRFs. Each member of the pairs is compared with the other members, the common elements are grouped together. Similarly it is done for all the pairs in the generated two lists. After comparing the members and making groups, the chains are formed using the members in each group.

3 Experiments, Results and Discussions

3.1 Pronominal Resolution

We used the five genres of data provided for the CoNLL Shared Task 2011. The genres are NewsWire (NW), Broadcast News (BN), Broadcast Communications (BC), Magazine (MZ) and Weblogs (WB). The distribution of anaphoric and non-anaphoric pronouns in the training and testing data are given in the table 2. The Training data of the CoNLL task was taken as training data and the development data is used as the testing data.

Table 2. Distribution of Anaphoric and Non-anaphoric pronouns in Training and Testing data

Genre	Training Data		Testing Data	
	Anaphoric	Non-Anaphoric	Anaphoric	Non-anaphoric
BN	7141	1738	900	221
NW	7705	1356	991	174
BC	18802	5176	1731	547
MZ	9678	2608	445	68
WB	11612	3754	783	323

To analyse the variation in the anaphoricity of the pronouns, we calculated the percentage of non-anaphoric pronouns in each genre. It is presented in table 3.

When we analyse the table 3, the variation in the non-anaphoric pronouns across genres are very well seen. In WB genre, the person pronouns such as ‘he’, ‘she’, ‘him’ and plural pronouns such as ‘we’, ‘us’ are mostly non-anaphoric. Similarly in BN genre except person pronoun such as ‘he’, ‘she’, ‘they’ ‘them’ most of the pronouns have more number of non-anaphoric pronouns. In BC genre one-third of the pronouns are non-anaphoric.

Training: We extracted the features described in section 2.1 for each possible antecedent and pronoun pair from each files in each Genre. The training data has the positive and negative pairs. Tree CRFs is trained using the features extracted from the files in each genre and individual language models are generated for each genre.

Testing: Similarly extracted the features for each pair of possible antecedents and the anaphors. This is tagged with the corresponding genre language model. The results obtained are presented in the table 4.

Table 3. Distribution of Non-anaphoric pronouns in percentage

Pronoun	BN %	BC %	NW %	MZ %	WB %
Yours	-	-	-	-	6.60
Ours	50.00	4.61	-	-	-
Your	75.00	24.14	-	75.00	36.96
Myself	75.00	24.14	-	-	-
mine	-	24.14	-	-	-
we	31.13	42.11	43.86	18.18	90.54
they	6.40	12.82	2.54	3.23	8.33
them	4.35	6.67	6.06	45.45	7.50
their	75.00	2.50	6.38	45.45	2.82
us	29.63	47.06	36.84	50.00	93.33
It	41.06	49.00	24.47	45.45	42.41
its	29.63	47.06	9.42	7.69	5.71
I	14.02	4.61	2.27	3.33	6.60
me	12.50	7.69	33.33	45.45	13.33
you	57.80	44.50	66.67	-	72.58
my	57.80	6.98	18.18	3.23	3.45
he	2.13	1.26	1.96	45.45	42.41
him	31.25	3.23	20.83	8.33	33.33
she	2.94	1.35	43.86	18.18	90.54
her	29.63	47.06	18.18	7.69	5.71
his	29.63	1.20	6.90	50.00	1.32
our	31.25	33.96	20.83	8.33	55.56
herself	-	1.35	-	18.18	90.54
himself	31.25	33.96	16.67	8.33	33.33
yourself	-	-	-	-	66.67
themselves	-	2.50	20.00	75.00	13.33
itself	75.00	24.14	25.00	75.00	36.96
ourselves	-	75.00	24.14	-	-

Table 4. Performance of Pronominal resolution in each Genre

Genre	Precision %	Recall %
BN	74.10	61.55
NW	73.12	65.7
BC	70.13	26.13
MZ	64.72	68.1
WB	50.6	23.0

From table 5 we are able to infer that treeCRFs identifies the antecedents of the person pronouns such as ‘he’, ‘him’, ‘she’, ‘her’, ‘his’ fairly good. Even though BC genre has more number of non-anaphoric pronouns, the system has identified its antecedents with good precision. In resolving 3rd person neuter pronouns ‘they’, ‘them’, ‘their’, ‘it’, ‘its’, treeCRFs has resolved the antecedents badly, when compared to the person pronouns. As the person, number and gender of the 3rd person neuter pronouns match with many of the other non-antecedent nouns. In NW genre, it has failed poorly for pronouns, ‘we’, ‘us’, and ‘them’, though most of these pronouns

are anaphoric. The resolution of the pronouns ‘we’ and ‘us’, in WB genre is more interesting. Though most of the occurrence (more than 90%) of the pronouns ‘we’ and ‘us’ are non-anaphoric, the antecedent for the anaphoric ‘we’ and ‘us’ are identified fairly good.

Table 5. Percentage of Pronominal resolution in each Genre, Pronoun-wise

Pronoun	BN %	BC %	NW %	MZ %	WB %
yours	8.99	-	-	-	2.82
ours	12.98	-	-	-	-
your	8.70	34.21	-	50.00	15.28
myself	43.18	100.00	-	-	-
mine	-	100.00	-	-	-
we	28.57	24.39	8.57	12.50	45.00
they	42.54	44.75	55.08	42.86	45.00
them	8.70	34.21	15.15	33.33	27.50
their	43.18	35.00	38.14	46.77	15.28
us	29.17	48.00	25.00	20.00	52.00
it	12.98	1.13	35.02	9.62	5.06
its	58.33	7.14	62.32	40.74	10.81
i	8.99	0.97	34.15	65.52	2.82
me	74.83	1.96	72.67	33.33	27.50
you	42.54	44.75	-	-	45.00
my	40.00	3.45	10.00	100.00	45.00
he	74.83	71.21	72.67	70.83	46.67
him	68.18	48.00	53.33	70.00	52.00
she	71.43	94.83	61.11	20.00	72.73
her	50.00	76.92	50.00	60.00	58.33
his	74.44	77.03	64.77	70.31	58.67
our	20.59	20.00	17.39	40.00	100.00
herself	-	94.83	-	20.00	100.00
himself	50.00	33.33	80.00	50.00	100.00
yourself	-	-	55.08	-	-
themselves	-	100.00	45.45	50.00	28.57
itself	43.18	100.00	25.00	100.00	50.00
ourselves	-	100.00	100.00	-	100.00

Pronominal resolution is poor in WB genre, which consists more of casually written documents. In WB genre, pronouns such as ‘we’, ‘us’, ‘you’, ‘she’ and ‘herself’ has more than 75% as non-anaphoric pronouns, the system has identified its antecedents in higher rate of accuracy, these show that the treeCRFs is capable of learning rules to disambiguate ambiguous pronouns. The system has failed to identify antecedents of the first person pronouns such as ‘I’ and ‘me’ in most of the genres, particularly in BC genre. Most of the first person pronouns points to the speaker or the author of the log. This can be handled easily using heuristic rules.

3.2 Non-pronominal Resolution and Coreference Chains

In non-pronominal resolution, we extracted the features described in section 2.2. From training data, the positive and negative pair of mentions from each file is

extracted with the features for training. We have used only 70 files of NW genre to create a language model from non-pronominal resolution.

In the testing phase, we extracted the features for mention pairs obtained from each file. Using the language model built, positive pairs of mention are identified for each file.

The non-pronominal resolution is not separately evaluated as the pronominal resolution. Only the coreference chains are evaluated.

The coreference chains are built using the positive pairs obtained from the pronominal resolution and the non-pronominal resolution. These chains are evaluated with B³, CEAFE, MUC scoring metrics. The results for each genre are tabulated in the following table 6.

Table 6. Evaluation of the Coreference Chains

Metric	Genre	Recall	Precision	F Measure
MUC	NW	51.67	51.19	51.43
	BN	48.4	61.97	54.35
	MZ	51.89	54.72	53.72
	BC	24.76	45.25	32
	WB	36.31	46.66	40.48
Average		42.60	51.95	46.39
BCUB	NW	68.67	70.37	69.51
	BN	61.76	79.06	69.35
	MZ	66.48	74.34	70.16
	BC	45.7	79.81	58.12
	WB	59.32	74.86	66.17
Average		60.38	75.68	66.66
CEAFE	NW	41.24	41.65	41.44
	BN	54	41.43	46.89
	MZ	48.75	45.39	47.01
	BC	48.75	25.92	33.44
	WB	45.52	36.18	40.32
Average		47.65	38.11	41.82
Overall Average		50.21	55.24	51.62

The performance of pronominal resolution is reflected exactly in the coreference chain score. Here without using any semantic modules and dictionaries the comparison of the words in the mention pairs are done, we obtain a comparably good result. Though the word match based on the head noun helps in increasing the precisions, it affects the precision. The word match needs to be flexible.

4 Conclusion

The paper focused on building Coreference chains, where Tree CRFs is used for pronominal resolution and identified anaphors and antecedents are used in building coreference chain, where non-pronominal anaphors are identified using linear CRFs. The pronominal resolution is evaluated across genre and understood that tree CRFs

works even for genres having more non-anaphoric pronouns. The non-pronominal anaphoric identification is done with word match as primary features. We will further work towards improving the coreference chain using semantic recourses.

References

1. Aone, C., Bennett, S.: Evaluating automated and manual acquisition of anaphora resolution strategies. In: 33rd Annual Meeting of the Association for Computational Linguistics, pp. 122–129 (1995)
2. Carbonell, J.G., Brown, R.D.: Anaphora resolution: A multi-strategy approach. In: 12th International Conference on Computational Linguistics, pp. 96–101 (1988)
3. Carter, D.: Interpreting anaphors in natural language texts. Ellis Horwood Ltd., Chisester (1987)
4. Cohen, W.: Fast effective rule induction. In: 12th International Workshop Conference on Machine Learning, pp. 115–123. Morgan Kaufmann Publishers, Inc. (1995)
5. Dagan, I., Itai, A.: Automatic processing of large corpora for the resolution of anaphora references. In: 13th Conference on Computational Linguistics, Helsinki, Finland, vol. 3, pp. 330–332 (1990)
6. Li, F., Shi, S., Chen, Y., Lv, X.: Chinese Pronominal Anaphora Resolution Based on Conditional Random Fields. In: International Conference on Computer Science and Software Engineering, Washington, DC, USA, pp. 731–734 (2008)
7. Hendrickx, I., Hoste, V., Daelemans, W.: Semantic and Syntactic Features for Dutch Coreference Resolution. In: Gelbukh, A. (ed.) CICALing 2008. LNCS, vol. 4919, pp. 351–361. Springer, Heidelberg (2008)
8. Hobbs, J.: Resolving pronoun references. *Lingua* 44, 339–352 (1978)
9. Bradley, J.K., Guestrin, C.: Learning Tree Conditional Random Fields. In: 27th International Conference on Machine Learning (2010)
10. Joshi, A.K., Kuhn, S.: Centered logic: The role of entity centered sentence representation in natural language inferencing. In: International Joint Conference on Artificial Intelligence (1979)
11. Joshi, A.K., Weinstein, S.: Control of inference: Role of some aspects of discourse structure - centering. In: International Joint Conference on Artificial Intelligence, pp. 385–387 (1981)
12. Lappin, S., Leass, H.J.: An algorithm for pronominal anaphora resolution. *Computational Linguistics* 20(4), 535–561 (1994)
13. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: 18th International Conference on Machine Learning, pp. 282–289. Morgan Kaufmann, San Francisco (2001)
14. Lalitha Devi, S., Rao, P.R.K., Vijay Sundar Ram, Malarkodi, C., Alikadeswari, A.: Hybrid Approach for Coreference Resolution. In: Fifteenth Conference on Computational Natural Language Learning: Shared Task, Portland, Oregon, USA, pp. 93–96 (2011)
15. Lalitha Devi, S., Vijay Sundar Ram, Rao, P.R.K.: Resolution of Pronominal Anaphors using Linear and Tree CRFs. In: 8th DAARC, Faro, Portugal (2011)
16. McCarthy, J.F., Lehnert, W.G.: Using decision trees for coreference resolution. In: Mellish, C. (ed.) Fourteenth International Conference on Artificial Intelligence, pp. 1050–1055 (1995)

17. Mitkov, R.: Robust pronoun resolution with limited knowledge. In: 17th International Conference on Computational Linguistics (COLING 1998/ACL 1998), Montreal, Canada, pp. 869–875 (1998)
18. Moreau, E., Tellier, I.: The crotal SRL system: a generic tool based on tree-structured CRF. In: Thirteenth Conference on Computational Natural Language Learning: Shared Task, pp. 91–96 (2009)
19. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: 40th Annual Meeting of the Association for Computational Linguistics, pp. 104–111 (2002)
20. Ng, V.: Supervised Noun Phrase Coreference Research: The First Fifteen Years. In: ACL 2010, pp. 1396–1411 (July 2010)
21. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In: Fifteenth Conference on Computational Natural Language Learning: Shared Task, Portland, Oregon, USA, pp. 1–27 (2011)
22. Poesio, M., Kabadjov, M.: A general-purpose, off-the-shelf anaphora resolution module: implementation and preliminary evaluation. In: Language Resources and Evaluation Conference (2004)
23. Recasens, M., Hovy, E.: A Deeper Look into Features for Coreference Resolution. In: Lalitha Devi, S., Branco, A., Mitkov, R. (eds.) DAARC 2009. LNCS(LNAI), vol. 5847, pp. 29–42. Springer, Heidelberg (2009)
24. Arora, S., Lin, F., Shima, H., Wang, M., Mitamura, T., Nyberg, E.: Tree Conditional Random Fields for Japanese Semantic Role Labeling (2008) (unpublished manuscript)
25. Soon, W., Ng, H., Lim, D.: A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* 27(4), 521–544 (2001)
26. Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D., Hysom, D.: Coreference Resolution with Reconcile. In: ACL, pp. 156–161 (2010)
27. Sutton, C.: GRMM: A Graphical Models Toolkit (2006)
<http://mallet.cs.umass.edu>
28. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) *Introduction to Statistical Relational Learning*. MIT Press (2006)
29. Taku Kudo: CRF++, an open source toolkit for CRF (2005),
<http://crfpp.sourceforge.net>
30. Versley, Y., Ponzetto, S., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: BART: A modular toolkit for coreference resolution. In: Language Resources and Evaluation Conference (2008)
31. Wainwright, M.J., Jaakkola, T., Willsky, A.S.: Tree-based reparameterization for approximate inference on loopy graphs. In: NIPS, pp. 1001–1008 (2001)

Arabic Entity Graph Extraction Using Morphology, Finite State Machines, and Graph Transformations

Jad Makhoul¹, Fadi Zaraket¹, and Hamza Harkous¹

American University of Beirut
{jem04, fz11, hhh20}@aub.edu.lb

Abstract. Research on automatic recognition of named entities from Arabic text uses techniques that work well for the Latin based languages such as local grammars, statistical learning models, pattern matching, and rule-based techniques. These techniques boost their results by using application specific corpora, parallel language corpora, and morphological stemming analysis. We propose a method for extracting entities, events, and relations amongst them from Arabic text using a hierarchy of finite state machines driven by morphological features such as part of speech and gloss tags, and graph transformation algorithms. We evaluated our method on two natural language processing applications. We automated the extraction of narrators and narrator relations from several corpora of Islamic narration books. We automated the extraction of genealogical family trees from Biblical texts. In all applications, our method reports high precision and recall and learns lemmas about phrases that improve results.

1 Introduction

Named entity recognition (NER) is harder for Arabic text than it is for Latin. Arabic does not have upper case letters which distinguish the start and end of named entities (NE). Arabic is morphologically rich, more ambiguous than Latin languages, and its short vowels are frequently omitted [15].

Industrial tools that perform NE extraction for Arabic exist [23,20,13,14], but little is said about the techniques behind them and no formal evaluation of the tools exist. Researchers proposed and evaluated local grammars with morphological stemming [24,21], rule-based systems powered with local grammars [19] and stop words [2], statistical analysis boosted with Arabic-English parallel corpora [12], statistical analysis boosted with task specific gazetteers [11,10], and pattern matching powered with morphological stemming [3,17] to perform Arabic NER. We discuss these techniques and compare to them in the “Related work” Section.

In this paper, we consider the problem of NER in the context of more complex natural language processing (NLP) tasks that look for NEs and relations amongst the detected NEs and build entity graphs. An *entity graph* is a graph where NEs are nodes and relational entities (RE) are edges. For example, consider the tasks of extracting genealogical family trees based on person names from biblical text, extracting routes and maps from travel itineraries, and extracting narrator graphs based on narration documents that include narrator sequences. We hypothesize that NEs of the same kind happen at proximity of each other, and we base our method on that working hypothesis. We target

NLP applications and corpora where our working hypothesis obviously holds. These applications help build databases of NEs that we can later use in other related contexts.

We propose a method for extracting NEs and relations amongst entities using morphological features such as part of speech (POS) and gloss tags, a hierarchy of finite state machines, and graph transformations such as node merge, node split, and replace edge transformations. Morphological analysis helps reduce ambiguities and resolves lexical items with their morphological variations. For each application, we augment the lexicon of Sarf [22], an inhouse morphological analyzer, with lexical elements and tags that describe categories relevant to the application. We also build finite state transducers (FST) that take morphological tags as input and detect the NEs. Manually built FSTs can express rules that are not intuitive (and sometimes impossible) to express with local grammars using regular expressions. Consider, for example, counting the number of tags up to a threshold, or capturing left and right recursive structures which are common in Arabic since an Arabic phrase-sentence can start with a name or a verb. We finally infer relations between the NEs based on their context.

Our method takes text and computes morphological features that match, precede, follow, or connect NEs and uses the features to learn unknown NEs. Our method also computes morphological features that correspond to relations between NEs, and learns semantics for them. For example, consider a graph with two predefined edge labels l_s = “spouse” and l_p = “parent”, with names of persons n_1, n_2 , and n_3 as nodes, and with (n_1, n_2, l_s) , (n_2, n_3, l_p) , and (n_1, n_3, l_p) as edges inferred from a document. Our method can learn that the expression ولدت له *wldt lh* (gave him a child) connecting n_1 to n_2 and n_3 means that n_2 is the spouse of n_1 and the parent of n_3 . The learned semantics for the expression can be expressed as a graph transformation that leads to a graph with only l_s and l_p as labels.

We evaluated our method and used it to solve two NLP applications including extracting genealogical family trees from Biblical text, and extracting a graph of narrators from hadith books. Our results show high precision and recall metrics and report learning interesting named entities and relational lemmas. In this paper we make the following contributions.

- We present a method for named entity and entity graph extraction from Arabic text using morphology, state machines, and graph transformation algorithms.
- We present an Arabic lexicon augmented with (1) person names collected from several resources as well as learned from our corpora, and (2) tags suitable to extract complex names, kinship, and narration sequence relations.
- We learn semantics for morphological features associated with detected phrases that are equivalent to predefined features modulo one or two graph transformations. The learned semantics improved recall and precision by 5 and 6%, respectively in the genealogy application.
- We perform Arabic named entity and relational entity extraction using graph algorithms.

2 The Method

Our method takes a set of Arabic text documents and builds a graph where named entities are nodes and edges are relations between the nodes. Our target graph has

a predefined set of relations expressed as edge labels such as “parent” and “spouse” where the nodes are person names. We use Sarf, an inhouse morphological analyzer, augmented with a list of lexicon elements relevant to the application. We also augment Sarf with a set of tags that classify the elements into several categories relevant to the targeted named entities and edge labels.

The method passes the input text string to Sarf. Sarf processes the input text and whenever it identifies a morpheme, it calls the application morpheme classifier with the morpheme, the tags of the morpheme, and the current solution context. Note that several morphological solutions may exist for the same input string. The morpheme classifier either stores the morpheme in a list of unresolved morphemes, or resolves the unresolved morphemes and produces a category as input to the application FST. The application FST is manually built to detect (1) named entities, (2) sequences of named entities, (3) predefined edges connecting named entities, (4) previously unknown named entities occurring between predefined edges, and (5) previously unknown edge labels occurring between named entities. The detected named entities and edges form an entity graph.

Finally, the method uses a distance function to compute equality between the named entities and to consequently merge the nodes of the graph that represent equal named entities. The method then learns semantics for the discovered edge labels and suggests edge replacement algorithms to transform the discovered edge labels into the predefined edges. The method also interactively learns the semantics of other detected edge labels that it could not infer in terms of the predefined set of relations.

3 The Hadith Application

A حَدِيثٌ *ḥadyt* (tradition) is a narration related to the prophet Mohammad through a سَنَدٌ *sanad* or a sequence of narrators. The collections of traditions are the second source of jurisprudence after the قُرْآنٌ *qorān* for all Islamic schools of thought. Figure 1 shows an example *ḥadyt* in Arabic with its transliteration and translation. We show proper names in boxes connected to form complex names of narrators. For example, قتيبة is the first name of narrator n_1 , and سعيد is the name of his father as the word بن (son of) indicates. The sequence of names from n_1 to n_5 constitutes the *sanad* of the *ḥadyt*. The second part of the *ḥadyt* is the متن *matn* (content) and constitutes its actual content.

Al-Azami [6] cites more than eleven books of digitized tradition books each of several volumes, and a dozen other biography and secondary authentication books such as a geographical dictionary of places in hadith.

Several researchers [8,25] attempted to automate the analysis of the hadith literature. We fully automate the analysis of three books of hadith selected arbitrarily [16,4,5] 1. We accept a book as input and segment it into a vector of hadiths, and segment each hadith into its sanad and matn parts. We detect the sequence of narrators in the sanad and the relation that links each narrator to his ancestor and predecessor in the sequence. We also detect the full name of each narrator that is composed of several proper names with connectors in between.

¹ We obtained the digitized books from <http://www.yasoob.com/> and <http://www.al-eman.com/>

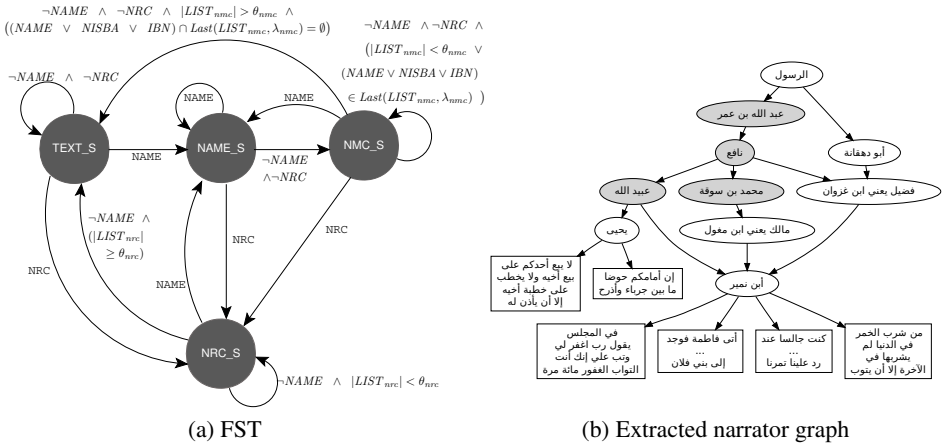


Fig. 2. FST and extracted graph from the Hadith application

The FST moves to state $NAME_S$ on $NAME$. It moves to state NRC_S when Sarf reports an NRC . State NMC_S indicates that the FST expects a name to appear within a tolerance threshold expressed by θ_{nmc} and λ_{nmc} . It returns to $NAME_S$ when a name is met, loops when still within the threshold, and resets to $TEXT_S$ when the threshold is exceeded signaling that the last collection of names met do not qualify as a narrator or a sequence of narrators.

Similarly, NRC_S tolerates θ_{nrc} words before it gives up on its expectations. Note that we reach the $NAME_S$ state only when a valid $NAME$ is detected and we leave when no more $NAME$'s are detected. The NRC_S state can only be reached if an NRC is detected. The state NMC_S can only be reached from a $NAME_S$ state.

Whenever the FST transitions to NRC_S from either $NAME_S$ or NMC_S , the FST reports a narrator detected and adds it to $LIST_{narr}$. Whenever the FST transitions back to $TEXT_S$, the FST reports a sanad detected if $|LIST_{narr}| \geq N_{min}$ where $LIST_{narr}$ is the list of detected narrators and N_{min} is a threshold denoting the minimum number of narrators required in a sanad.

We used the values of 3, 5 and 5 for θ_{nmc} , λ_{nmc} , and θ_{nrc} , respectively to obtain the results in Table 2.

3.2 Extracting the Narrator Graph Using Node Merge and Split Transformations

The hadith FST generates sequences of hadith narrators such as $\langle n_1, n_2, n_3, n_4, n_5 \rangle$ shown in Figure 1. Our target is to automate the extraction of narrator graphs, similar to the directed acyclic graph (DAG) in Figure 2(b), from one or more narration books. The nodes in boxes are the *matn* of the hadith, and the other nodes are the narrators.

Our method uses two graph transformations to build the narrator graph. It considers the graph with the disconnected sequence components, and performs a sequence of *merge* transformation for nodes that represent equivalent narrators. A merge

transformation takes two equivalent nodes n_1 and n_2 , appends the edge list of n_2 to that of n_1 , directs the incoming edges of n_2 to n_1 , and finally removes n_2 from the graph.

Then the method checks for cycles in the graph introduced by the merge transformations and breaks the cycles using node split transformations. The method also reports the cycles that require splitting nodes with high equivalency values to the user since such cycles may indicate inconsistencies or fabricated narrations.

Scholars can use the narrator graph to perform interesting queries that were not possible before such as (1) find important narrators who happen to be articulation nodes in the narrator graph, and (2) conduct what-if analysis by simulating the effect of removing a narrator on the hadith literature as several hadith with the same content may have several different narration sequences.

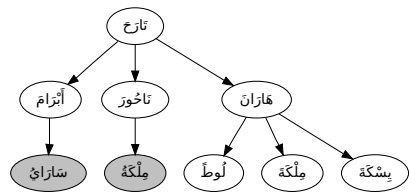
4 Genealogical Family Tree Application

Biblical genealogical lists trace key biblical figures such as Israelite kings and prophets. They used to determine rights and privileges such as reverted ownership of sold land to original owners and their descendents. Beleivers used them to compute the age of man kind, and to defend the historicity of Bilical events against other accounts they thought are myths. The example in Figure 3(a) from Genesis traces the heirs of تارح *tā-rh* (Terah). Separate genealogical lists that overlap may have consistency issues [11]. For example, Matthew and Luke report two different genealogies of Jesus. Scholars constructed partial genealogical graphs manually from several lists and their work is short from being comprehensive [9,18]. Creating a comprehensive genealogical graph from all biblical accounts greatly enhances the study of the bible. Such comprehensive graphs can be tagged with time durations between a parent and the birth of its children as is common in many biblical accounts.

Our method takes as input Arabic text with a genealogical list such as the genealogical list in Figure 3(a) and automatically extracts the genealogical family tree where person names are entities and edges represent “spouse” and “child” labels such as Figure 3(b). We marked the person names with bars above them, and we filled the spouse nodes with grey. To perform this task, our method first passes the text to Sarf, the morphological analyzer, that reports detected morphemes with their corresponding tags.

وَهَذِهِ مَوَالِيدُ تَارَحَ : وَوَلَدَ تَارَحَ أَبْرَامَ وَ نَاحُورَ وَ هَارَانَ . وَوَلَدَ هَارَانَ لُوطًا . وَنَمَاتَ هَارَانَ قَبْلَ تَارَحَ أَيُّدٍ فِي أَرْضِ مِصْرَ فِي أَوْرَ الْكَلْدَانِيِّينَ ، وَأَخَذَهُ أَبْرَامَ وَ نَاحُورَ لِأَنْفُسِهِمَا امْرَأَتَيْنِ : ائِمُّ امْرَأَتِهِ أَبْرَامَ سَارَاتِي ، وَاِئِمُّ امْرَأَتِهِ نَاحُورَ مَلِكَةُ بَنِي حَارَانَ ، أُمُّ مَلِكَةِ بَنِي يَسْكَةَ .
wahādhīhi mawāliidhu tārahā: walada tārahu ābraama wa naāhuru wa hāraana . wawalada hāraana luṭa . wamaāta hāraānu qabla taāraha raabiyhi fiy raardi miṣraādihī fiy awwi alkaaldayyīyīna . wawāḥadhu raabraāma wa naāhuru lianfusihimāi āmra raatayni : āmmu āmra raati raabraāma saāraānu , wāāmmu āmra raati naāhuruwa milkatu bintu hāraāna . raabiyhi millata wa raabiy yiskata .
 And the following are the heirs of Terah. Terah became the father of Abram, Nahor and Haran; and Haran became the father of Lot. Haran before died before the death of his father Terah in his mother land, in Ur of the Chaldeans. Abram and Nahor took wives for themselves: the name of Abrams wife was Sarai; and the name of Nahors wife was Milcah, the daughter of Haran, the father of Milcah and Iscah.

(a) Genesis 11:27-29



(b) Extracted family tree

Fig. 3. Genealogy example

We augmented the tag set of Sarf with the following categories relevant to the genealogical application.

- *NAME* a person name that can be either male or female. 509 lexicon items.
- *CHILD* a connector with a descent semantics such as ابن *ābn* (son). 12 lexicon items.
- *CHILDREN* a connector with a plural descent semantics such as أبناء *ābnā* (children). 16 lexicon items.
- *PARENT* a connector with parenthood semantics such as أباه *abāh* (his father). 7 lexicon items.
- *SPOUSE* a connector with marital partnership semantics such as سرّيته *sryth* (his concubine). 11 lexicon items.
- *SPOUSES* a connector with plural partnership semantics such as سراري *srāry* (his concubines). 7 lexicon items.
- *SIBLING* a connector with sisterhood semantics such as أخيه *aḥyḥ* (his brother). 14 lexicon items.
- *SIBLINGS* a connector with plural sisterhood semantics such as اخوته *āḥwth* (his siblings). 18 lexicon items.

Note that (1) some lexicon items might be categorized into more than one of the above categories, and (2) the method infers the gender from POS and gloss tags of the person name as well as from those of the other categories. In case of conflict, the method gives precedence to the gender inferred from the person names. Note also that the phrases اسحاق بن يعقوب *āshāq bn yqwb* (Izhac - son of - Jacob) and اسحاق ابنه يعقوب *āshāq ābnh yqwb* (Izhac - his son - Jacob) contain the same named entities, and are identical as far as stemming is concerned. However, the analysis of the suffix morphemes of ابنه *ābnh* results in inferring a parent category in the second phrase.

The method considers the detected categories and passes them to a genealogy FST. The genealogy FST distinguishes detected names as follows.

- *NAME* denotes a newly detected name that is not a node in the current tree.
- *LEAF* denotes a name that is a leaf in the current tree.
- *NODE* denotes a name that is an internal node in the current tree.

The rest of the categories are presented to the FST as is and are all denoted with *EDGE*.

Table 1 described the states, input, transitions and the actions and outputs of the genealogy FST. The FST keeps track of the last detected *NODE* and *EDGE* in the state elements n_i and e_i . It also keeps track of the number of the words in sequence that do not match a category in w . Once w passes the threshold θ , and if the FST has any nodes and edges detected, the tree is reported as output. The FST has three abstract states where *Text* is the initial state and denotes that the FST is outside the context of a genealogical list. State *Parent* denotes that the FST is in the middle of a genealogical list and has met some names. State *Child* denotes that the FST is in the middle of a genealogical list and in the context of names of children of a discovered name. When a *NAME* is met, the FST adds a node and possibly an edge to the tree. When a *NODE* or a *LEAF* is met, the FST adds an edge to the tree. When words that do not belong to the specified categories are met between *EDGE* words or between *NAME*, *LEAF*, or

Table 1. State transitions and actions for genealogy application

State	Input	Transition	Actions and output
<i>Text</i>	$n = \text{NAME}$	<i>Parent</i>	$\text{addNode}(n), n_i = n, w = 0, e_i = \text{CHILD}$
	$n = \text{LEAF} \vee \text{NODE}$	<i>Parent</i>	$n_i = n, w = 0, e_i = \text{CHILD}$
	$e = \text{EDGE}$	<i>Parent</i>	$n_i = n, w = 0, e_i = e$
<i>Parent</i>	$n = \text{NAME}$	<i>Parent</i>	$\text{addNode}(n), \text{addEdge}(n_i, n, e_i), n_i = n, w = 0$
	$n = \text{LEAF}$	<i>Child</i>	$n_i = n$
	$e = \text{EDGE}$	<i>Parent</i>	$w = 0, e_i = e$
	otherwise $\wedge w \leq \theta$	<i>Parent</i>	$w = w + 1$, add unclassified edges and nodes
	otherwise $\wedge w > \theta$	<i>Text</i>	local tree detected
<i>Child</i>	$n = \text{NAME}$	<i>Child</i>	$\text{addNode}(n), \text{addEdge}(n_i, n, e_i)$
	$n = \text{LEAF}$	<i>Child</i>	$n_i = n$
	$e = \text{EDGE}$	<i>Parent</i>	$w = 0, e_i = e$
	ENDLINE	<i>Parent</i>	$w = 0$
	otherwise	<i>Child</i>	add unclassified edges and nodes

NODE words the FST adds them as labels to unclassified edges. The FST uses these unclassified names and edges to learn edge semantics that can help in a subsequent run to resolve ambiguities; i.e. connected nodes with un-classified edges.

The FST produces a set of genealogical trees. The method takes those trees and merges equivalent nodes to build a global genealogical tree. The method merges nodes with the same names and whose merger does not cause a cycle in the genealogical tree with only the “child” and “spouse” edges considered.

5 Related Work

The iTree [7,8] tool uses a context free grammar (CFG) approach to solve the narrator extraction problem. It preprocesses the text to remove punctuation and diacritics and to normalize white spaces, and uses a similarity-based approach to memory based learning [7] in order to perform shallow parsing of the preprocessed text. The iTree CFG enumerates several stop phrases that surround names narrators. The iTree tool fails when a morphological variation of one of the stop phrases occurs. For example, Figure 4 extracted using iTree shows that iTree fails to detect مُحَمَّدًا *mḥmdā* as a name of the prophet since it is a morphological variation of it and does not stop at it when computing the sequence of the hadith: « حدثنا وكيع، حدثني سعيد بن السائب، عن داود بن أبي عاصم الثقفي، قال سألت ابن عمر عن الصلاة، بمنى فقال هل سمعت محمدا، صلى الله عليه وسلم قلت نعم وأمنت فاهتديت به قال فإنه كان يصلي بمنى ركعتين» *ḥḍṭnā wky; ḥḍṭny syd bn ālsā'yb, n dāwd bn ʔaby āšm āltqfy, qāl sʔalt ābn ʔmr n ālšlāt, bmnā fqāl hl smʔ, šlā āllh ʔyh wslm qlt nʔm wāmnt fāhtdyt bh qāl finh kān yšly bmnā rktyn'*.

We differ in that we do not preprocess the text, we use morphological analysis, and we do not base our method on stop phrases that surround narrator names. Rather, we assume that narrator names are likely to happen in localities with a high number of person names. We use narrator connectors and their morphological equivalents and thus



Fig. 4. Output of iTree fails to detect *محمدا* *mḥmdā*

we do not suffer from the “parser noise words” problem of iTree. The iTree tool extracts successfully 86.7% of the narration sequences of 90 selected traditions with 34 simple cases and 56 hard cases. We ran our method against the five hadith that ship with iTree and the two hadith from the iTree paper and we were able to extract all narration sequences even on the examples where iTree reports a “parser noise” problem.

Our NE detection method differs from local grammar based approaches [24,21] in that they enumerate several local grammar rules to capture the desired named entities, and use techniques to automatically compute finite state machines that detect the structures expressed in the local grammars. The work in [24] uses morphological stemming for Arabic with local grammars developed for Latin languages and suited to fit Arabic and reports 87% precision and 66% recall for person names. Similarly, the work in [3,17] uses morphological stemming and pattern matching. We differ in that we are not limited to the use of stems and we use the augmented set of morphological tags for all prefixes and suffixes as well. This allows us to lift the restriction of the stop words used in local grammars and in [2]. We also differ in that we directly build the FST to detect the named entities as well as the relational entities, we do not enumerate all the structures that express our targeted entities, and we are not restricted to the expressive power of a local grammar language where it is often not intuitive (and in cases not possible) to express thresholds using regular expressions. In the future, we plan to develop a local grammar language that has that expressive power.

ANERSys boosts its capabilities to detect named entities by using task specific corpora and gazetteers and ignores POS tags [11,10]. We are similar to ANERSys in that we augment the lexicon Sarf with application relevant lexicon items and tags. We differ in that they use statistical analysis models such as n-gram and maximum entropy measures. In other work, they achieve major improvements when they use lexical and morphosyntactic features and a parallel Arabic and English corpora to bootstrap noisy features [12].

We differ from rule-based systems with local grammars [19] in that we our system uses FSTs instead of rules, and in that it can learn named entities and infer semantics for entities that are equivalent within one or two graph transformations to predefined graph edge labels.

6 Results

We evaluated our method against three books of hadith [16,54], and the book of Genesis. We define our metrics, present our evaluation reference with interannotation

agreement results, and then present the results for the hadith and genealogical list applications.

Metrics. Segmentation recall refers to the ratio of the narrations correctly detected against the total number of narrations present. Segmentation precision refers to the fraction of correctly detected narration sequences compared to all the extracted narration sequences. Similarly, sanad boundary recall measures whether we detected all narrators in the sanad, while sanad boundary precision measures whether we did so without introducing false positives. The same concept applies to the narrator boundary where we count the number of words constituting a valid narrator. For the genealogy application, segmentation refers to the ratio of the genealogical lists correctly detected against the total number of lists present. Segmentation precision refers to the fraction of correctly detected genealogical lists compared to all the extracted lists. Recall for genealogical boundary detection measures whether we detected all person names in the lists; precision measures whether we did so without introducing false positives.

Interannotation. We developed a GUI tool and asked two volunteers to tag the sanad boundaries and the narrator names therein for 10% of the hadith text, and to tag the lists, the person names therein, and to edit the tree structure within the genealogy list for the Genesis. The annotators agreed with above 98% precision and recall on hadith segmentation, sanad boundaries and name boundaries. The annotators agreed with 99% and 97% recall on genealogy segmentation and list boundaries. They also scored 95% precision on list boundaries. For the genealogy segmentation, the tags of the first annotator were a subset of the tags of the second and therefore they scored 39% precision as we measured the second against the first. The Genesis contains texts that are intended to describe genealogy, and also contains person names with mentions of family relations that are not intended to be genealogical lists. The second annotator aggressively selected both types of genealogical lists, while the first annotator selected the first type and was very conservative in selecting the second type. Some genealogy lists may contain more than one family tree and our GUI was limited to allow the annotators to select one of the trees. This resulted in a 65% precision score for genealogy list boundaries agreement. The two annotators agreed on a reference tag set that we used to compute the following accuracy results.

Hadith application results. The results in Table 2 compare the results of our method against iTree 8 for the hadith application. We augmented iTree with the list of names used by our method to conduct a fair assessment. The iTree tool does not tackle the problem of segmentation as it takes as input an isolated hadith. In Table 2, instead of the segmentation columns we report the percentage of times iTree succeeded to recognize the valid hadith we pass. This measure is equivalent to recall; nonetheless, precision in this context is not applicable.

On average, our method achieved a 97% segmentation recall rate compared to 35.7% for iTree. The difference depended highly on the book under consideration. For Musnad, both methods performed well. However, iTree reported less than 5% recall for the Kafi and Istibsar books due to the less structured nature of the narrations in both books. For example, in a significant number of narrations a narrator name preceded narration words (قال *qāl* “said”, حدّث *ḥaddaṭ* “narrated”, etc...) contrary to what iTree expects.

Table 2. Hadith results compared with iTree

our method	hadith count	segmentation			sanad boundary			narrator name boundary		
		recall	precision	F-score	recall	precision	F-score	recall	precision	F-score
musnad	212	0.991	0.976	0.983	0.969	0.967	0.968	0.999	0.996	0.998
kafi	199	0.970	0.909	0.938	0.934	0.954	0.944	0.999	0.998	0.998
istibsar	189	0.968	0.973	0.971	0.946	0.945	0.945	0.999	0.998	0.999
	600	0.976	0.953	0.964	0.950	0.956	0.953	0.999	0.997	0.998

iTree	hadith count	accept as hadith			sanad boundary			narrator name boundary		
		recall	precision	F-score	recall	precision	F-score	recall	precision	F-score
musnad	62	0.999	NA	NA	0.997	0.977	0.987	0.998	0.999	0.999
kafi	23	0.043	NA	NA	0.999	0.313	0.476	0.999	0.999	0.999
istibsar	36	0.028	NA	NA	0.875	0.583	0.700	0.999	0.999	0.999
	124	0.357	NA	NA	0.957	0.624	0.721	0.999	0.999	0.999

Note that it is not trivial (if even possible) to take care of this variation using local grammars and CFGs as this introduces a high level of ambiguity. For the hadith that iTree succeeded to recognize, iTree had a lower sanad boundary precision (62.4% on the average). This is mainly due to the stop phrases used to detect the termination of sanad boundaries. Sanad termination is often implicit in Kafi and Istibsar causing iTree to extend the boundary of the sanad until the next stop phrase. Our method detects the end of the sanad when it meets a sequence of words of length $> \theta$ with no names. The narrator boundary accuracy showed similar results between iTree and our method since we augmented iTree with the names from the lexicon of Sarf.

Narrator graph. The results in Table 3(a) report the precision and recall metrics for generating the full narrator graph via merging the detected hadith sequences. For each narrator node, we computed the number of nodes it correctly merges with against the number of nodes it should merge with and averaged that across all nodes to compute the recall metric. The precision metric denotes the average of the ratio of correctly merged nodes against the number of merged nodes for each narrator node.

The “detected” row includes only the narrator nodes detected by our method and indicates the accuracy of the graph building routine isolated from the rest of the method. The “all” row includes all narrators and sanads in the text. The recall rate was 63% mainly due to the conservative cycle breaking transformation that splits merged narrator nodes even if they were equivalent. We expected higher precision values than 82% and 77% and upon examining the results, we discovered two main issues. The first issue is in the reference graph since the manual annotator decided to keep nodes split whenever he was faced with confusing cases such as the narrator nodes سعيد بن السائب *syd bn ālsā'yb*, سعيد, and سعيد بن جبیر *syd bn ġbyr*. The text included several of these examples such as سليمان التيمي *slymān āltymy*, سليمان بن رزين *slymān bn rzyrn*, and سليمان بن موسى *slymān bn mwsā*. The automated merging will make a decision to merge two of the three names in each case and uses other helper information such as the rank of the nodes in the graph to make a decision.

The second issue has to do with the degree of equality checks that the merging algorithm performs when deciding to merge narrator nodes. Consider three narrator nodes

Table 3. Narrator graph and name learning results.

narrators	recall precision F-score			names	recall precision F-score		
detected	0.673	0.824	0.741	sanad	0.649	0.940	0.768
all	0.632	0.771	0.694	all	0.794	0.733	0.762

(a) Narrator graph accuracy

(b) Name learning accuracy

Table 4. Results for Biblical genealogy family tree extraction.

	trees	nodes			neighbors			context		
		recall	precision	F-score	recall	precision	F-score	recall	precision	F-score
Before learning	local	0.583	0.898	0.707	0.778	0.760	0.769	0.552	0.588	0.570
	aligned	0.849	0.868	0.858	0.777	0.738	0.757	0.554	0.590	0.572
	full	0.784	0.800	0.792	0.652	0.652	0.652	0.536	0.573	0.554
After learning	local	0.586	0.900	0.710	0.780	0.725	0.751	0.557	0.577	0.567
	aligned	0.854	0.871	0.862	0.785	0.723	0.753	0.582	0.600	0.591
	full	0.820	0.810	0.815	0.701	0.708	0.704	0.580	0.630	0.604

merged into one node n_{merged} . Now consider merging a new narrator node n_* with n_{merged} . We currently compare n_* with only one of the constituents of n_{merged} and make the simplifying assumption that our equality metric is transitive. Comparing to more narrators in n_{merged} helps improve precision.

Learning names. Table 3(b) reports the precision and recall of the names that our method learned from the books of hadith. Within the sanad segments, our method learned 65% of the names it should learn with 94% accuracy. It learned 79% of the names it should learn from the whole text at a 73% precision. The precision is higher within the sanad since names are more frequent in the sanad context. The recall is higher for the whole text since inside the sanad our method is more likely to resolve ambiguous tags as name and narrator connectors than if outside the sanad.

Genealogical family tree detection. We report the precision and recall of the detected names in the “nodes” columns of Table 4, the accuracy of their neighboring nodes in the “neighbors” column, and the accuracy of the semantics of the edge labels (“spouse” and “child” edge labels) in the “context” columns. The table shows the results before and after learning the semantics of unclassified edges. Learned edge semantics improved the recall of the full merged graph by 5% and the precision by 6%.

The local row shows the accuracy results without aligning the local detected lists to the local annotated lists. For example, if the manual annotation has a list that spanned two automatically detected lists, we compute the accuracy for each detected list against the manually detected list alone. The aligned row shows the accuracy results with maximum boundary inclusion. In this case, we merge the automatically detected lists and compute the metrics. The full row shows the accuracy results for the tree automatically generated from merging all automatically detected lists.

Our method extracted names with 77% recall and 91% precision, and segmented genealogical lists with an 80% recall and 74% precision. The manually generated full graph had 250 nodes, while our method extracted a graph with 245 nodes before

learning, and 253 nodes after learning. The largest manually annotated local graph contained 77 nodes, while the largest extracted local graph using our method had 35 nodes.

Acknowledgements. This work is supported by grant number 11-303-0522236 from the Lebanese National Council for Scientific Research (LNCRS). We would like also to acknowledge the Hadithopedia team including Marwan Zeineddine for the technical discussions, Hassen Al-Sadr for the logistic support, and Hussein El-Asadi for his input on the hadith application.

7 Conclusion and Future Work

We presented a method to detect named entities and relations amongst them from Arabic text using morphological analysis, FSTs, and graph transformation algorithms. We implemented our method and evaluated it using two NLP applications. Our method extracted named entities with high accuracy in all applications. Our method extracted relations amongst the named entities with high accuracy in the hadith application, and with acceptable accuracy in the genealogical lists application. Our method learns named entities and semantics of named entities in terms of graph transformations that replace a detected unclassified edge in a graph with predefined edges. In the future, we plan to extend the use of our method to other applications. We also plan to enhance the learning capabilities of our method. We are also exploring building a language that enables users to intuitively design and build flexible FSTs similar to the ones presented in this paper.

References

1. Complete Bible Genealogy (2005), <http://www.complete-bible-genealogy.com>
2. Abuleil, S.: Extracting names from Arabic text for question-answering systems. In: Recherche d'Information et ses Applications (RIAO), pp. 638–647 (2004)
3. Al-Jumaily, H., Martínez, P., Martínez-Fernández, J., Van der Goot, E.: A real time named entity recognition system for Arabic text mining. In: Language Resources and Evaluation, pp. 1–21 (2011)
4. Al Kulayni, M.I.Y.: Kitab al-Kafi. Taaruf (May 1996)
5. Al Tousi, M.B.H.: Al Istibsar. Taaruf (June 1995)
6. Azami, M.M.A.: A note on work in progress on computerization of hadith. Journal of Islamic Studies 2(1) (1991)
7. Azmi, A., Bin Badia, N.: e-Narrator: an application for creating an ontology of hadiths narration tree semantically and graphically. The Arabian Journal of Science and Technology 35(2C), 86–91 (2010)
8. Azmi, A., Bin Badia, N.: iTree - automating the construction of the narration tree of hadiths. In: Natural Language Processing and Knowledge Engineering (August 2010)
9. Belote, J.: Bible Genealogies with Notes on Bible Kinship and Family Systems (2008), <http://www.d.umn.edu/~jbelote/biblegenealogy.html>
10. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition using optimized feature sets. In: Empirical Methods in Natural Language Processing, Morristown, NJ, USA, pp. 284–293 (2008)

11. Benajjiba, Y., Rosso, P., BenediRuiz, J.M.: ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh, A. (ed.) CICLEing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007)
12. Benajjiba, Y., Zitouni, I., Diab, M.T., Rosso, P.: Arabic named entity recognition: Using features extracted from noisy data. In: ACL (Short Papers), pp. 281–285 (2010)
13. Cohen, S.: Entity extraction enables “discovery”. Tech. rep., Basis Technology (2006)
14. COLTEC: ANEE: Arabic named entity extraction. Tech. rep., Computer & Language Technology (2007)
15. Debili, F., Achour, H.: Voyellation automatique de l’Arabe. In: Workshop on Computational Approaches to Semitic Languages, pp. 42–49 (1998)
16. Ibn Hanbal, A.B.: Musnad. Noor Foundation (August 2005)
17. Maloney, J., Niv, M.: TAGARAB: A fast accurate Arabic name recognizer using high-precision morphological analysis. In: Workshop on Computational Approaches to Semitic Languages (1998)
18. Rouse, R.: Mapping God’s bloodline (April 2011), <http://soulliberty.com/View.php?ID=5052>
19. Shaalan, K.F., Raza, H.: NERA: Named entity recognition for Arabic. JASIST 60(8) (2009)
20. Technologies, B.: BBN Identifinder Text Suite, <http://www.bbn.com/technology/speech/identifinder>
21. Traboulsi, H.: Arabic named entity extraction: A local grammar-based approach. In: International Multi Conference on Computer Science and Information Technology (2009)
22. Arabic text mining framework (2009), <http://code.google.com/p/atmine/>
23. Sakhr inc. (September 2009), <http://www.sakhr.com/products/Mining>
24. Zaghouani, W., Pouliquen, B., Ebrahim, M., Steinberger, R.: Adapting a resource-light highly multilingual named entity recognition system to Arabic. In: Language Resources and Evaluation Conference, Valletta, Malta (May 2010)
25. Zeineddine, M., et al.: Platform for automated authentication of Islamic traditions and hadiths (2008), <http://code.google.com/p/hadithopaedia>

Integrating Rule-Based System with Classification for Arabic Named Entity Recognition

Sherief Abdallah^{1,2}, Khaled Shaalan^{1,2}, and Muhammad Shoaib²

¹ University of Edinburgh, UK

{sherief.abdallah,khaled.shaalan}@buid.ac.ae

² British University in Dubai, UAE
shoaibhafeez@hotmail.com

Abstract. Named Entity Recognition (NER) is a subtask of information extraction that seeks to recognize and classify named entities in unstructured text into predefined categories such as the names of persons, organizations, locations, etc. The majority of researchers used machine learning, while few researchers used handcrafted rules to solve the NER problem. We focus here on NER for the Arabic language (NERA), an important language with its own distinct challenges. This paper proposes a simple method for integrating machine learning with rule-based systems and implement this proposal using the state-of-the-art rule-based system for NERA. Experimental evaluation shows that our integrated approach increases the F-measure by 8 to 14% when compared to the original (pure) rule based system and the (pure) machine learning approach, and the improvement is statistically significant for different datasets. More importantly, our system outperforms the state-of-the-art machine-learning system in NERA over a benchmark dataset.

1 Introduction

We propose and implement a simple integration between a (previously developed) rule-based system and a machine-learning classifier for Arabic named entity recognition. A named entity (NE) is a word or a phrase that contains the name of: a person, an organization, or a location among others. For example, the sentence “U.N. official Ekeus heads for Baghdad” contains three named entities: Ekeus is a person, U.N. is an organization and Baghdad is a location [19]. Named entity recognition (NER) is the task of identifying proper nouns in unstructured text. NER is usually an integral component of various Natural Language Processing applications, such as Machine Translation, Search Results clustering, and Question Answering [5]. Most NER approaches can be classified either as a rule-based (RB-NER) or a machine-learning (ML-NER) approach. The RB-NER approach relies on linguistic knowledge, in particular grammar rules, while the ML-NER approach relies on machine learning techniques. RB-NER requires handcrafted rules whereas ML-NER needs an annotated (tagged) corpus. The linguistic knowledge-based approach achieves better results in specific domains, as the gazetteers can be adapted very precisely, and it is able to detect complex entities, as the rules can be tailored to meet nearly any requirement. However, if we deal with an unrestricted domain, it is better to choose the machine learning approach, as it would be expensive (both in terms of cost and time) to acquire and/or derive rules and gazetteers in this case.

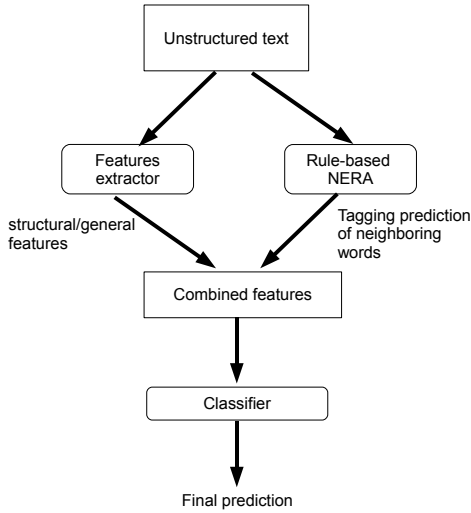


Fig. 1. Block diagram illustrating our proposed integration

The majority of the research on NER focused (naturally) on the English Language with few researchers working on other languages. This paper focuses on NER for the Arabic Language. Arabic is the official language of the ArabWorld (a population 340 million with an explosive growth) and the language of the Quran (the Islamic holy-book, therefore affecting 1.41-1.57 billion Muslims). Arabic is rich in morphology and syntax. Despite the influence of the Arabic language, the research in NER for Arabic is still in its early phases. A major reason for this lag is the lack of available tools (such as taggers and word level analyzers) and linguistic resources (such as named entity tagged corpora and gazetteers). Moreover, the Arabic language is highly challenging to deal with when it comes to perform linguistic grammar based processing. For example, Arabic does not have capital letters; a very important feature in identifying proper nouns. Also it is normally written with optional diacritics (such as short vowels or shadda) which leads to different types of ambiguity in Arabic texts (both structural and lexical), because different diacritics represent different meanings. We describe these challenges in detail in the background section.

We propose in this paper an integration of a rule-based NERA (NER for Arabic) approach, and a machine learning classification approach, as depicted in Figure 1. From the unstructured text, two sets of features are extracted for each word. The first set, which we call the rule-based features, consists of the NE tags predicted by the rule based component for the word in question and a window of surrounding words. The second set of features are general features that are based on our experience.

We verify through extensive experimental results that by complementing the human expertise (through the rule-based component) with automatic fine tuning (through traditional classifiers such as decision trees) we were able to achieve 8-12% improvement over the state-of-the-art NERA system (which used conditional random fields [5]). Interestingly, we also show that relying only on rule-based features does not improve

performance. Also relying on general features does not improve performance (actually leads to a degrading performance). Only when both sets of features are combined does machine-learning classifiers out-performs the-state-of-the-art. These results confirm the value of the integration between RB-NER and ML-NER.

2 Background

In this section we provide the necessary background to understand our contribution. First we give brief overview of Arabic NER, then we describe the rule-based NER for Arabic system which we used as a component in our architecture.

2.1 Arabic Named Entity Recognition

The concept of Name Entity Recognition was born in Message Understanding Conferences in 1990s. In Sixth Message Understanding Conference¹ held in November 1995, the NER task was formally broken down into three subtasks. These subtasks included:

Named Entities - ENAMEX tag. To identify proper names including Person, Organization and Location Names.

e.g. <ENAMEX TYPE="LOCATION">North< /ENAMEX>

Temporal Expression - TIMEX tag. To identify absolute temporal expressions including Date and Time.

e.g. <TIMEX TYPE="DATE">fiscal 1989< /TIMEX>

Number Expression - NUMEX tag. To identify two type of numeric expressions including Money and Percentage.

e.g. <NUMEX TYPE="MONEY">\$42.1 million< /NUMEX>

As mentioned earlier, in this work we focus primarily on Arabic NER. The Arabic language has several distinctive challenges when compared to Latin languages [12][17][18]:

Complex Morphology. Arabic is a highly inflected language. Words are formed using stem or root, with prefixes and suffixes characters. This concatenative strategy to form words in Arabic causes data sparseness; hence this peculiarity of the Arabic language poses a great challenge to NER systems [17].

Lack of Capital Letters. Arabic language lacks the capital letters and thus other heuristics have to be applied for detecting Named Entity boundaries such as preceding or succeeding indicator words [17][18].

Non Standard Written Text. The translated and transliterated words to Arabic are not standardized. This is problematic as most of the time all possible spelling variants are not possible to take into consideration [12].

Ambiguity and lack of Diacritization. The written Arabic lacks the Diacritics (short vowels) [2]:

"As most Arabic texts that appear in the media (whether in printed documents or digitalized format) are undiacritized, restoring diacritics is a necessary step for various NLP tasks that require disambiguation or involve speech processing."

¹ <http://cs.nyu.edu/cs/faculty/grishman/muc6.html>

Missing diacritics are not the only problem. The Arabic words can have different meanings in different contexts which increases the complexity of Named Entity Recognition Systems.

Lack of Resources. The lack of resources for Arabic NER is the major reason of the research in this field being in its infancy. Most of the available resources are either very costly or are of low quality. Thus researchers have to build up their own resources. The lack of using standardized resources thus creates problem of comparing performance among different systems.

We have used the following two corpora for data acquisition and system evaluation (training/testing our classification component):

1. The ACE 2003 Multilingual Training Set¹
2. ANERcorp Corpus Prepared by Yassine Benajiba².

ACE stands for Automatic Content Extraction, a technology that supports automatic processing of human language in textual form.³ ACE 2003 Multilingual Training Set corpus is distributed by Linguistic Data Consortium (LDC) under the Catalog number LDC2004T09 and ISBN 1-58563-292-9. ACE provides several different files in Standard Generalized Markup Language (SGML) format. These files contain data from Broadcast News and Newswire articles. Each data file in ACE corpus has corresponding XML file which provides Entity information for words in data file. The Entity types covered by ACE 2003 data includes Person, Organization, Location, Facility and Geo Political Entity (GPE). ANERcorp is a corpus prepared by Yassine Benajiba for Named Entity Recognition Task in Arabic Language. With more than 150,000 words annotated for Named Entity Recognition, ANERcorp is ideal for Machine Learning based system as large annotated text is required for better Machine Learning. The details of ANERcorp corpus along with parsing information is described in [8] and [6]. The ANERcorp is easy to parse as each line contains single word with its Entity Information (the corpus is tagged in CONLL format). The possible entity information attached to each tag as described in is listed below:

O Words that are not named entities and referred to as 'Other'.

B-PERS Beginning of Person Name

I-PERS Inside of Person Name

B-ORG Beginning of Organization Name

I-ORG Inside of Organization Name

B-LOC Beginning of Location Name

I-LOC Inside of Location Name

B-MISC Beginning of Miscellaneous Word

I-MISC Inside of Miscellaneous Word

In order to utilize Corpora described in previous sections, we transformed them into XML format using JAVA code. Only Person, Organization and Location entities are

¹ Available to BUID under License.

² Available for download from <http://users.dsic.upv.es/ybenajiba/>

³ <http://www.itl.nist.gov/iad/mig/tests/ace/>

taken into consideration from source corpora during transformation, while other entity types are ignored. For ACE Training set all the files were parsed and transformed into two XML files, one for Broadcast News data and other for Newswire data. All the data of ANERcorp was transformed into single XML file. The XML format is in compliance with the NERA system specification, the rule-based system for Arabic NER that we use in our study. The following section describes the NERA system.

2.2 The NERA System

We have previously developed Named Entity Recognition for Arabic (NERA) prototype. As a proof of concept, we here focus on only three named entities (person name, location, and organization) of those.

We have reimplemented the NERA system [17,18] using the GATE platform⁴. NERA was a rule-based approach for recognizing the most important categories of named entities in Arabic script. The NERA system required a whitelist (gazetteer), a parser and a filtration mechanism. The recognition process included the following two steps: 1) A lookup procedure, called Whitelist, that performed the recognition based on a lookup gazetteer containing lists of known named entities, and 2) A parser, based on a set of grammar rules (represented as regular expressions) derived by analyzing the local lexical context. The Whitelists are fixed static gazetteers (dictionaries) of Named Entities that are matched with target text irrespective of the rules. The exact matches of target text with Whitelist gazetteer entries are reported as Named Entities. Sample entries in gazetteer are shown in Table 1.

Table 1. Sample Data in Gazetteers

Complete Names	حسن نصر الله	محمد سعيد	كوفي أنان
	Hassan Nasar Allah	Muhammad Saeed	Kofi Anan
First Names	عبدالله	عمر	إسحاق
	Abdullah	Umar	Ishaq
City Names	مسقط	الطائف	شيكاغو
	Muscat	Taif	Chicago
Prefix Business	الزراعية	التجارية	الصناعية
	Agricultural	Commercial	Industrial

The parser of the NERA system consisted of pattern matching rules that encapsulated linguistic expertise. The rules were based on regular expressions and utilized several different dictionaries within the rules. The Parser was a vital resource as it can deal with peculiarities and complexity of the Arabic language. For instance, the Parser can largely deal with the lack of capitalization for proper nouns by means of using indicator words for named entities. These indicators were used to formulate recognition rules. The NE indicators were obtained as a result of a thorough contextual analysis of various

⁴ <http://gate.ac.uk/>

Arabic scripts. The indicators formed a window around a named entity, which helped in identifying Named Entities without being recognized itself.

3 The Proposed Integrated Approach

Our integration is done by feeding the output of the rule-based system as features to machine-learning classifiers. We call these features the rule-based features. These features are then complemented with other general features that we added through experience. We call the latter features the machine-learning features. We have used Stanford POS Tagger⁵ to compute some of these features, such as word category and affixation. All features are then combined and fed to a classifier. We have evaluated several classifier and the results were very similar. We will focus on the decision tree classifier, because its model is easy to understand. Figure 1 illustrates the idea. The features we have used are defined as follows.

Rule-based features. The Named Entity tags from NERA system are used as features. An N-word sliding window (in the experiments we used N=5) is used for each word in corpus. Thus for every word its own tag along with the tag for two left neighbors and two right neighbors are used. Table 2 provides sample instances of these features for 3 words.

Machine-learning features Word-Length. A boolean feature which is TRUE if the word length is greater than three and FALSE otherwise. As pointed out by [10] that very small words are rarely Named Entities.

Noun-Flag. A boolean feature which is TRUE if the part of speech Tag is Noun and FALSE otherwise.

Speech-Tag. Part of speech tag for the current word.

Type-Current. three boolean features, indicating whether the current word is present in the Person Gazetteer, the Organization Gazetteer, or the Location Gazetteer.

Type-Left. similar to Type-Current but for the word to the left of the current word.

Type-Right. similar to Type-Current but for the word to the right of the current word.

Statement-End. A Troolean feature whose value is 1 if the left neighbor of current word is full stop '.', 2 if the right neighbor of current word is full stop '.' and 3 otherwise.

Prefix-Suffix. Prefix of length one for current word, suffix of length one for current word, prefix of length two for current word, and suffix of length two for current word.

4 Experimental Results

Table 3 summarizes the statistical tests of the F-measure that we have conducted, using the J48 decision tree classifier.⁶ It is interesting to see that the results are consis-

⁵ available at <http://nlp.stanford.edu/software/stanford-postagger-2010-05-26.tgz>

⁶ J48 is an implementation of C4.5 Algorithm for decision trees [16].

Table 2. Sample Rule based features for 5 Word Window

Word	NMinusTwo	NMinusOne	N	NPlusOne	NPlusTwo
الرئيس	OTHER	OTHER	OTHER	OTHER	Person
الروسي	OTHER	OTHER	OTHER	Person	Person
فلاديمير	OTHER	OTHER	Person	Person	OTHER
بوتين	OTHER	Person	Person	OTHER	OTHER

tent across datasets. The rule-based system NERA is at least as good as the Machine-learning approach that uses only rule-based features (MLR) or only our proposed features (ML). However, the Machine-learning approach is significantly better than NERA when all features are used (Hybrid). The results are statistically significant.

Table 3. The F-measure performance using the pure rule-based system (NERA) as a reference point. For example, the first row compares the F-measure performance between NERA and ML approaches. The first column shows that mean difference in F-measure between NERA and ML approaches ($F(\text{NERA}) - F(\text{ML}) = 6.15$), which means that NERA outperforms ML for the first dataset (positive difference). The second column shows the statistical significance (a difference is statistically significant if the two-tail probability is less than 0.05, and lower is better).

	Mean Difference	Two Tail Probability	95% Confidence Interval
ANERcorp Data			
F(NERA) - F(ML)	6.15	0.0011	(-3.2,-9.11)
F(NERA) - F(MLR)	-1.03	0.44956	(-1.91,3.97)
F(NERA) - F(Hybrid)	-8.68	0.000089	(5.75,11.61)
ACE Newswire Data			
F(NERA) - F(ML)	3.14	0.016353	(-5.54,-0.73)
F(NERA) - F(MLR)	2.6	0.137472	(-6.2,1.01)
F(NERA) - F(Hybrid)	-15.48	0.0077	(5.23,25.73)
ACE Broadcast News Data			
F(NERA) - F(ML)	0.7	0.745468	(-5.44,4.04)
F(NERA) - F(MLR)	2.83	0.18167	(-7.26,1.59)
F(NERA) - F(Hybrid)	-6.55	0.0049	(2.55,10.55)

Table 4 compares the results of our integrated approach to previously reported results of the state-of-the-art approach for NERA [7]. We can see that our approach is significantly better for both the person and organization named entities, while our approach has comparable performance in case of the location NE.

An interesting question that we have investigated is why and when our approach disagrees with the RB-NER. To answer this question we investigate here in more depth the resulting decision tree. An example tree that we have obtained from the J48 classifier [16] with all the features described earlier and when applied on ANERcorp Data [8] consists of 1126 leaves and the size of the tree is 1684 nodes in total. Figure 2 shows

Table 4. Comparison of F-measure performance between our proposed hybrid approach and the conditional random fields approach

	Person			Organization			Location			Mean
	P	R	F	P	R	F	P	R	F	F
Our integrated approach	94.9	90.78	92.8	86.26	85.99	86.12	90.6	84.4	87.39	88.77
Conditional random fields	80.41	67.42	73.35	84.23	53.94	65.76	93.03	86.67	89.74	76.28

the subtree where top node N (the type predicted by the RB-NER) has the value Organization. This subtree is interesting because it shows cases of disagreements, where the final class in some cases is Location. Consider an example where word ألمانيا (Germany) is shown with three tags and few words surrounding to the left and right of ألمانيا in the ANERcorp Dataset:

فرانكفورت (د ب أ) أعلن اتحاد صناعة السيارات في
 ألمانيا Location_ Location_ Organization_ ألمانيا
 امس الاول أن شركات صناعة السيارات في ألمانيا تواجه ...

Translation: “**Frankfurt, Auto Industry Association in Germany said the day before yesterday that Automakers in Germany is facing ...**”

In this example the word ألمانيا is followed by first tag “Organization” which is recognized by rule based system. “Location” is the second tag for word ألمانيا and is identified by Decision Tree. The final tag is actual tag in corpus for word ألمانيا and it is also “Location”. As per the actual tagging in corpus i.e. “Location”, the recognition of word ألمانيا as “Organization” is incorrect by rule based system. The order of tree traversal is given in Figure 2 to correctly classify the word as “Location”. The values of the features (used in Decision Tree) for this word are N=Organization, isLookupOrganization = FALSE, NPlusOne = OTHER, Prefix=2, NMinusOne = Organization, Actual=Location. Another similar example is given below:

تحقيق السلام في دارفور . الظواهري قال إن حكومة
 الخرطوم Location_ Location_ Organization_ الخرطوم
 عاجزة عن حل أزمة دارفور (رويترز) وكان ...

Translation: “**Achieving peace in Darfur. Al-Zawahiri said that the Khartoum government is powerless to solve the Darrfor crisis (Reuters) and was ...**”

In this example also the recognition of the word الخرطوم (Khartoum) by rule based system as “Organization” is incorrect as it is tagged as “Location” in reference corpus. The correct classification of the word الخرطوم is given by Decision tree as “Location”. The the order of tree traversal for is given in Figure 2 to correctly classify this word as “Location”. The values of features (used in Decision Tree) for this word are N = Organization, isLookupOrganization = FALSE, NPlusOne = OTHER, Prefix = 2, NMinusOne = Organization, Actual=Location.

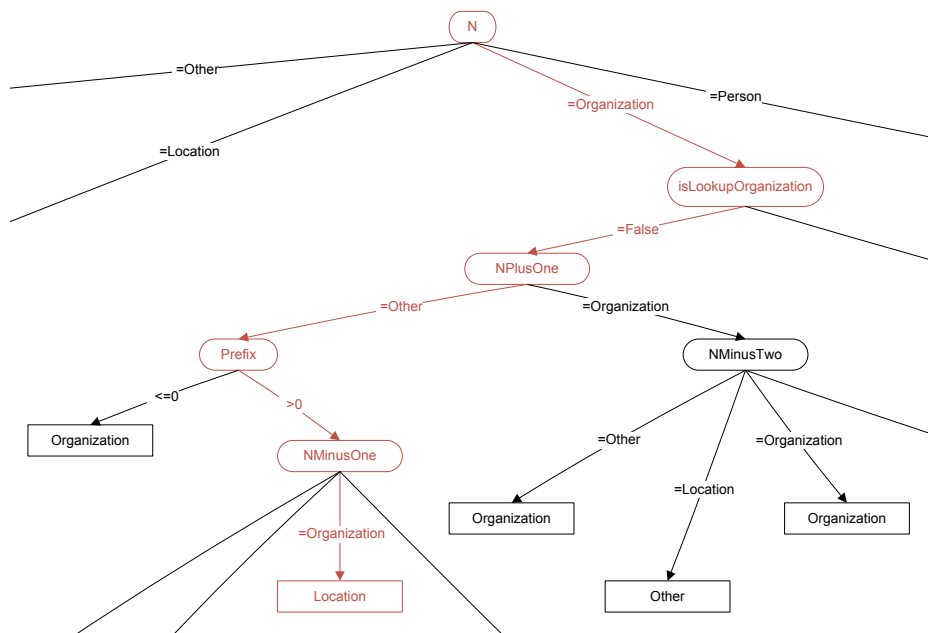


Fig. 2. Part of the decision tree learned by the J48 algorithm for the ANERCorp using all the features. Highlighted path corresponds to the example in the text.

We were initially surprised that the rule-based system was not able to correctly recognize the above Location NEs. However, upon further investigation we have learned that the errors of NERA (the rule-based system) above are actually corpus specific. In the original corpus where NERA rules were developed, an Organization NE would include the organization's location. However, in the ANERcorp corpus an organization location is considered a separate Location NE. Our hybrid approach was successfully able to adapt the rules to account for these differences across corpora.

5 Related Work

As we have mentioned earlier, the work in NER generally fell under either rule-based or machine-learning approaches. Rule based systems allowed expert linguists to handcraft rules for the NER task. This encoding of human expertise required extensive work from expert linguists and usually targeted a single language. As a result, only few researchers used rule-based systems to tackle NER for Arabic. The rules were implemented as regular expression for pattern matching mostly in conjunction with list of lookup gazetteers.

TAGARAB [13] was one of the early systems that used rule based pattern matching for NER in Arabic. TAGARAB used morphological analysis of text in conjunction with pattern matching to achieve higher accuracy as compared to simple pattern matching. Another work discussed the application of local grammar based approach in domain

of Arabic language [20]. The grammar was extracted by applying corpus analysis over range of untagged Arabic corpora. The result was a finite state automata to extract named entities from Arabic text.

Machine Learning is the mostly applied method for NER for all major languages including Arabic. NER was viewed as classification problem, where text features are used to classify words either as a particular NE or as normal text. The features include both language specific features (e.g. Part of Speech information, Morphological features etc) and language independent features (e.g. length of the word etc). A major shortcoming of the machine learning approach is requiring large corpora of annotated text. This shortcoming is more magnified in Arabic NER due to the lack of linguistic resources. Our integrated approach complements this limitation with the human expertise encapsulated in the rule-based component.

One of the early attempts to utilize Machine Learning for NER [3] used word-level features, dictionary Look-Up, part-of-speech tags and punctuation. The reported accuracy of the system was comparable to state-of-the-art rule-based system at the time. ANERSys, an NER System, was based on Maximum Entropy [8]. The baseline results was acquired by assigning each word in the test set a class that was most frequently assigned to it in the training set. Later the training and testing were done using the Maximum Entropy approach. The authors reported significant improvement over baseline results.

The work of ANERSys was extended to ANERSys 2.0 [6]. The approach used Maximum Entropy along with part of the speech information. The same baseline was used as in ANERSys. The authors [6] reported significant improvement over the baseline results, results from ANERSys and results from demo version of Siraj (Sakhr) which is a commercial system for Named Entity Recognition. Using Conditional Random Fields instead of Maximum Entropy for ANERSys system resulted in further improvement [7]. A similar approach used leading and trailing character n-grams in words as features [1], which reported better performance over previous work. Support vector machines with very large number of features were also used [14,11,4,5] but suffered from (very) slow training time and could not incorporate human knowledge if available.

Hybrid approaches combined hand crafted rule based system and Machine Learning system (our approach falls in this category). A recent hybrid approach applied Maximum Entropy (ME) with Hidden Markov Model (HMM) followed by rules to detect NE [9]. Our approach, on the other hand, uses RB-NER component followed by ML-NER. Perhaps the most similar work to our approach used rule-based systems to provide training labels [15]. In the first stage, authors passed text through the rule-based system to tag the words. The tags were then used as the ground truth for training a classifier. In other words, they did not use previously-labeled corpus. In the evaluation stage, text was tagged, independently, by both the Rule Based System and the classifier. Cases of disagreement were presented to an expert Linguist. Unlike our approach, there was no expert-tagged corpus that was utilized in the training phase of Machine Learning Model.

6 Conclusion

We have proposed in this paper an architecture for Arabic Named Entity Recognition that integrates rule-based with machine learning classifiers. As a proof of concept, we

have re-implemented a rule-based system and then integrated it with a decision-tree classifier. Experimental results confirm that our hybrid approach is significantly better than the pure rule-based system or the pure machine-learning classifier. Our approach is also better than the state-of-the-art Arabic NER (which relied on conditional random fields). Our hybrid approach was successfully able to adapt the rules to account for the tagging differences across corpora.

References

1. Abdul Hamid, A., Darwish, K.: Simplified feature set for arabic named entity recognition. In: Proceedings of the 2010 Named Entities Workshop, pp. 110–115. Association for Computational Linguistics, Uppsala (2010), <http://www.aclweb.org/anthology/W10-2417>
2. Attia, M., Toral, A., Tounsi, L., Monachini, M., van Genabith, J.: An automatically built named entity lexicon for arabic. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta (May 2010)
3. Baluja, S., Mittal, V.O., Sukthankar, R.: Applying machine learning for high performance named-entity extraction. *Computational Intelligence* 16(4), 586–595 (2000)
4. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition: An svm-based approach. In: The International Arab Conference on Information Technology, ACIT 2008 (2008)
5. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition using optimized feature sets. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 284–293. Association for Computational Linguistics, Morristown (2008)
6. Benajiba, Y., Rosso, P.: Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. In: IICAI, pp. 1814–1823 (2007)
7. Benajiba, Y., Rosso, P.: Arabic named entity recognition using conditional random fields. In: Workshop on HLT & NLP within the Arabic World. Arabic Language and Local Languages Processing: Status Updates and Prospects (2008)
8. Benajiba, Y., Rosso, P., Benedi Ruiz, J.M.: ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh, A. (ed.) CICLEing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007)
9. Biswas, S., Mishra, S.P., Acharya, S., Mohanty, S.: A hybrid oriya named entity recognition system: Harnessing the power of rule. *International Journal of Artificial Intelligence and Expert Systems (IJAE)* 1, 1–6 (2010)
10. Ekbal, A., Bandyopadhyay, S.: Voted ner system using appropriate unlabeled data. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, NEWS 2009, pp. 202–210. Association for Computational Linguistics, Morristown (2009)
11. Ekbal, A., Bandyopadhyay, S.: Named entity recognition using support vector machine: A language independent approach. *International Journal of Electrical, Computer, and Systems Engineering* 4(2), 155–170 (2010)
12. Habash, N.Y.: Introduction to Arabic Natural Language Processing. Morgan & Claypool Publisher (2010)
13. Maloney, J., Niv, M.: Tagarab: a fast, accurate arabic name recognizer using high-precision morphological analysis. In: Proceedings of the Workshop on Computational Approaches to Semitic Languages, Semitic 1998, pp. 8–15. Association for Computational Linguistics, Morristown (1998)

14. Mayfield, J., McNamee, P., Piatko, C.: Named entity recognition using hundreds of thousands of features. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 184–187. Association for Computational Linguistics, Morristown (2003), <http://dx.doi.org/10.3115/1119176.1119205>
15. Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D.: Using machine learning to maintain rule-based named-entity recognition and classification systems. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL 2001, pp. 426–433. Association for Computational Linguistics, Morristown (2001)
16. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
17. Shaalan, K., Raza, H.: Arabic Named Entity Recognition from Diverse Text Types. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 440–451. Springer, Heidelberg (2008)
18. Shaalan, K., Raza, H.: NERA: Named entity recognition for arabic. *Journal of the American Society for Information Science and Technology*, 1652–1663 (2009)
19. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 142–147. Association for Computational Linguistics, Stroudsburg (2003), <http://dx.doi.org/10.3115/1119176.1119195>
20. Traboulsi, H.: Arabic named entity extraction: A local grammar-based approach. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 4, pp. 139–143 (2009)

Space Projections as Distributional Models for Semantic Composition

Paolo Annesi, Valerio Storch, and Roberto Basili

Department of Enterprise Engineering,
University of Roma Tor Vergata, Roma, Italy
{annesi,basili}@info.uniroma2.it, storch@uniroma2.it

Abstract. Empirical distributional methods account for the meaning of syntactic structures by combining word vectors according to algebraic operators. In this paper, a novel approach for semantic composition based on space projection techniques over lexical vector representations is proposed. In line with the principle of compositionality, the meaning of a phrase is modeled in terms of the subset of properties shared by co-occurring words. Syntactic bi-grams are thus projected in the so called *Support Subspace*, corresponding to such properties. State-of-the-art results are achieved in a well known phrase similarity task, used as a benchmark for this class of methods.

1 Introduction

While compositional approaches to language understanding have been largely adopted, semantic tasks are still challenging for Natural Language Processing. Traditional logic-based approaches (as the Montague’s approach in [1] and [2]) rely on the *Principle of Compositionality* for which the meaning of a sentence is a function of the meanings of its parts. The resulting theory allows an algebra on the discrete propositional symbols to represent the meaning of arbitrarily complex expressions. More recently, distributional models of lexical semantics have been proposed (e.g. Firth [3] or Schütze [4]) based on Wittgenstein’s later philosophy, whereby the lexical meanings is determined by the context of use [5]. This seems to be a completely orthogonal *view* on meaning representation with respect to logical models.

Computational models of semantics based on symbolic logic representations account naturally for the meaning of sentences, through the notion of compositionality for which the meaning of complex expressions can be determined by using the meanings of their constituents and the rules to combine them. Despite the fact that they are formally well defined, logic-based approaches have limitations in the treatment of ambiguity, vagueness and cognitive aspects intrinsically connected to natural language. Inherent limitations are the inadequate tools to model and overcome the uncertainty in the interpretation of specific phrases, such verb-object or adjective-noun pairs.

Distributional models early introduced by Schütze [6] and recently surveyed in [7] rely on the Word Space model, inspired by Information Retrieval. They

manage semantic uncertainty through mathematical functions grounded in probability or vector spaces. Points in the space represent semantic concepts, such as words, and can be learned from corpora, in such a way that similar, or related, concepts are near to one another in the space. The distance between two points (via angular or Euclidean metrics) represents semantic dissimilarity between concepts. Methods for constructing representations for phrases or sentences through vector composition has recently received a wide attention in literature (e.g. [8]). However, vector-based models typically represent isolated words and ignore grammatical structure [7]. They have thus a limited capability to model compositional operations over phrases and sentences.

In order to overcome these limitations a so-called compositional distributional semantics (DCS) model is needed and its development is still object of ongoing and controversial research (e.g. [9], [10]). A compositional model based on distributional analysis should provide semantic information consistent with the meaning assignment typical of human subjects. For example, it should support synonymy and similarity judgments on phrases, rather than only on single words. The objective should be a measure of similarity between quasi-synonymic complex expressions, such as "... *buy a car* ..." vs. "... *purchase an automobile* ...". Another typical benefit should be a computational model for entailment, so that the representation for "... *buying something* ..." should be implied by the expression "... *buying a car* ..." but not by "... *buying time* ...". Distributional compositional semantics (DCS) needs thus a method to define: (1) a way to represent lexical vectors \mathbf{u} and \mathbf{v} , for words u, v dependent on the phrase (r, u, v) (where r is a syntactic relation, such as verb-object), and (2) a metric for comparing different phrases according to the selected representations \mathbf{u}, \mathbf{v} .

Existing models are still controversial and provide general algebraic operators (such as tensor products) over lexical vectors. By focusing on the geometry of latent semantic spaces a novel distributional model for semantic composition is proposed. The aim is to model semantics of syntactic bigrams as projections in lexically-driven subspaces. Distances in such subspaces (called *Support Spaces*) emphasize the role of *common* features that constraint in "parallel" the interpretation the involved lexical meanings and better capture phrase-specific aspects. While Section 2 discusses existing methods of compositional distributional semantics, Section 3 presents our model based on support spaces. Experiments in Section 4 are used to show the beneficial impact of the proposed model.

2 Related Work

While compositional semantics allow to govern the recursive interpretation of sentences or phrases, vector space models (as in IR [11]) and, mostly, semantic space models, such as LSA [12][13], represent lexical information in metric spaces where individual words are represented according to the distributional analysis of their co-occurrences over a large corpus.

Distributional models are based on the theory that words occurring within similar contexts are semantically similar (Harris in [14]). Words are represented

as vectors and their meaning is distributed across many dimensions. Word meaning is obtained empirically by examining the contexts in which a word appears. The meaning of a word w corresponds strictly to the distributional context in which w occurs, i.e. depends on the contexts distribution it shares with other words. Vector components reflect the corresponding contexts so that two words close in the space are systematically found in similar contexts.

Semantic spaces have been widely used for representing the meaning of words or other lexical entities (e.g. [7]), with successful applications in lexical disambiguation [4] or harvesting thesauri (as in Lin [15]). In this work we will refer to the so-called **word-based spaces**, in which target words are represented by gathering probabilistic information of their co-occurrences calculated in a fixed range window over all sentences. In such that models vectors components correspond to the entries f of the vocabulary V (i.e. to features that are individual words). In some works (e.g. [8]) pure co-occurrence counts are adopted as weights for individual features f_i , where $i = 1, \dots, N$ and $N = |V|$; in other works (e.g. [16]), weights are the pointwise mutual information scores between the target word w and the captured co-occurrences in the window,

$$pmi(w, i) = \log_2 \frac{p(w, f_i)}{p(w) \cdot p(f_i)} \quad i = 1, \dots, N$$

A vector $\mathbf{w} = (pmi_1, \dots, pmi_N)$ for a word w is thus built over all the words f_i belonging to the dictionary. When w and f never co-occur in any window their pmi is by default set to 0. Weights of vector components depend on the size of the co-occurrence window and express the global statistics in the entire corpus. Larger values of the adopted window size aim to capture *topical similarity* (as in the document based models of IR), while smaller sizes (usually between the $\pm 1-3$ surrounding words) lead to representation better suited for *paradigmatic similarities* between word vectors \mathbf{w} . Cosine similarity between vectors \mathbf{w}_1 and \mathbf{w}_2 is modeled as the normalized scalar product, i.e. $\frac{\langle \mathbf{w}_1, \mathbf{w}_2 \rangle}{\|\mathbf{w}_1\| \|\mathbf{w}_2\|}$ that expresses *topical* or *paradigmatic similarity* according to the different representations (e.g. window sizes). Notice that dimensionality reduction methods, such as LSA [12,13] are also applied in some studies, to capture second order dependencies between features f , i.e. applying semantic smoothing to possibly sparse input data. Applications of an LSA-based representation to Frame Induction or Semantic Role Labeling are presented in [17] and [18], respectively.

The main drawback of the above models is their non-compositional nature: they ignore the grammatical structure underlying phrases, such as "... buy a car ..." that are thus not clearly connected to the base vectors \mathbf{w}_{buy} and \mathbf{w}_{car} . Distributional methods hence can not compute the meanings of phrases (and sentences) as efficiently as they do indeed over words.

2.1 Distributional Compositional Semantic Models

Distributional methods have been thus recently extended to better account compositionality, in the so called distributional compositional semantics (DCS)

approaches. Mitchell and Lapata in [8] follow Foltz [19] and assume that the contribution of syntactic structure can be ignored, while the meaning of a phrase is simply the *commutative sum of the meanings of its constituent words*. More formally, [8] defines the composition $\mathbf{p}^\circ = \mathbf{u} \circ \mathbf{v}$ of vectors \mathbf{u} and \mathbf{v} through an additive class of composition functions expressed by:

$$\mathbf{p}^+ = \mathbf{u} + \mathbf{v} \quad (1)$$

This perspective clearly leads to a variety of efficient yet shallow models of compositional semantics compared in [8]. For example pointwise multiplication is defined by the multiplicative function:

$$\mathbf{p}^\cdot = \mathbf{u} \odot \mathbf{v} \quad (2)$$

where the symbol \odot represents multiplication of the corresponding components, i.e. $p_i = u_i \cdot v_i$. Since the cosine similarity function is insensitive to the vectors magnitude, in [8] a more complex asymmetric type of function called *dilation* is introduced. It consists in multiplying vectors \mathbf{v} by the quadratic factor $\mathbf{u} \cdot \mathbf{u}$ and \mathbf{v} by a stretching factor λ as follows: $\mathbf{p}^d = (\mathbf{u} \cdot \mathbf{u})\mathbf{v} + (\lambda - 1)(\mathbf{u} \cdot \mathbf{v})\mathbf{u}$. Notice that either \mathbf{u} can be used to dilate \mathbf{v} , or \mathbf{v} can be used to dilate \mathbf{u} . The best dilation factor λ for the dilation models is studied and tuned in [8]. Dilation and point-wise multiplication seem to best correspond with the intended effects of syntactic interaction, as experiments in [8] demonstrate.

In [20], the concept of a *structured vector space* is introduced, where each word is associated to a set of vectors corresponding to different syntactic dependencies. Every word is thus expressed by a tensor, and tensor operations are imposed.

The main differences among these studies lies in (1) the lexical vector representation selected (e.g. some authors do not even commit to any representation, but generically refer to any lexical vector, as in [10]) as well as in (2) the adopted compositional algebra, i.e. the system of operators defined over such vectors. In most work, operators do not depend on the involved lexical items, but a general purpose algebra is adopted. Since compositional structures are highly lexicalized, and the same syntactic relation gives rise to very different operators with respect to the different involved words, a proposal that makes the compositionality operators dependent on individual lexical vectors is hereafter discussed.

3 A Quantitative Model for Compositionality

Let's start to discuss the above compositional model over an example, where we want to model the semantic analogies and differences between "... buy a car ..." and "... buy time ...". The involved lexicals are *buy*, *car* and *time*, while their corresponding vector representation will be denoted by \mathbf{w}_{buy} , \mathbf{w}_{car} and \mathbf{w}_{time} . The major result of most studies on DCS is the definition of the function \circ that associates to \mathbf{w}_{buy} and \mathbf{w}_{car} a new vector $\mathbf{w}_{buy-car} = \mathbf{w}_{buy} \circ \mathbf{w}_{car}$.

We consider this approach misleading since vector components in the word space are tied to the syntactic nature of the composed words and the new vector

w_{buy_car} should not have the same type of the original vectors. Mathematical operations between the two input vectors (e.g. point wise multiplication \odot as in Eq. 2) produce a vector for a structure (i.e. a new type) that possesses the same topological nature of the original vectors. As these latter are dedicated to express arguments, i.e. a verb and its object in the initial space, the syntactic information (e.g. the relation and the involved POS) carried independently by them is neglected in the result. For example, the structure "... *buy a car* ..." combines syntactic roles that are different and the antisymmetric relationship between the head verb and the modifier noun is relevant. The vectorial composition between w_{buy} and w_{car} , as proposed in Eq. 2 [21], even if mathematically correct, results in a vector w_{buy_car} that does not exploit this syntactic constraint and may fail to express the underlying specific semantics.

Notice also that the components of w_{buy} and w_{car} express all their contexts, i.e. interpretations, and thus senses, of *buy* and *car* in the corpus. Some mathematical operations, e.g. the tensor product between these vectors, are thus open to misleading contributions, brought by not-null feature scores of buy_i vs. car_j ($i \neq j$) that may correspond to senses of *buy* and *car* that are not related to the specific phrase "*buy a car*".

On the contrary, in a composition, such as the verb-object pair (*buy, car*), the word *car* influences the interpretation of the verb *buy* and viceversa. The model here proposed is based on the assumption that this influence can be expressed via the operation of projection into a subspace, i.e. a subset of original features f_i . A projection is a mapping (a selection function) over the set of all features. A subspace local to the (*buy, car*) phrase can be found such that only the features specific to its meaning are selected. It seems a necessary condition that any correct interpretation of the phrase has to be retrieved and represented on the subspace of the properties shared by the proper sense of individual co-occurring words. In order to separate these word senses and neglect irrelevant ones, a projection function Π must identify these common semantic features. The resulting subspace has to preserve the compositional semantics of the phrase and it is called **support subspace** of the underlying word pair.

Consider the bigram composed by the words *buy* and *car* and their vectorial representation in a co-occurrence N -dimensional Word Space. Notice that different vectors are usually derived for different POS tags, so that the verbal and nominal use of *buy* are expressed by two different vectors, i.e. *buy.V* and *buy.N*. Every component of the vectors in a word space expresses the co-occurrence strength (in terms of frequency or *pmi*) of *buy.V* with respect to one feature, i.e. a co-occurring POS tagged word such as *cost.N*, *pay.V* or *cheaply.Adv*. The support space selects the most important features for both words, e.g. *buy.V* and *car.N*. Notice that this captures the conjunctive nature of the scalar product to which contributions come from feature with non zero scores in both vectors. Moreover, the feature score is a weight, i.e. a function of the relevance of a feature for the represented word.

As an example, let us consider the phrase "... *buy time* ...". Although the verb *buy* is the same of "... *buy a car* ...", its meaning (i.e. to do something in

order to achieve more time) is clearly different. Since vector w_{buy} expresses at least both possible meanings of the verb *buy*, different subspaces must be evoked in a distributional model for *buy car* vs. *buy time*.

Ranking features from the most important to the least important for a given phrase (i.e. pair u and v) can be done by sorting in decreasing order the components $p_i = u_i \cdot v_i$, i.e. the addends in the scalar product. This leads to the following useful:

Definition (k -dimensional support space). A k -dimensional support subspace for a word pair (u, v) (with $k \ll N$) is the subspace spanned by the subset of $n \leq k$ indexes $\mathbf{I}^k(u, v) = \{i_1, \dots, i_n\}$ for which $\sum_{t=1}^n u_{i_t} \cdot v_{i_t}$ is maximal. We will hereafter denote the set of indexes characterizing the support subspace of order k as $\mathbf{I}^k(u, v)$.

Table I reports the $k = 10$ features with the highest contributions of the point wise product of the pairs (buy, car) and $(buy, time)$. It is clear that the two pairs give rise to different support subspaces: the main components related with *buy car* refer mostly to the automobile commerce area unlike the ones related with *buy time* mostly referring to the time wasting or saving.

Similarity judgments about a pair can be thus computed within its support subspace. Given two pairs the similarity between syntactic equivalent words (e.g. nouns with nouns, verbs with verbs) is measured in the support subspace derived by applying a specific projection function. In the above example, the meaning representation of *buy* and *car* is obtained by projecting both vectors in their own subspace in order to capture the (possibly multiple) senses supported by the pair. Then, compositional similarity between *buy car* and the latter pairs (e.g. *buy time*) is estimated by (1) immersing w_{buy} and w_{time} in the selected "... *buy car* ..." support subspace and (2) estimating similarity between corresponding arguments of the pairs locally in that subspace. As exemplified in Table II, two pairs give rise to two different support spaces, so that there are two ways of projecting the two pairs. In order to provide precise definitions for these notions, formal definitions will be hereafter provided through linear algebra operators.

Space projections and compositionality. Support spaces (of dimension k) are isomorphic to projections in the original space. A projection $\Pi^k(u, v)$ can be used and provides a computationally simple model for expressing the intrinsic meaning of any underlying phrase (u, v) . Given a pair (u, v) , a unique matrix $\mathbf{M}_{uv}^k = (m_{uv}^k)_{ij}$ is defined for a given projection $\Pi^k(u, v)$ into the k -dimensional support space of any pair (u, v) according to the following definition:

Table 1. Features in the $k=10$ dimensional support space of *buy car* and *buy time*

Buy-Car	Buy-Time
<i>cheap::Adj</i>	<i>consume::V</i>
<i>insurance::N</i>	<i>enough::Adj</i>
<i>rent::V</i>	<i>waste::V</i>
<i>lease::V</i>	<i>save::In</i>
<i>dealer::N</i>	<i>permit::N</i>
<i>motorcycle::N</i>	<i>stressful::Adj</i>
<i>hire::V</i>	<i>spare::Adj</i>
<i>auto::N</i>	<i>save::V</i>
<i>california::Adj</i>	<i>warn::N</i>
<i>tesco::N</i>	<i>expensive::Adj</i>

$$(m_{uv}^k)_{ij} = \begin{cases} 1 & \text{iff } i = j \in \mathbf{I}^k(\mathbf{u}, \mathbf{v}) \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The vector $\tilde{\mathbf{u}}$ projected in the support subspace can be thus estimated through the following matrix operation:

$$\tilde{\mathbf{u}} = \Pi^k(\mathbf{u}, \mathbf{v}) \quad \tilde{\mathbf{u}} = \mathbf{M}_{uv}^k \mathbf{u} \quad (4)$$

A special case of the projection matrix is given when no k limitation is imposed to the dimension and all the positive addends in the scalar product are taken. This maximal support subspace, denoted by removing the superscript k , i.e. as $\mathbf{M}_{uv} = (m_{uv})_{ij}$, is defined as follows:

$$(m_{uv})_{ij} = \begin{cases} 0 & \text{iff } i \neq j \text{ or } u_i \cdot v_i \leq 0, \\ 1 & \text{otherwise.} \end{cases} \quad (5)$$

From Eq. 5 it follows that the support subspace components are those with positive product.

Definition. (*Left and Right Projections*). Two phrases (u, v) and (u', v') give rise to two different projections, defined as follows

$$(\text{Left projection}) \Pi_1^k = \Pi^k(\mathbf{u}, \mathbf{v}) \quad (\text{Right projection}) \Pi_2^k = \Pi^k(\mathbf{u}', \mathbf{v}') \quad (6)$$

We will denote the two projection matrices as \mathbf{M}_1^k and \mathbf{M}_2^k , correspondingly. In order to achieve a unique symmetric projection Π_{12}^k , it is possible to define the corresponding combined matrix \mathbf{M}_{12}^k as follows:

$$\mathbf{M}_{12}^k = (\mathbf{M}_1^k + \mathbf{M}_2^k) - (\mathbf{M}_1^k \mathbf{M}_2^k) \quad (7)$$

where the mutual components that satisfy Eq. 3 (or Eq 5) are employed as \mathbf{M}_{12}^k (or \mathbf{M}_{12} respectively).

Compositional Similarity Judgments. The projection function that locates the support subspace of a word pair (v, o) , whose syntactic type is *verb-object*, i.e. $\mathbf{V0}$, will be hereafter denoted by $\Pi_{vo}(\mathbf{v}, \mathbf{o})$. Given two word pairs $p_1 = (v, o)$ and $p_2 = (v', o')$, we define here a compositional similarity function $\Phi(p_1, p_2)$ as a model of the similarity between the underlying phrases. As the support subspace for the pair p_1 is defined by the projection Π_1 , it is possible to immerse the latter pair p_2 by applying Eq. 4. **This results in the two vectors $\mathbf{M}_1 \mathbf{v}'$ and the $\mathbf{M}_1 \mathbf{o}'$.** It follows that a compositional similarity judgment between two verbal phrase over the left support subspace can be expressed as:

$$\Phi_{p_1}^{(\circ)}(p_1, p_2) = \Phi_1^{(\circ)}(p_1, p_2) = \frac{\langle \mathbf{M}_1^k \mathbf{v}, \mathbf{M}_1^k \mathbf{v}' \rangle}{\|\mathbf{M}_1^k \mathbf{v}\| \|\mathbf{M}_1^k \mathbf{v}'\|} \circ \frac{\langle \mathbf{M}_1^k \mathbf{o}, \mathbf{M}_1^k \mathbf{o}' \rangle}{\|\mathbf{M}_1^k \mathbf{o}\| \|\mathbf{M}_1^k \mathbf{o}'\|} \quad (8)$$

where first cosine similarity between syntactically correlated vectors in the selected support subspaces are computed and then a composition function \circ , such

as the sum or the product, is applied. Notice how the compositional function over the right support subspace evoked by the pair p_2 can be correspondingly denoted by $\Phi_2^{(\circ)}(p_1, p_2)$. A symmetric composition function can thus be obtained as a combination of $\Phi_1^{(\circ)}(p_1, p_2)$ and $\Phi_2^{(\circ)}(p_1, p_2)$ as:

$$\Phi_{12}^{(\circ)}(p_1, p_2) = \Phi_1^{(\circ)}(p_1, p_2) \diamond \Phi_2^{(\circ)}(p_1, p_2) \quad (9)$$

where the composition function \diamond (again the sum or the product) between the similarities over the left and right support subspaces is applied. Notice how the left and right composition operators (\circ) may differ from the overall composition operator \diamond , as we will see in experiments. The above definitions in fact characterize several projection functions Π^k , local composition function $\Phi_1^{(\circ)}$ as well as global composition function $\Phi_{12}^{(\circ)}$. It is thus possible to define variants of the models presented above according to four main parameters:

Support Selection. Two different projection functions Π have been defined in Eq. 3 and Eq. 5, respectively. The MAXIMAL SUPPORT Π denotes the support space defined in Eq. 5. The k -DIMENSIONAL SUPPORT defined in Eq. 3 is always denoted by the superscript k in Π^k instead.

Symmetry of the similarity judgment. A SYMMETRIC judgment (denoted by simple Φ_{12}) involves Eq. 9 in which compositionality depends on both left and right support subspaces. In an ASYMMETRIC projection the support subspace belonging to a single (left Φ_1 , or right Φ_2) pair is chosen. In all the experiments we applied Eq. 8, by only considering the left support subspace, i.e. Φ_1 .

Symmetry of the support subspace. A support subspace can be build as:

- an INDEPENDENT SPACE, where different, i.e. left and right, support subspaces are built, through different independent projections \mathbf{M}_1 and \mathbf{M}_2
- a UNIFIED SPACE, where a common subspace is built according to Eq. 7, and denoted by the projection matrix \mathbf{M}_{12}

Composition function. The composition function Φ° in Eq. 8 and 9 can be the product or the sum as well. We will denote Φ_i^+ or Φ_i^\cdot as well as Φ^+ and Φ^\cdot to emphasize the use of sum or product in Eq. 8 and 9. The only case in which no combination is needed is when the unified support space (as in Eq. 7) is used, and thus no left or right Π_i is applied, but just Π_{12} .

4 Experimental Evaluation

The aim of this evaluation is to estimate if the proposed class of projection based methods for distributional compositional semantics is effective in capturing similarity judgments over phrases and syntactic structures. We tested our method over binary phrase structures represented by verb-object, noun-noun and adjective-noun. Evaluation is carried out over the dataset proposed by [21], which is part of the *GEMS 2011 Shared Evaluation*. It consists of a list of 324

pairs rated with scores ranging from 1 to 7. For each of them 18 scores derived by different human annotators are provided, for a global of 5832 scores. The type of pairs is designed to investigate 3 kinds of syntactic relations: adjective-noun (AdjN), verb-object (VO) or noun-noun (NN). For every syntactic type, thus, there are 108 different pairs, each annotated with 18 scores.

In Table 2, examples of pairs and scores are shown: notice how the similarity between the (VO) *offer support* and *provide help* is higher than the one between *achieve end* and *close eye*. The correlation of the similarity judgements output by a DCS model against the human judgements is computed using Spearman’s ρ , a non-parametric measure of statistical dependence between two variables proposed by [8].

Table 2. Example of Mitchell and Lapata dataset for the three syntactic relations verb-object (VO), adjective-noun (AdjN) and noun-noun (NN)

Type	First Pair	Second Pair	Rate
VO	<i>support offer</i>	<i>provide help</i>	7
	<i>use knowledge</i>	<i>exercise influence</i>	5
	<i>achieve end</i>	<i>close eye</i>	1
AdjN	<i>old person</i>	<i>right hand</i>	1
	<i>vast amount</i>	<i>large quantity</i>	7
	<i>economic problem</i>	<i>practical difficulty</i>	3
NN	<i>tax charge</i>	<i>interest rate</i>	7
	<i>tax credit</i>	<i>wage increase</i>	5
	<i>bedroom window</i>	<i>education officer</i>	1

sentence-based space, is derived by applying SVD to a $M = \text{term} \times \text{sentence}$ adjacency matrix. Each column of M represents thus a sentence of the corpus, with about 1,500,000 sentences and *tf-idf* scores for words w in each row. The dimensions of the resulting SVD matrix in the sentence-based space is $N = 250$.

The second space employed is a *word space* built from the ukWak co-occurrences where left contexts are treated differently from the right ones for each target word tw . Each column in M represents here a word w in the corpus and in rows we found the *pmi* values for the individual features f_i , as captured in a window of size ± 3 around w . The most frequent 20,000 left and right features f_i are selected, so that M expresses 40,000 contexts. SVD is here applied to limit dimensionality to $N = 100$.

Comparative analysis with results previously published in [21] has been carried out. We also recomputed the performance measures of operators in [21] (e.g. M&L multiplicative or additive models of Eq. 1 and 2) over all the word spaces specifically employed in the rest of our experiments.

Table 3 reports M&L performances in first three rows. In the last row of the Table the max and the average interannotator agreement scores for the three

We employed two different word spaces derived from a corpus, i.e. ukWak [22], including about 2 billion tokens. Each space construction proceeds from an adjacency matrix M on which Singular Values decomposition [12] is then applied. Part-of-speech tagged words have been collected from the corpus to reduce data sparseness. Then all target words *tw*s occurring more than 200 times are selected, i.e. more than 50,000 candidate features. A first space, called *sentence-*

categories derived through a leave one-out resampling method, are shown. For each category with a set of subjects responses of size m , a set of $m - 1$ (i.e., the response data of all but one subject) and a set of size one (i.e., the response data of the single remaining subject) are derived. The average rating of the set of $m - 1$ subjects is first calculated and then Spearman's ρ correlation coefficient with respect to the singleton set is computed. Repeating this process m times results in an average and maximum score among the results (as reported in row 6). The distributional compositional models discussed in this paper are shown in rows 4 and 5, where different configurations are used according to the models described in Section 3. For example, the system denoted in Table 3 as $\Phi_{12}^{(+)}$, $\Phi_i^{(+)}$, Π_i^k ($k=40$), corresponds to an additive symmetric composition function $\Phi_{12}^{(+)}$ (as for Eq. 9) based on left and right additive compositions $\Phi_i^{(+)}$ ($i = 1, 2$ as in Eq. 8), derived through a projection Π_i^k in the support space limited to the first $k = 40$ components for each pair (as for Eq. 6).

First, Mitchell and Lapata operators applied onto our sentence and word space models over perform results previously presented in [21] (i.e. row 2 and 3 vs. row 1). Moreover, the Additive operator seems to significantly outperform the Multiplicative one, against the evidence discussed in [21]. This is mainly due to both the benefits of the adopted SVD modeling and to the different kind of spaces used here, whereas left and right contexts are taken separate into different lexical features. The use of *pmi* scores in word spaces or *tf-idf* values in sentence spaces, then subject to the SVD factorization, seems thus beneficial to the multiplicative and additive M&L models.

The best performances are achieved by the projection based operators proposed in this paper. The word space version (denoted by $\Phi^{(+)}$, Π_{12}^k ($k=30$)) gets the best performance over two out of three syntactic patterns (i.e. AdjN and NN) and is close to the best figures for VO. Notice how parameters of the projection operations influence the performance, so that different settings provide quite different results. This is in agreement with the expected property for which different syntactic compositions require different vector operations.

If compared to the sentence space, a word space, based on a small window size, seems better capture the lexical meaning useful for modeling the syntactic composition of a pair. The subset of features, as derived through SVD, in a resulting support space is very effective as it is in good agreement with human judgements ($\rho=0.71$) A sentence space leads in general to a more topically-oriented lexical representations and this seems slightly less effective. In synthesis it seems that specific support subspaces are needed: a unified additive model based on a Word Space is better for adjective-noun and compound nouns while the additive symmetric model based on a sentence space is much better for verb-object pairs.

A general property is that the results of our models are close to the average agreement among human subjects, this latter representing a sort of upper bound for the underlying task. It seems that latent topics (as extracted through SVD from sentence and word spaces) as well the projections operators defined by support subspaces provide a suitable comprehensive paradigm for compositionality.

Table 3. Spearman’s ρ correlation coefficients across Mitchell and Lapata models and the projection-based models proposed in Section 3. Topical Space and Word space refer to the source spaces. is used as input to the LSA decomposition model.

Model		AdjN	NN	VO
Mitchell&Lapata, [21]	Additive	.36	.39	.30
	Multiplicative	.46	.49	.37
	Dilation	.44	.41	.38
Mitchell&Lapata Topical SVD	Additive	.53	.67	.63
	Multiplicative	.29	.35	.40
	Dilation	.44	.49	.50
Mitchell&Lapata Word Space SVD	Additive	.69	.70	.64
	Multiplicative	.38	.43	.42
	Dilation	.60	.57	.61
Sentence Space	$\Phi_1^{(+)}, \Pi_1^k (k=20)$.58	.62	.64
	$\Phi_{12}^{(+)}, \Phi_i^{(+)}, \Pi_i^k (k=40)$.55	.71	.65
	$\Phi_{12}^{(+)}, \Phi_i^{(+)}, \Pi_i^k (k=10)$.49	.65	.66
Word Space	$\Phi^{(+)}, \Pi_{12}^k (k=30)$.70	.71	.63
	$\Phi_{12}^{(\cdot)}, \Phi_i^{(+)}, \Pi_i^k (k=40)$.68	.68	.64
	$\Phi_{12}^{(\cdot)}, \Phi_i^{(\cdot)}, \Pi_i$.70	.65	.61
Agreement among Human Subjects	Max	.88	.92	.88
	Avg	.72	.72	.71

They seem to capture compositional similarity judgements that are significantly close to human ones.

5 Conclusions

In this paper, a distributional compositional semantic model based on space projection guided by syntagmatically related lexical pairs is defined. Syntactic bi-grams are here projected in the so called *Support Subspace* and compositional similarity scores are correspondingly derived. This represents a novel perspective on compositional models over vector representations with respect to shallow vector operators (e.g. additive, or multiplicative, tensorial algebraic operations) as proposed elsewhere, e.g. in [21]. The approach presented here focuses on first selecting the most important components for a specific word pair in a relation and then modeling their similarity. This captures their meanings locally relevant to the specific context evoked by the pair. The proposed *projection-based* method of DCS, evaluated over a well known dataset [21], is very effective for the syntactic structures of VO, NN and AdjN. It achieves the same results than the average human inter-annotator agreement, by outperforming most previous results (e.g. [21]). Given the small size of the adopted dataset (only 108 pairs per syntactic type) the statistical significance of some of the reported results could not be fully assessed and further evaluation is needed against larger data sets. Future work will also investigate other compositional prediction tasks (e.g. selectional preference modeling or the ranking of short texts) as well as cross-linguistic

experiments. Finally, the specific task adopted in the tests assumes compositionality to hold for every input pair, although some complex expressions, such as the idiomatic ones, do not satisfy this need. However the applicability of the proposed methods to the detection of idiomatic structures is still possible and will be also part of future investigation.

References

1. Montague, R.: *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press (1974)
2. Coecke, B., Sadrzaden, M., Clark, S.: Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschrift, Linguistic Analysis* 36 (2010)
3. Firth, J.: A synopsis of linguistic theory 1930-1955. In: *Studies in Linguistic Analysis*. Philological Society, Oxford (1957); reprinted in Palmer, F. (ed.) *Selected Papers of J. R. Firth*. Longman, Harlow (1968)
4. Schütze, H.: Automatic Word Sense Discrimination. *Computational Linguistics* 24, 97–124 (1998)
5. Wittgenstein, L.: *Philosophical Investigations*. Blackwells, Oxford (1953)
6. Schütze, H.: Word space. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) *NIPS* 5, pp. 895–902. Morgan Kaufmann Publishers, San Mateo (1993)
7. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37, 141 (2010)
8. Mitchell, J., Lapata, M.: Vector-based models of semantic composition. In: *Proceedings of ACL/HLT 2008*, pp. 236–244 (2008)
9. Baroni, M., Zamparelli, R.: Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In: *EMNLP 2010*, pp. 1183–1193. Association for Computational Linguistics, Stroudsburg (2010)
10. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. *CoRR* abs/1106.4058 (2011)
11. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. *Communications of the ACM* 18, 613–620 (1975)
12. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *Journal of The American Society For Information Science* 41, 391–407 (1990)
13. Landauer, T.K., Dumais, S.T.: A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 211–240 (1997)
14. Harris, Z.S.: *Mathematical Structures of Language*. Wiley, NY (1968)
15. Lin, D.: Automatic retrieval and clustering of similar word. In: *Proceedings of COLING-ACL, Montreal, Canada* (1998)
16. Pantel, P., Lin, D.: Document clustering with committees. In: *Proceedings of SIGIR 2002, Montreal, Canada*, pp. 199–206 (2002)
17. Pennacchiotti, M., Cao, D.D., Basili, R., Croce, D., Roth, M.: Automatic induction of framenet lexical units. In: *EMNLP*, pp. 457–465 (2008)
18. Croce, D., Giannone, C., Annesi, P., Basili, R.: Towards open-domain semantic role labeling. In: *Proceedings of ACL*, pp. 237–246 (2010)
19. Foltz, P.W., Kintsch, W., Landauer, T.K.: The measurement of textual coherence with latent semantic analysis. *Discourse Processes* 25, 285–307 (1998)

20. Erk, K., Pad, S.: A structured vector space model for word meaning in context. In: EMNLP 2008: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 897–906. ACL (2008)
21. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. *Cognitive Science* 34, 1388–1429 (2010)
22. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources And Evaluation* 43, 209–226 (2009)

Distributional Models and Lexical Semantics in Convolution Kernels

Daniilo Croce, Simone Filice, and Roberto Basili

Department of Enterprise Engineering
University of Roma, Tor Vergata
Via del Politecnico 1, 00133 Roma
{croce, filice, basili}@info.uniroma2.it

Abstract. The representation of word meaning in texts is a central problem in Computational Linguistics. Geometrical models represent lexical semantic information in terms of the basic co-occurrences that words establish each other in large-scale text collections. As recent works already address, the definition of methods able to express the meaning of phrases or sentences as operations on lexical representations is a complex problem, and a still largely open issue. In this paper, a perspective centered on Convolution Kernels is discussed and the formulation of a Partial Tree Kernel that integrates syntactic information and lexical generalization is studied. The interaction of such information and the role of different geometrical models is investigated on the question classification task where the state-of-the-art result is achieved.

1 Introduction

Language learning systems usually generalize linguistic observations into rules and patterns that are statistical models of higher level semantic inferences. Statistical learning methods make the assumption that lexical or grammatical observations are useful hints for modeling different semantic inferences, such as in document topical classification, predicate and role recognition in sentences as well as question classification in Question Answering. Features are then generalized into predictive components in the final model that is effectively induced from the training examples. When the availability of training data is scarce, lexical information (such as lemmas, multiword expressions or Named Entities) can be limited by data sparseness effects and generalization is thus needed. Suitable representations of word meaning as derived from texts play a crucial role here, being a core problem in Computational Linguistics.

Geometrical models represent lexical semantic information through the analysis of observations across large-scale corpora. The core idea is that the meaning of a word can be described by the set of textual contexts in which it appears (*Distributional Hypothesis* as described in [1]). Words can be represented as vectors whose components reflect the corresponding contexts: two words close in the space (i.e. they have similar contexts) are likely to be related by some type of generic semantic relation, either paradigmatic (e.g. synonymy, hyperonymy, antonymy) or syntagmatic (e.g. meronymy, conceptual and phrasal association), as observed in [2]. Semantic spaces have been widely used for representing the meaning of words or other lexical entities ([3]), with

successful applications in lexical disambiguation ([4]), harvesting thesauri [5], Name Entity Classification [6] or the Semantic Role Labeling task, as in [7].

Obviously, lexical information usually implies different words to provide different contributions but usually neglects other crucial linguistic properties, such as word ordering. In some approaches, symbolic expressions, i.e. pseudo-words, are extracted from the syntactic parse trees and used as lexical features. However, the overall limitation of geometrical models is their non-compositional nature. In general, they ignore the grammatical structure of sentences but also. On the other side, even in more structured versions, they do not integrate any syntax in computing the meanings of phrases, as that they implicitly do for words, e.g. [8]. As recent works already address, e.g. [9], the definition of methods able to express the meaning of phrases, or sentences, through composition operations acting over the underlying lexical representations is a complex problem, and a still largely open issue. Some studies, e.g. [10] [11] [12], propose classes of algebraic operators (e.g. tensor products) to effectively combine the lexical information of constituents. Their focus is to explicitly combine vectors representing words of a phrase in order to obtain a new vector that represents the semantics of entire phrase.

In this work we follow a different approach inspired by *Convolution Kernel* methods introduced in [13]. The idea is that we do not need to compose the geometric representation of words to estimate the similarity among two sentences, but instead we compute it in an implicit space by exploiting their grammatical structure. Such kernel methods are very useful as they can be applied to many well-known learning algorithms, such as Perceptrons or Support Vector Machines (SMVs). A key property of these algorithms is that the only operation they require is the evaluation of dot products between pairs of examples. The dot product can be replaced with a Mercer kernel, implicitly mapping feature vectors into a much larger feature space where the original algorithm can be applied and the most representative feature can be automatically selected. Automatic feature engineering of syntactic or shallow semantic structures has been carried out by means of Syntactic Tree Kernels (STK), e.g. [14].

One main limitation of these approaches is that they apply a hard matching between node labels: two words, e.g. *boy* and *child*, are different and will provide no contribution to the overall similarity estimation, although they support the same inductive inferences in a learning process. A more effective similarity estimation between tree structures should consider lexical generalization and apply a more expressive strategy than string matching between labels. Most notably, the work in [15] encodes lexical similarity in tree kernels. This is essentially the STK in which syntactic fragments from constituency trees can be matched, even if they only differ in the leaf nodes (i.e. they have different surface forms). This implies matching scores lower than 1, depending on the semantic similarity of the corresponding leaves in the syntactic fragments. In [16] a more general formulation of the a semantically *Smoothed Partial Tree Kernel* (SPTK) has been provided. With respect to [15] it can be applied to every tree node (not only the leaves) and it has been successfully applied to dependency parse trees.

One open issue is that different kinds of generalizations can be obtained by changing the adopted lexical similarity function, as this generalizes different semantic aspects of the involved words. While similarity can be modeled directly over lexical resources, e.g. WordNet as discussed in [17], their development can be very expensive thus limiting the

coverage of the resulting convolution kernel, especially in specific application domains. Moreover in [16] the impact of a specific lexical resource has been compared with the one achievable with lexical information gathered through the distributional analysis of a large scale corpus. Experimental findings show that a distributional approach provides better results. This is very interesting as distributional approaches are unsupervised and largely applicable directly from the application domain texts. However a proper investigation of the impact of different possible geometrical representations is still needed. By employing different notions of *context*, we can assume two words similar when they appear *in the same documents* [18,19] or *in the same sentences* (modeled as word co-occurrences in short windows [2]) or even *in the same syntactic structures* [8].

In this work, we investigate how the choice of the above different semantic representations impact on the generalization capability of the SPTK in a fined-grained semantic task, i.e. the Question Classification task. In Section 2 different Convolution Kernels among linguistic structures will be discussed. Section 3 evaluates the impact of different semantic representations. Section 4 derives the conclusions.

2 Kernel-Based Learning and Distributional Information

In kernel-based machines, both learning and classification algorithms only depend on the inner product between instances. If an example is not represented by a vector, the product can be implicitly computed by kernel functions by exploiting the following dual formulation: $\sum_{i=1..l} y_i \alpha_i \phi(o_i) \phi(o) + b = 0$, where o_i and o are two objects, ϕ is a mapping from the objects to feature vectors x_i and $\phi(o_i) \phi(o) = K(o_i, o)$ is a kernel function that implicitly captures the similarity. In this section different kernels among tree structures, i.e. Tree Kernels, will be discussed. Then geometrical models of lexical semantics will be presented according to a kernel perspective. Finally the formulation of a kernel, able to combine syntactic and lexical information, i.e. the Smoothing Partial Tree Kernel will be discussed.

2.1 Convolution Tree Kernels

Convolution Tree Kernels (TK) compute the number of common substructures between two trees T_1 and T_2 without explicitly considering the whole fragment space. For this purpose, let the set $\mathcal{F} = \{f_1, f_2, \dots, f_{|\mathcal{F}|}\}$ be a tree fragment space and $\chi_i(n)$ be an indicator function, equal to 1 if the target f_i is rooted at node n and equal to 0 otherwise. A tree-kernel function over T_1 and T_2 is $TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$, N_{T_1} and N_{T_2} are the sets of the T_1 's and T_2 's nodes, respectively and

$$\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \chi_i(n_1) \chi_i(n_2)$$

The latter is equal to the number of common fragments rooted in the n_1 and n_2 nodes. The Δ function determines the richness of the kernel space.

A largely known kernel, i.e. Syntactic Tree Kernel (STK), has been introduced in [20] to define a similarity between two sentences by exploiting their syntactic structures,

i.e. the parse trees. It is sufficient to compute $\Delta_{STK}(n_1, n_2)$ as follows (recalling that since it is a syntactic tree kernels, each node can be associated with a production rule): (i) if the productions at n_1 and n_2 are different then $\Delta_{STK}(n_1, n_2) = 0$; (ii) if the productions at n_1 and n_2 are the same, and n_1 and n_2 have only leaf children then $\Delta_{STK}(n_1, n_2) = \lambda$; and (iii) if the productions at n_1 and n_2 are the same, and n_1 and n_2 are not pre-terminals then $\Delta_{STK}(n_1, n_2) = \lambda \prod_{j=1}^{l(n_1)} (1 + \Delta_{STK}(c_{n_1}^j, c_{n_2}^j))$, where $l(n_1)$ is the number of children of n_1 and c_n^j is the j -th child of the node n .

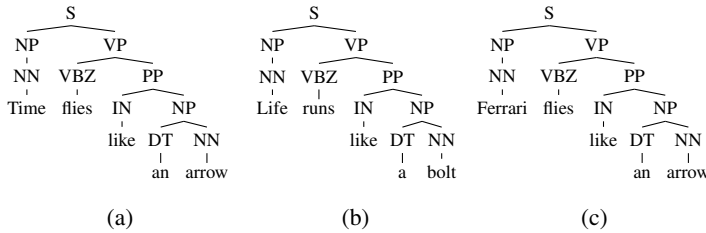


Fig. 1. Examples of syntactic parse trees

It is a very powerful method as it counts the common subtree structures shared by the sentences in a implicit space where each component corresponds to one possible tree fragment. Figure 1 shows the parse trees of the sentences *Time flies like an arrow*, *Life runs like a bolt* and *Ferrari flies like an arrow*, respectively. Common subtrees that would contribute to a kernel would be $(S (VP) (NP))$ or $(NP (DT) (NN))$. The main advantage is that it is not necessary to define explicitly all the possible tree configurations, as only components useful to estimate the similarity will be taken into account. In this implicit space the resulting vector can be seen as the composition of all the atomic information (i.e. the tree fragments) needed to reflect the syntactic structure of the sentence, as well the lexical information of the tree leaves (i.e. the words). STK are rigid measures of semantic similarity as for their strict requirements on the matching of syntactic substructures. The tree kernel discussed in [20] only triggers matches that fully satisfy derivation rules in the underlying grammars: this implies that only identical words appearing in the corresponding syntactic position are matched.

Partial Tree kernels (PTK, [21]) are an attempt to relax these grammatical constraints, but they only act at the syntagmatic level. If a partial match between two syntactic structures is applied, the corresponding skipped material is fully neglected. It does not provide any contribution to the kernel, i.e. no lexical contribution can be observed. The computation of PTK is carried out by the following Δ_{PTK} function: if the labels of n_1 and n_2 are different then $\Delta_{PTK}(n_1, n_2) = 0$; else $\Delta_{PTK}(n_1, n_2) =$

$$\mu \left(\lambda^2 + \sum_{I_1, I_2, l(I_1)=l(I_2)} \lambda^{d(I_1)+d(I_2)} \prod_{j=1}^{l(I_1)} \Delta_{PTK}(c_{n_1}(I_{1j}), c_{n_2}(I_{2j})) \right) \tag{1}$$

where $d(I_1) = I_{1l(I_1)} - I_{11}$ and $d(I_2) = I_{2l(I_2)} - I_{21}$. This way, we penalize both larger trees and child subsequences with gaps. PTK is more general than the STK as if we only consider the contribution of shared subsequences containing all children of nodes, we implement the STK kernel.

2.2 Lexical Semantic Kernel

One main limitation of TKs is their poor ability in generalizing the lexical information carried out by words, i.e. the tree leaves. For example sentences represented in Figure (1a) and (1b) are clearly related, as they both evoke the notion of time that passes quickly. Sentence in Figure (1c) instead refers to the Ferrari as a fast car. According to a STK, the first sentence shares with the last one all the syntactic parse trees and 4 (of 5) lexical items, obtaining the highest similarity. A smoothed approach, able to infer that *runs* and *flies* (as well as *bolt* and *arrow*) can contribute positively to the kernel estimation, is needed. This is important for NLP task especially in scenarios where the number of examples is reduced. According to the Distributional Hypothesis, we define a space in which dimensions represent different contexts, so that words sharing similar dimensions have similar meanings as they occur in the same contexts. Obviously, different context types define geometric spaces with different semantic properties and different generalization grain in the resulting kernel estimation. For example, a wider context will provide a shallower generalization while a smaller one will capture more specific lexical aspects of words, as well as their syntactic behavior. In this work, three different kinds of context are investigated.

Topical space: a document-based geometric space represents words by focusing on coarse grain textual elements, such as documents [18]. Two words will have a similar geometric representation if they appear in the same documents of a corpus. In Information Retrieval it is usually employed to represent texts via linear combinations of (usually orthonormal) vectors corresponding to their component words. This captures contextual information by expressing the distribution of words across text collections. Dually, these models map words in vector spaces whose dimension is equal to the number of documents in the corpus and the individual score is computed according the term frequency-inverse document frequency (tf-idf) schema, [18].

Word-based space: In such space, contexts are words, as lemmas, appearing in a n -window of the target words [2]. Co-occurrence counts are collected in a words-by-words matrix, where the elements record the number of times two words co-occur within a set window of word tokens. To provide a robust weighting schema and penalize common words, whose high frequency could imply an unbalanced representation, Pointwise Mutual Information (PMI) [3] scores are commonly adopted. The window width n is a parameter that allows the space to capture different lexical properties: larger values for n introduces more words, i.e. entire topics including possibly noisy information. Lower values lead to sparse representations that seem more oriented to capture paradigmatic lexical classes.

Syntax-based space: In order to make the representation sensitive to the grammatical information, in the construction of the space, contexts words can be enriched by features that directly express syntactic information, as discussed in [8]. The words-by-words matrix here records the number of times a target word w (i.e. the rows) co-occurs with another word in a specific syntactic relation r . Columns are thus corresponding to word-relation pairs, so that each space dimension reflects the pair $\langle r, w \rangle$. For example, given the verb *fly.v*, it records the number of times *fly.v* is grammatically connected with $\langle \text{SUBJ}, \text{ferrari.n} \rangle$ or $\langle \text{SUBJ}, \text{time.n} \rangle$, but also $\langle \text{like_PMOD}, \text{arrow.n} \rangle$ or

(`like_PMOD`, `bolt.n`). Vector components here provide a more precise representation as a lot of (possibly noisy) material is filtered out, although resulting in a very sparse representation. The individual score is also computed according to PMI.

In order to have a more robust representation, the original word-by-context matrix M is decomposed through Singular Value Decomposition (SVD) [19][22] into the product of three new matrices: U , S , and V so that S is diagonal and $M = USV^T$. M is approximated by $M_k = U_k S_k V_k^T$ in which only the first k columns of U and V are used, and only the first k greatest singular values are considered. This approximation supplies a way to project a generic term w_i into the k -dimensional space using $W = U_k S_k^{1/2}$, where each row corresponds to the representation vectors w_i . The original statistical information about M is captured by the new k -dimensional space which preserves the global structure while removing low-variance dimensions, i.e. distribution noise. These newly derived features may be thought of as latent concepts, each one representing an emerging meaning component as a linear combination of many different contexts. It is worth noticing here that the application of SVD to different spaces results in very different latent topics. The emerging of special directions in the space as caused by different linguistic contexts (e.g. from documents to short windows around words) provides significantly different linguistic implications. When larger contexts are used, the resulting latent topics act as primitive concepts that characterize the document topics, i.e. domain knowledge characterizing the corpus. When shorter contexts are adopted in M , latent topics characterize very simple patterns such as noun-verb, adjective-noun or verb-noun. These are primitive features needed to distinguish paradigmatic lexical classes, for which syntactic substitutability holds¹.

Given two words w_1 and w_2 , the term similarity function σ is estimated as the cosine similarity between the corresponding projections w_1 , w_2 , i.e

$$\sigma(w_1, w_2) = \frac{w_1 \cdot w_2}{\|w_1\| \|w_2\|} \quad (2)$$

The similarity in Eq. 2 is known as *Latent Semantic Kernel (LSK)*, as defined in [23], and the σ function defines a Gram matrix $G = \sigma(w_1, w_2) \forall w_1, w_2$ that is positive semi-definite ([23][24]). It implies that σ is a valid kernel and it can be combined with other kernels, as discussed in the next session. There are also other advantages. First, the overall computational cost of *LSK* is smaller than the one on the original space, as for its much fewer dimensions (e.g. 10K vs. 100). Moreover, *LSK* captures second-order relations among words, thus improving the generalization capability of the similarity measure.

2.3 Smoothing Partial Tree Kernels

Combining lexical and structural kernels provides clear advantages on all-vs-all words similarity, which tends to semantically diverge. Indeed syntax provides the necessary restrictions to compute an effective semantic similarity. Following this idea, Bleedhorn & Moschitti [15] modified step (i) of Δ_{STK} computation as follows: (i) if n_1

¹ Note that SVD emphasizes directions in the space with maximal covariance for M , i.e. feature clusters maximizing the difference between individual words. This gives rise to corpus-specific word clusters.

and n_2 are pre-terminal nodes with the same number of children, $\Delta_{STK}(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} \sigma(\text{lex}(n_1), \text{lex}(n_2))$, where lex returns the node label. This allows to match fragments having same structure but different leaves by assigning a score proportional to the product of the lexical similarities of each leaf pair. Although it is an interesting kernel, the fact that lexicals must belong to exactly the same structures and on the leaf nodes limits its applications. As described in [16], a smoothed tree kernel, that can be applied to any tree, exploits a lexical semantic kernel, while respecting the syntax enforced by the tree. When a partial tree kernel is employed, i.e. Eq. 1 its smoothed counterpart is defined as follows: if n_1 and n_2 are leaves then $\Delta_\sigma(n_1, n_2) = \mu\lambda\sigma(n_1, n_2)$; else

$$\Delta_\sigma(n_1, n_2) = \mu\sigma(n_1, n_2) \times \left(\lambda^2 + \sum_{I_1, I_2, l(I_1)=l(I_2)} \lambda^{d(I_1)+d(I_2)} \prod_{j=1}^{l(I_1)} \Delta_\sigma(c_{n_1}(I_{1j}), c_{n_2}(I_{2j})) \right) \quad (3)$$

where σ is any lexical similarity between nodes (i.e. Eq. 2) and the other variables are the same as in PTK, Eq. 1. We call this kernel a *smoothed partial tree kernel*, i.e. *SPTK*. In this formulation, for every tree pair the σ function estimates the similarity among the nodes, so if labels are the same (i.e. $\sigma = 1$) the contribution is equal to $\mu\lambda^2$, as in the PTK; otherwise the contribution of the nodes and the subtrees is weighted according to the information provided by the word space³, whose quality is crucial. If σ tends to confuse words not semantically related or apply too much smoothing, the overall learning algorithm will not be able to well characterize useful examples. A too strict function will otherwise produce the same results of a pure PTK.

3 Experimental Evaluation

The aim of the experiments is to measure how different grammatical representations, i.e. dependency structures, and different lexical semantic representations impact on the effectiveness of the *SPTK* kernel. Accordingly, we carried out extensive experiments on Question Classification (QC), as a specific, yet complex, semantic inference.

3.1 General Experimental Setup

For these experiments, we used the UIUC QC dataset [25], made by a training set of 5,452 questions and a test set of 500 questions⁴. Question classes are organized in two levels: 6 coarse-grained classes (like ENTITY or HUMAN) and 50 fine-grained sub-classes (e.g. PLANT, FOOD as subclasses of ENTITY). While the former is more sensitive to syntax, the latter is highly dependent on lexical information. Giving the particularly limited number of training examples available for the individual fine-grained classes, the lexical generalization acquired from the external corpus through distributional analysis is crucial. We employed the SVM learning algorithm, by extending the SVM-LightTK software⁵ [21] with the *SPTK* defined by Equation 3. According to findings discussed in [16], the SPTK achieves best results when applied to structures obtained from dependency parse trees. Sentences are parsed with the LTH dependency parser described in [26].

² When n_1 and n_2 are not lexical nodes σ will be 0 when $n_1 \neq n_2$.

³ <http://cogcomp.cs.illinois.edu/Data/QA/QC/>

⁴ <http://disi.unitn.it/moschitti/Tree-Kernel.htm>

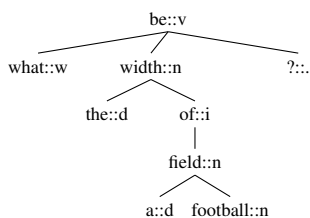


Fig. 2. Lexical Only Centered Tree (LOCT)

Figure 2 shows the Lexical Only Centered Tree (LOCT) which is directly derived by the parse tree. It only accounts on the lexicals, where untyped binary relations are used for recursive structures. The grammatical generalization provided by the syntactic edge labels is thus neglected. In the empirical perspective pursued here, this structure is interesting as the lexical generalization is applied to all tree nodes corresponding to content words. We apply lemmatization to the lexicals to limit sparseness and, at the same time, we also adopt a set of 10 simplified PoS-tags, e.g. noun (n::), verb (v::), adjective (:a). This allows to measure similarity only between lexicals in the same grammatical category. In contrast, the LCT shown in Figure 3 represents the dependency structures where both grammatical and all PoS-Tags are retained as rightmost children.

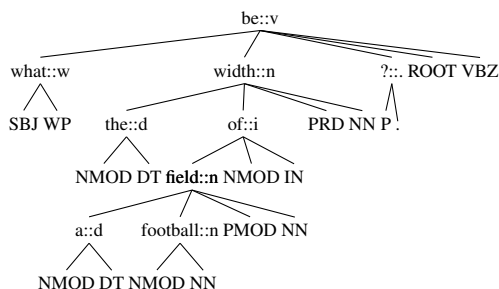


Fig. 3. Lexical Centered Tree (LCT)

The corpus employed to develop the word spaces, i.e. ukWak [27], is a large scale document collection made by 2 billion tokens. To reduce data sparseness all target words tw s that occur in the ukWak more than 200 times have been selected, i.e. more than 50,000 words. Each tw corresponds to a matrix row and it is labeled with the pair \langle lemma, ::POS \rangle . Then different approaches are applied to build the word-by-context matrix M , as described in Sections 2.2.

Topical Space: the entire corpus has been split so that each column of M represents a sentence. The number of different sentences is about 1,500,000 and each matrix item contains the $tf-idf$ score of tw with one corresponding sentence.

Word-based Space win. n: for this co-occurrence word space, left contexts are treated differently from the right ones. Each column of M represents a word in the corpus and

each item measures the number of times this word co-occurs with *tw* in a window of size $\pm n$. The most frequent 20,000 items are selected, so that M models 40k contexts (i.e. right and left contexts). Two window sizes are employed: a size $n=3$ to have a more precise representation and better capturing syntactic properties of words and $n=6$ to provide a more smoothed generalization.

Syntax-based Space: contexts here are made of syntactically-typed co-occurrences within dependency graphs built from the entire set of ukWak sentences. This is a very sparse space and the most frequent 150,000 basic features, i.e. $\langle \text{synt_rel}, \text{lemma}::\text{pos} \rangle$, are employed as contextual features corresponding to PMI scores.

The SVD reduction is then applied to M , with three different dimensionality cuts of 30, 100 and 250, where a lower dimensionality provides a larger compression but a less precise generalization. We experiment with multi-classification, which we model through *one-vs-all* scheme by selecting the category associated with the maximum SVM margin. In all the experiments the kernel estimation is normalized⁵. The quality of such classification is measured with accuracy, i.e. the percentage of test examples that are correctly classified. The parameterization of each classifier is carried on a held-out set (30% of the training) and concerns with the setting of the trade-off parameter (option `-c`). This fix split between train and test is useful to have a more meaningful comparison between the different employed spaces. Moreover, it is the same experimental setup provided in [25]. In contrast, the cost-factor parameter of the SVM-LightTK (option `-j`) is set as the ratio between the number of negative and positive examples, for attempting to get a balanced contribution of training examples.

3.2 QC Experiments: Results

In these experiments, a model that does not account on the syntactic structure of the sentence (i.e. a Bag of Word model) is employed as baseline. When only lemmatized words are considered and no SVD reduction is applied, an accuracy of 89.4% and 83.8% for the coarse-grained and fined-grained setting is estimated respectively. The outcome of the several kernels applied to several structures for the coarse and fine grained QC is reported in Table 1 and 2 respectively. The first column shows different experimented spaces employed with the SPTK. The second column refers to different dimensionality reduction via SVD employed. The last two columns report the accuracy scores obtained by applying the *SPTK* kernel to the LOCT and LCT structures of Section 3.1. The first line contains accuracy where no generalization is applied, i.e. kernel formulation is comparable with the PTK described in [21].

In the coarse grain setting, best results are obtained by the LCT structure where the improvement of 4.4% in accuracy (from 90.8% to 94.8%) confirms that the lexical generalization is very useful even for tasks like the coarse grained QC, for which the syntactic structure of the question is the most discriminative feature. This is confirmed by results achieved by the LCT that, although using explicit syntactic labels, outperform the LOCT. It is worth to notice that best results are obtained by the co-occurrence word

⁵ To have a similarity score between 0 and 1, a normalization in the kernel space, i.e. $\frac{TK(T_1, T_2)}{\sqrt{TK(T_1, T_1) \times TK(T_2, T_2)}}$ is applied.

Table 1. Accuracy of structural kernels for coarse grained QC

Space	Dimens.	LOCT	LCT
-	-	89,2%	90,8%
Topical	30	71,8%	86,8%
	100	85,8%	91,4%
	250	88,6%	92,0%
Word-based (win. 3)	30	86,6%	93,4%
	100	90,4%	94,4%
	250	93,6%	94,8%
Word-based (win. 6)	30	89,4%	92,2%
	100	92,8%	93,6%
	250	93,0%	93,8%
Syntax-based	30	86,4%	91,8%
	100	91,6%	94,0%
	250	94,2%	93,8%

space with a window size of 3, thus confirming the need of a specific generalization for lexicals. As already noticed in Section 2.2, different word spaces seem to capture different linguistic generalizations. The co-occurrence word space outperforms the Syntactic Word Space (94.8% respect to 93.8%). It suggests that, while the latter space is more precise, its overall accuracy can be reduced by parsing errors as well as by data sparseness, as every component in the space corresponds to a word typed by a syntactic relation. This finding is also confirmed in the fine grained setting, in which the impact of a co-occurrence word space is more beneficial. The fine grained setting represents a task where the lexical information is much more effective. The LOCT representation here achieves the best results (i.e. 87.4%) although the differences among word spaces are negligible. Not every space overcomes the baseline. The topical space, in both settings, is quite unstable, especially for LOCT, where no explicit syntax is encoded in the tree. This is in line with the assumption that applying SVD over document-based spaces results in domain (or topical) similarity that is a rather different notion than paradigmatic similarity. As the *SPTK* kernel requires a semantic smoothing harmonic with the syntax, paradigmatic relations are preferable, as they better comply to substitutability in interpretation. These latter relations seem better captured by co-occurrence word spaces with smaller windows, as the difference in performance between $n=3$ and $n=6$ suggests. As an example, a question whose classification is mistaken by the bag-of-word approach, as well as by the PTK (with no lexical smoothing) is Q: *What French ruler was defeated at the battle of Waterloo?*. Also the classification with a *SPTK* built over a topical space wrongly associates Q with ENTITY rather than with the correct coarse category of HUMAN. It is clearly an example of a question whose lexical information needs to be generalized to induce that a *ruler* is a man as it can be *defeated*. An analysis of the latent semantic topics obtained by SVD over the different source spaces, i.e. topical and word-based, has been carried out to find the possible generalizations obtained in the two cases. By projecting the words *ruler.n* in the topical space the 5 most similar words are *persia.n*, *persian.n*, *rebels.n*, *dominium.n*, *medes.n* while in the word space (with $n=3$) are *conqueror.n*, *emperor.n*, *dominium.n*, *dynasty.n* and

Table 2. Accuracy of structural kernels for fine grained QC

Space	Dimens.	LOCT	LCT
-	-	85,4%	85,4%
Word-based (win. 3)	30	79,2%	82,4%
	100	85,8%	85,2%
	250	87,2%	86,8%
Word-based (win. 6)	30	80,0%	80,6%
	100	85,4%	85,0%
	250	87,4%	86,6%
Syntax-based	30	71,2%	78,4%
	100	84,4%	84,2%
	250	87,2%	86,4%
Topical	30	59,2%	76,6%
	100	81,2%	81,8%
	250	84,0%	84,6%

tyrant.n. For *defeat.v* the best topically similar words are *fight.v*, *lieutenant-colonel.n*, *knight.n*, *whip.n* and *wavell.n*, while *victory.n*, *defeat.n*, *overthrow.v*, *victorious.j* and *fight.v* for the word-based space. Accordingly, the topical space based *SPTK* still assigns ENTITY to Q, while the word space is correctly suggesting HUMAN. The example seems to show that paradigmatic generalizations are captured in the word-based space, whereas *ruler.n* and *defeat.v* are correctly generalized in synonyms (such as *emperor.n/duchess.n* and *overthrow.v/fight.v*, respectively). The document space instead seems to suggest topical similarity (such as *persian.n* vs. *ruler.n* or *knight.n, whip.n* vs. *defeat.v*) that is much less useful for the tree kernel computation.

4 Conclusion

In this work an extensive study of the role of vector space approaches to lexical meaning in tree kernel based natural language learning has been carried out over a Question Classification task. The lexical generalization provided by the word space approaches is always beneficial (87,4 vs. 85,4 - 2,4%), with significant performance improvements in coarse as well as fine grained QC tasks. However, not all the vector spaces are equivalently useful, when they are employed as generalization functions for tree kernels. While document oriented representations, as suggested in [15], are not well suited to support the required lexical generalizations, word spaces with smaller co-occurrence windows seem to capture paradigmatic relations that are quite useful. The improvements achieved in this paper are remarkable. They are in fact providing a novel state-of-the-art on a well known task (i.e. QC) also successfully tackled by previous, more complex, models. This is inline with the results achieved in [16]. Future work will investigate if the beneficial role of proper lexical generalizations in SPTKs is also observable in other tasks, e.g. Semantic Role Labeling. The general outcome of this work

suggests that vector representations are not all equally expressive of the variety of semantic relations they tend to capture, and their employment in semantic NLP tasks must be carefully designed.

References

1. Harris, Z.: Distributional structure. In: Katz, J.J., Fodor, J.A. (eds.) *The Philosophy of Linguistics*. Oxford University Press (1964)
2. Sahlgren, M.: *The Word-Space Model*. PhD thesis, Stockholm University (2006)
3. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37, 141–188 (2010)
4. Schutze, H.: Automatic word sense discrimination. *Journal of Computational Linguistics* 24, 97–123 (1998)
5. Lin, D.: Automatic retrieval and clustering of similar word. In: *Proceedings of COLING-ACL, Montreal, Canada* (1998)
6. Giuliano, C.: Fine-grained classification of named entities exploiting latent semantic kernels. In: *Proceedings of CoNLL 2009, Stroudsburg, PA, USA*, pp. 201–209 (2009)
7. Croce, D., Giannone, C., Annesi, P., Basili, R.: Towards open-domain semantic role labeling. In: *ACL*, pp. 237–246 (2010)
8. Pado, S., Lapata, M.: Dependency-based construction of semantic space models. *Computational Linguistics* 33(2) (2007)
9. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. *Cognitive Science* 34, 1388–1429 (2010)
10. Baroni, M., Lenci, A.: One distributional memory, many semantic spaces. In: *Proceedings of the GEMS 2009 Workshop, GEMS 2009, Stroudsburg, PA, USA*, pp. 1–8 (2009)
11. Clark, S., Pulman, S.: Combining Symbolic and Distributional Models of Meaning. In: *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, pp. 52–55 (2007)
12. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. In: *Proceedings of EMNLP 2011, Edinburgh, Scotland, UK*. (2011)
13. Haussler, D.: Convolution kernels on discrete structures. Technical report, University of Santa Cruz (1999)
14. Collins, M., Duffy, N.: New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In: *Proceedings of ACL 2002* (2002)
15. Bloehdorn, S., Moschitti, A.: Combined Syntactic and Semantic Kernels for Text Classification. In: Amati, G., Carpineto, C., Romano, G. (eds.) *ECIR 2007*. LNCS, vol. 4425, pp. 307–318. Springer, Heidelberg (2007)
16. Croce, D., Moschitti, A., Basili, R.: Structured Lexical Similarity via Convolution Kernels on Dependency Trees. In: *Proceedings of EMNLP 2011* (2011)
17. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet: Similarity - Measuring the Relatedness of Concept. In: *Proc. of 5th NAACL, Boston, MA* (2004)
18. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. *Communications of the ACM* 18 (1975)
19. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review* 104 (1997)
20. Collins, M., Duffy, N.: Convolution kernels for natural language. In: *Proceedings of Neural Information Processing Systems (NIPS)*, pp. 625–632 (2001)
21. Moschitti, A.: Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *ECML 2006*. LNCS (LNAI), vol. 4212, pp. 318–329. Springer, Heidelberg (2006)

22. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*
23. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. In: Brodley, C., Danyluk, A. (eds.) *Proceedings of ICML 2001, 18th International Conference on Machine Learning*, pp. 66–73. Williams College, Morgan Kaufmann Publishers, San Francisco, US (2001)
24. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press (2004)
25. Li, X., Roth, D.: Learning question classifiers. In: *Proceedings of ACL 2002* (2002)
26. Johansson, R., Nugues, P.: Dependency-based syntactic–semantic analysis with PropBank and NomBank. In: *Proceedings of CoNLL 2008*, pp. 183–187 (2008)
27. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *LRE* 43(3), 209–226 (2009)

Multiple Level of Referents in Information State

Gábor Alberti and Márton Károly*

University of Pécs, Department of Linguistics,
Research Team $\mathfrak{R}eALIS$ for Theoretical, Computational and Cognitive Linguistics
Ifjúság 6, H7624 Pécs, Hungary
alberti.gabor@pte.hu, harczymarczy@gmail.com

Abstract. As we strive for sophisticated machine translation and reliable information extraction, we have launched a subproject pertaining to the practical elaboration of “intensional” levels of discourse referents in the framework of a representational dynamic discourse semantics, the DRT-based [14] $\mathfrak{R}eALIS$ [2], and the implementation of resulting representations within a complete model of communicating interpreters’ minds as it is captured formally in $\mathfrak{R}eALIS$ by means of functions σ , α , λ and κ [5]. We show analyses of chiefly Hungarian linguistic data, which range from revealing complex semantic contribution of small affixes through pointing out the multiply intensional nature of certain (pre)verbs to studying the embedding of whole discourses in information state. An outstanding advantage of our method, due to our theoretical basis, is that not only sentences / discourses are assigned semantic representations but relevant factors of speakers’ information states can also be revealed and implemented.

Keywords: representational dynamic discourse semantics, intensionality, information state.

1 Introduction

As we strive for sophisticated machine translation and reliable information extraction, we have launched a subproject pertaining to the practical elaboration of “intensional” levels of discourse referents [1] in the framework of a representational dynamic discourse semantics, the “post-Montagovian” [8], (S)DRT-based [14] [6] $\mathfrak{R}eALIS$ [2] [3], and the implementation of resulting representations within a complete model of communicating interpreters’ minds as it is captured formally in $\mathfrak{R}eALIS$ by means of four functions: the formula-building σ , the anchoring/identifying α , the “box”-level specifying λ and the cursor-like κ [5].

In the first period of the project, thus, as grounding for implementation, we apply theoretical constructions of $\mathfrak{R}eALIS$ (Section 2) to certain groups of linguistic data (chiefly Hungarian data in this period; Sections 3-5). That is, we are striving for

* We are grateful to SROP-4.2.1.B-10/2/KONV/2010/ KONV-2010-0002 (Developing Competitiveness of Universities in the Southern Transdanubian Region) for their contribution to our costs at CILing (2012, Delhi) and ensuring the working of Research Team $\mathfrak{R}eALIS$. The second author is grateful to the foundation “Aktion Österreich–Ungarn” for contributing to the costs of his ongoing researches concerning German.

specified formal representations of affixes and words—which exactly capture their complex intensional nature—such as suffixes of mood and modality, expressions of aspect, and different modal verbs, adverbs, adjectives and particles (e.g. *bevesz* ‘lap up’, *valószínűleg* ‘it is likely that’, *állítólagos* ‘alleged’, *is* ‘also’/‘again’/‘indeed’).

In the second period of the project we implement these representations within a complete model of communicating interpreters’ minds (Section 6). The task of capturing the complex intensional nature of linguistic elements, due to level function λ of \Re ALIS, essentially boils down to assigning a “worldlet index” $\gamma = \langle \langle \mu_1, \tau_1, i_1, \pi_1 \rangle, \langle \mu_2, \tau_2, i_2, \pi_2 \rangle, \dots, \langle \mu_k, \tau_k, i_k, \pi_k \rangle \rangle$, or rather, a set $\Gamma = \{\gamma^1, \gamma^2, \dots, \gamma^N\}$ of indices, to each referent in the DRS-style “box structure” of their discourse-semantic representations in order to show its position(s) / level(s) in this “box structure”. It will turn out soon what this set Γ of sequences of quadruples consists of and how this construction can capture intensionality in different linguistic expressions.

2 Foundations of \Re ALIS

First of all, we should tell some words about the theory that our semantic analyses, DRS-style representations and their computational implementation rely on.

\Re ALIS, *RECiprocal And Lifelong Interpretation System*, can be introduced as a new “post-Montagovian” theory [8] concerning the formal interpretation of sentences constituting coherent discourses [14] [6], with a *lifelong* model of lexical, interpersonal and cultural/encyclopedic knowledge of interpreters in its center including their *reciprocal* knowledge on each other. Its 40 page long formal definition can be found here: [2] <http://lingua.btk.pte.hu/realispapers>; and different aspects and applications of the theory are demonstrated in further publications [1] [3] [5].

What is relevant here is that in this approach, Kamp-style DRSs —gigantic ones, of course— are used as *lifelong* representations of interpreters’ *information states*; and what serve as ilks that play the role of *possible worlds* in (post-) Montagovian discourse-semantic analyses [8] are embedded DRS boxes, which are practically finite information pools not closed under logical operations. Due to unbounded embedding of “boxes”, we can express interpreters’ beliefs / desires / intentions including BDI’s concerning BDI’s (concerning BDI’s)* of each other. An interpreter’s information state, thus, is captured formally as a labeled tree system of “worldlets” (the above mentioned finite information pools) and can be construed practically as the description of his/her brain – his/her “internal world”, which is a part of the entire world model also containing the external world.

\Re ALIS, thus, can be formally defined by *simultaneous recursion* as an epistemic multi-agent system $\Re = \langle W_o, W, \text{Dyn}, \text{Tru} \rangle$ where “agents” (‘interpreters’) get information about the world around them (including each other’s brains). W_o denotes the external world, a “full history” (containing also a temporal dimension), on the basis of which both *static* (truth-conditional) evaluation (Tru) and *dynamic* (DRS-constructing) *interpretation* (Dyn) can be carried out, in cooperation with each other [14]. W is a function where $W[i, t]$ is interpreter i ’s information state (=internal world) at moment t . $W[i, t]$ is a labeled tree system of worldlets, as has been mentioned. The interpretation of *modal* expressions can be based on certain worldlets (the appropriately labeled ones) instead of the external world W (or any kind of “possible worlds”).

What this means is no less than there is simply no *intensionality* in \Re ALIS (in the customary sense) as interpreters’ worldlets (in description of their brains within the

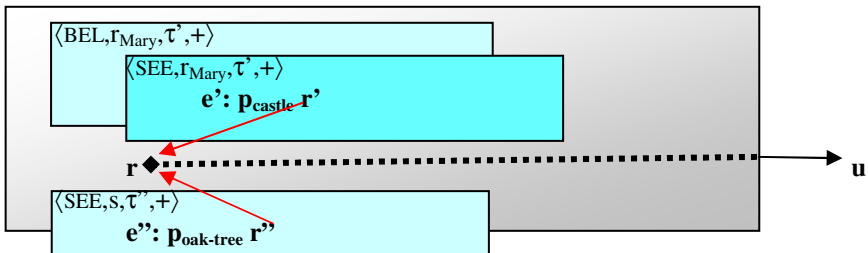
entire model of the universe) carry all kinds of information (BDI, guesswork, dream) typically “entrusted to” possible worlds. In other words, interpretation is always *extensional*, with its basis either W_o or a sector of an interpreter’s $W[i,t]$, or, quite frequently, some combination of the external world and more interpreters’ different worldlets. The hypothesis behind this approach is as follows: all (linguistic) problems whose solution has been held to require “possible worlds” can be solved by means of worldlets.

In this paper the stubborn problem of cross-reference between elements in distinct modal contexts, i.e. *modal anchoring*, will serve as an illustration. The puzzle in the two-sentence discourse below in (1a) is that “a noun phrase [*the castle*] is modally subordinated to a constituent occurring in previous discourse, while the sentence the noun phrase is part of is not modally subordinated” [19:243]. This contradictory case means a serious problem to semantic theories based on the *elimination* of possible worlds, as different parts of the sentence in question require distinct ways of eliminating possible worlds. What correspond to these “distinct modal contexts” in \Re eALIS, however, belong to the same world model (as all internal worlds belong to the same, single, world model); hence, their referents can be anchored to each other (under appropriate conditions).

The representation in (1e) below, for instance, shows the relevant part of an ideal interpreter’s dynamic interpretation. *Dynamic interpretation* of a sentence (or rather, discourse) is defined as extending the interpreter’s information state [2, 2.2]. What practically takes place in the course of this extension is that new sectors are built up in the interpreter’s information state due to the (morpheme by morpheme) consumption of the input performance: there will appear new blocks of partially ordered labeled worldlets. *Static interpretation* (truth evaluation) of a sentence is defined on the basis of the external world W_o or/and the content of certain worldlets of potentially more interpreters [2, 2.3]. It is some “union” of these structures ($W_o + \Sigma W[i,t]$) that the structure of the representation which is the output of dynamic interpretation ($W[i,t]$) is to be compared with: whether a sufficient *pattern matching* can be pointed out.

Example 1. MODAL ANCHORING AS AN EXTREME CASE OF INTENSIONAL IDENTITY

- a. Mary thought that there was a castle behind the trees.
The castle turned out to be a huge oak tree.
- b. General worldlet index: $\gamma = \langle \langle \mu_1, \tau_1, i_1, \pi_1 \rangle, \langle \mu_2, \tau_2, i_2, \pi_2 \rangle, \dots, \langle \mu_k, \tau_k, i_k, \pi_k \rangle \rangle$
- c. The worldlet index of r' : $\gamma' = \langle \langle BEL, r_{\text{Mary}}, \tau', + \rangle, \langle SEE, r_{\text{Mary}}, \tau', + \rangle \rangle$
- d. The worldlet index of r'' : $\gamma'' = \langle \langle SEE, r_{\text{speaker}}, \tau'', + \rangle \rangle$
- e. VISUAL REPRESENTATION OF THE RELEVANT WORLDLETS:



The contribution of the first sentence in (1a) is a referent r' with the piece of information that “Mary believes (at a certain moment τ) that she has seen that it is a castle”. The second sentence contains an assertion concerning something, which must have relied on the speaker’s visual observation (at τ').¹

What corresponds to the well-known “box structure” of DRT [14] is the *labeled* partial ordering of worldlets in $\Re\text{ALIS}$ [2, 1.2.4]. We would like to demonstrate these labels by means of the representation in (1e). They are quadruples providing the following factors of a label: its *modality* (belief / desire / intention / supposition / manner of observation / etc.), its *direct host*, its *moment* and its *polarity* (positive / neutral / negative). What is described by the upper box pair in (1e), then, is that at a moment τ' Mary (r_{Mary}) believes that she can see an eventuality e' , whose content is that something (r') is a castle (what predicate p_{castle} expresses is ‘to be a castle’). The lower single box carries the information that the speaker (s) visually observes at a (later) moment τ'' that something (r'') is an oak tree. (1c-d) above show the *worldlet indices* of r' and r'' (mentioned in Introduction in advance; see (1b)): the thing that Mary thinks, at τ' , to have seen, and the thing the speaker can see a bit later.

Let us return to the phenomenon of modal anchoring, said to be problematic to possible-worlds semantics. The second sentence, containing *the castle*, relies on the speaker’s perspective, and not Mary’s one; nevertheless, the retrieval of the antecedent is successful. How is this possible?

Unicity is the relevant factor: there must be a worldlet containing a referent which is unique in that worldlet in the respect that the content of the singular definite description holds true of it. The second sentence of the two-sentence discourse in (2a) below, for instance, does not meet this requirement, and the discourse is ill-formed, indeed, while there is no change in modal context.

Accessibility is the other relevant factor: The precise solution of the problem in (1a) even requires some *accommodation* because a referent should be *accessible* to the other referent to which we would like to anchor it in order to express their referential identity [13]. *Accessibility* in $\Re\text{ALIS}$ is defined on the basis of worldlet hierarchy – in the most straightforward way: r_1 is accessible to r_2 if the worldlet of r_1 is lower than that of r_2 according to the partial ordering that makes the worldlet hierarchy [2, 2.2.3.6].

The piece of information that should be *accommodated* here, as a result of the singular definite expression in the second sentence of (1a), is that the speaker accepts that “there *is* a huge entity behind the trees, indeed”. The interpreter of the discourse, thus, introduces a referent r to the relative root worldlet belonging to the dynamic interpretation of the discourse (the lowest one). This referent r , then, is accessible to both r' (what Mary thought to be a castle) and r'' (the huge oak tree the speaker can see), so an α type relation (see Introduction) can be formed between r' and r , on the one hand, and r'' and r , on the other; that is, r' and r'' have been *anchored* to r , expressing their coreferential character, i.e. the *identity* of their *designatum* u .

¹ One might think that the precise content of “boxes” in representations like (1e) or (3d) may contain *ad hoc* elements which would not necessarily come from compositional sentence parsing. Our temporary answer here is that what is relevant now is the structure of „boxes”. In Section 5 we study the question from a broader perspective.

One might think that it is suspiciously simple to have recourse to accommodation. We argue, however, that it is a straightforward strategy of speakers that they tend to speak as little as possible and, instead, they tend to entrust as much as (they hold) possible to the interpreter's information state. Instead of attempting to ignore information not expressed explicitly by words in formal semantic analyses, we should strive for capturing this implicit information formally. The "lifelong" approach of *ReALIS* makes it possible to capture implicit information.

Example 2. UNICITY AND ACCOMMODATION

- a. Yesterday we visited two castles in an old town. **The castle* was beautiful.
- b. Peter got married yesterday.
- c. *The minister* spoke very harshly.
- d. ?*The dog* barked very loudly.

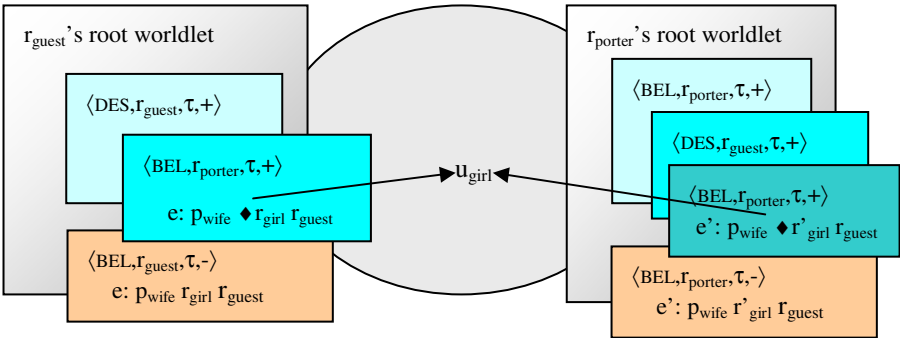
Kálmán's [13] example in (2b)+(2c/d) above is an excellent illustration of accommodation. In our culture a *minister* is a potential "distinguished participant" of a marriage whilst nothing similar holds true of a dog. Nevertheless, it is not excluded that an interpreter considers discourse (2b)+(2d) to be felicitous: what is needed is, say, a piece of interpersonal knowledge about a salient dog in Peter's life. It is relevant that neither the minister in the former case, nor the dog in the latter case can be found in any kind of closure of the interpreter's information state under logical entailment, so the cohesion holds between the content of present sentences and that of ones learned long ago by the interpreter, in an unbounded chronological distance. It is not logically closed possible worlds, thus, which can provide an explanation, but the lifelong perspective offered by *ReALIS*. The singular definite expression in (2c/2d) triggers a procedure in the course of which the interpreter tries to *extend* his/her information state resulting from understanding the first sentence (2b) so that the extended information state contain a worldlet with a unique minister / dog. This task *can* be executed, in the former case, by accommodating encyclopedic information concerning marriage in our Western culture, and it *might* be executed, in the latter case, by accommodating interpersonal information concerning Peter.

In (3a) below we have sketched another context in which a piece of information ("your wife") is used to identify a person, which belongs to different modal contexts to the speaker and to the interpreter and, what is more, is held to be false by both. *ReALIS* enables us to compose an explanation based on *unicity* of the appropriate referents in certain worldlets. What the worldlet blocks in (3d) represent is that *the wife* to the guest is "the unique person in the context such that he wants the porter to believe that she is his wife despite that he knows well that she is not his wife", and *the wife* to the porter is "the unique person in the context such that he thinks that the guest wants him to believe that she is his (the guest's) wife despite that he knows well that she is not the guest's

wife”. In (3b-c) we have provided the formal description of the concerned worldlet indices. Formulae like these capture the mathematical content of what is expressed by visual representations like the one in (3d); and it is on these formulae that a \Re ALIS-based implementation of communicating interpreters can be based.²

Example 3. SUCCESSFUL REFERENCE BY MEANS OF A FALSE PIECE OF INFORMATION:

- a. *A man arrives at a motel in the company of a girl who is not his wife at all in a country where the porter (who knows the girl well...) ought to prevent them, lawfully, to live in the same room. It is against his financial interest, however, to show them the door. Hence, the girl in question will be referred to as the guest’s wife by both the guest himself and the porter in spite of the fact that neither think this “presupposition” to be true and, moreover, neither think that the other considers it to be true either. The porter says, for instance: I hope your wife will enjoy this champagne.*
- b. $\Gamma_e = \{ \langle \langle \text{BEL}, r_{\text{guest}}, \tau, - \rangle \rangle, \langle \langle \text{DES}, r_{\text{guest}}, \tau, + \rangle \rangle, \langle \langle \text{BEL}, r_{\text{porter}}, \tau, + \rangle \rangle \}$
- c. $\Gamma_{e'} = \{ \langle \langle \text{BEL}, r_{\text{porter}}, \tau, - \rangle \rangle, \langle \langle \text{BEL}, r_{\text{porter}}, \tau, + \rangle \rangle, \langle \langle \text{DES}, r_{\text{guest}}, \tau, + \rangle \rangle, \langle \langle \text{BEL}, r_{\text{porter}}, \tau, + \rangle \rangle \}$
- d. VISUAL REPRESENTATION OF THE RELEVANT WORLDLETS:



² The foundations of \Re ALIS may recall instances from the Blending Theory (e.g. [9]), where mental spaces get integrated, and a blend of several mental spaces give ground to a novel meaning structure. In BT, however, mental spaces overlap, and in the intersection a new emergent structure accounts for special, distinct meanings, whereas in the \Re ALIS model distinct worldlets are encapsulated within each other. Both theories aim to account for interpretation of particulars in individual cases of linguistic meaning, both apply a mentalistic approach, and rely on a computer-based, connectionist simulation when modeling the dynamic, interactive process of meaning construction. \Re ALIS, as the acronym suggests, designates a lifelong, reciprocal model of interpretation, where mental states (beliefs, desires, and intentions) play a crucial role in the outcome of linguistic input. While BT analyzes partial realizations of meanings that overlap, \Re ALIS encapsulates different mental states and the representations they bring along, where meanings are dependent upon contextual, situational, cognitive and pragmatic constraints [special thanks are due to Zsuzsa Schnell for this footnote].

3 Modal Adjectives, Adverbs, Connectives, Verbs and Auxiliaries

A related problem is illustrated in (4) below. *Állítólagos* ‘alleged’ (in Hungarian) is qualified as an *irregular* adjective by [17:188] on the basis of its anomalous properties compared to regular adjectives like *old*, as is shown in (4b-c) and (4d-e).

The straightforward solution in our approach is that the difference between regular and irregular adjectives is that the discourse-semantic representation of a regular adjective is a predicate (like p_{castle} and p_{wife} above in (1-3)) whilst the contribution of *alleged* concerns the modal label of a worldlet. The speaker refers to a person by a piece of information to whose truth she does not commit herself (4g) – while she commits herself to the truth of the statement that Mary met somebody (4f). She can refer to him, nevertheless, by expressing that he is “a person such that some people believe that he is a spy” (4h). In this way we could account for both the fact that (4b) *He is a spy* is no correct implication (as the speaker has not committed herself to the truth of this statement) and the fact that (4d) *He is alleged* is ill-formed (*alleged* is not a predicate but a modal label).

Example 4. ÁLLÍTÓLAGOS ‘ALLEGED’: AN IRREGULAR / MODAL ADJECTIVE

- a. Yesterday Mary met an *alleged* spy.
- b. An *alleged* spy is a spy. \rightarrow *not (necessarily) true*
- c. An *old* spy is a spy. \rightarrow *necessarily true*
- d. *He is *alleged*. \rightarrow *ill-formed*
- e. He is *old*. \rightarrow *well-formed*
- f. $\Gamma_{e:\text{met}} = \{ \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, + \rangle \rangle \}$
- g. $\Gamma_{s:\text{spy}} = \{ \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, 0 \rangle \rangle \}$
- h. $\langle \langle \text{BEL}, r_{\text{speaker}}, \tau, + \rangle, \langle \text{BEL}, r^*, \tau, + \rangle \rangle \}$

Modal auxiliaries show similar phenomena – in German, for instance. (5a-b) both imply that the speaker does not commit herself to the truth of state *s*, *viz.* “Peter was ill” ($\langle \langle \text{BEL}, r_{\text{speaker}}, \tau, 0 \rangle \rangle$) in (5c-d). She attributes the statement to someone else (5c) / the subject (5d): the concerned more-quadruple worldlet indices in (5c-d) capture these special points of view.

Example 5. SOLL AND WILL: GERMAN MODAL PREVERBS

- a-b. Peter *soll* / *will* krank gewesen sein. ‘Peter was ill.’ (but see (5c-d))
 Peter *soll* / *will* ill be.PERF be.INF
- c. $\Gamma_{s:\text{ill/a}} = \{ \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, + \rangle, \langle \text{BEL}, r^*, \tau, + \rangle \rangle \}$
- d. $\Gamma_{s:\text{ill/b}} = \{ \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}, r_{\text{speaker}}, \tau, + \rangle, \langle \text{INT}, r_{\text{Peter}}, \tau, + \rangle, \langle \text{BEL}, r^*, \tau, + \rangle \rangle \}$

(6) illustrates the ideal speaker’s information state as a result of the dynamic interpretation of a sentence containing the modal adverb *valószínűleg*. What makes the performance in (6a) fascinating from the view-point of truth evaluation, is that it cannot be qualified as a lying even if *s* (“M. is at home”) is false (6b). The information hearers can obtain rather pertains to the speaker’s information state (at least an ideal one’s one in the Gricean sense

[10] [5]), which we have sketched below as follows: She considers Mary to be likely ($\langle \text{BEL}_{\text{great}}, s, \tau, + \rangle$) to be at home, and she intends to convince her hearers, too, of the likeliness of this state s (6c). But she cannot see s (if she is an ideal speaker); and she considers the hearer to be likely to be in a similar situation (6d). Thus she believes that the hearer can neither see that Mary's at home (6d) nor is sure that that is the case (6e).

Example 6. *VALÓSZÍNŰLEG*: 'IT IS LIKELY THAT': A MODAL ADVERB

- a. Mari valószínűleg otthon van. 'Mary is likely to be at home.'
Mari likely at-home is
- b. It is irrelevant if state s ("Mary is at home") is true "externally", in W_o .
- c. $\Gamma_{s:\text{be-at-home}} = \{ \langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle \rangle, \langle \langle \text{INT}, s, \tau, + \rangle, \langle \text{BEL}_{\text{great}}, i, \tau, + \rangle \rangle \}$,
- d. $\langle \langle \text{SEE}, s, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle, \langle \text{SEE}, i, \tau, 0 \rangle \rangle$,
- e. $\langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, i, \tau, 0 \rangle \rangle$

Connectives also exhibit some intensionality, which is easy to capture formally in $\Re\text{eALIS}$ and to implement by means of worldlet indices. The answer in (7a) cannot be attributed to an ideal speaker if, see (7b), she precisely *knows* ($\langle \text{BEL}_{\text{MAX}} \rangle$) either s' ("Mary's in Delhi") or s'' ("M.'s in Bombay"). It is practically impossible for the speaker to have a *certain* ($\langle \text{BEL}_{\text{MAX}} \rangle$) piece of knowledge concerning what $s' \vee s''$ expresses in traditional logics; modal label $\text{BEL}_{\text{a-max}}$ in (7c) refers to a quite firm but not direct knowledge.

Example 7. *VAGY* 'OR': INTENSIONALITY IN THE MEANING OF A CONNECTIVE

- a. (Where is Mary?) She is in Delhi or in Bombay.
- b. $\Gamma_{s':\text{in-Delhi}} = \{ \langle \langle \text{BEL}_{\text{MAX}}, s, \tau, 0 \rangle \rangle \}$; $\Gamma_{s'':\text{in-Bombay}} = \{ \langle \langle \text{BEL}_{\text{MAX}}, s, \tau, 0 \rangle \rangle \}$
- c. $\Gamma_{s:[s' \text{ or } s'']} = \{ \langle \langle \text{BEL}_{\text{a-max}}, s, \tau, + \rangle \rangle \}$

The intensional "impact" of a Hungarian modal verb (*bevesz* 'lap up') on an interpreter is analyzed in the last example of this section. She can learn that Mary had not believed at τ that Paul was married (s), which is to be regarded (preferably) as false indeed (8b), but at a later moment τ she believed in s (8c). This change is "due" to a wily person r^* (who is not necessarily known). This r^* precisely knew that s was false and she also knew that Mary had not think that it would be true (8d); but she had wished Mary had believed s , and she had intended to convince Mary of s (8e) – successfully (8c). Moreover, the interpreter can also learn that Mary believed at τ that the wily person had believed the same as she believed (that Paul was married), and Mary could not notice that r^* had wanted to make her believe that (8f). This all can be captured formally and then implemented in our approach!

Example 8. *BEVESZ*: 'LAP UP': A MODAL VERB

- a. Mari bevette, hogy Pál nős. 'Mary lapped up the lie that Paul is married.'
Mari lap-up.PAST that Paul married
- b. State s ("Peter is married") is not true in the external world.
- c. $\Gamma_{s:\text{be-married}} = \{ \langle \langle \text{BEL}, r_M, \tau, - \rangle \rangle$ or $\langle \langle \text{BEL}, r_M, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}, r_M, \tau, + \rangle \rangle$,
- d. $\langle \langle \text{BEL}, r^*, \tau, - \rangle \rangle, \langle \langle \text{BEL}, r^*, \tau, - \rangle, \langle \text{BEL}, r_M, \tau, + \rangle \rangle$,
- e. $\langle \langle \text{DES}, r^*, \tau, + \rangle, \langle \text{BEL}, r_M, \tau, + \rangle \rangle, \langle \langle \text{INT}, r^*, \tau, + \rangle, \langle \text{BEL}, r_M, \tau, + \rangle \rangle$,
- f. $\langle \langle \text{BEL}, r_M, \tau, + \rangle, \langle \text{BEL}, r^*, \tau, + \rangle \rangle, \langle \langle \text{BEL}, r_M, \tau, 0 \rangle, \langle \text{INT}, r^*, \tau, + \rangle, \langle \text{BEL}, r_M, \tau, + \rangle \rangle$

4 “Intensionality” of Mood, Modality and Aspect in Hungarian

It is an important part of our project to specify the intensional character of suffixes / preverbs of mood and modality in Hungarian. The table below shows (a simplified analysis of) a few combinations in Past Tense.

Each combination is (at least doubly) ambiguous. They exhibit the ideal speaker's (s) or someone else's (r*) beliefs, desires and/or intentions (BEL, DES, INT) of different intensity (MAX > amax > great > med). BEL_{MAX}, for instance, refers to a piece of sure knowledge. $\langle \text{INT}, r^*, \pi \rangle$ can refer to a person r*'s demand ($\pi=1$; see c., g.), prohibition ($\pi=-$) or permission ($\pi=0$; see a., e.). Label BEL-PART is intended to capture “partial knowledge” (see b., d.: *epistemic* readings): due to the “lifelong” character of *ReALIS*, certain pieces of information are assumed to be associated with a piece of information, serving as its “witnesses”; and the label in question shows that there are worldlets of knowledge in the ideal speaker's information state which contain (a certain amount of) “witness information”. *Hazamehetett*, thus, is modeled as follows: it means either that “someone went home due to some permission” (a.) or that “someone is quite likely to have gone home because, for instance, she cannot be found in her office, her coat and umbrella cannot be found there either, it is already 18¹⁰, etc.” (b.).

Modality → ↓Mood	<i>haza-megy + -(V)t</i> home-go + PAST		<i>haza-megy + -(V)t + vol- + -nA</i> home-go + PAST + COPULA + COND	
	<i>haza-me-het-ett</i>		<i>haza-me-het-ett vol-na</i>	
-hAt 'can/may'	a. $\langle \text{INT}, r^*, 0 \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, + \rangle$	b. $\langle \text{BEL}_{\text{med}}, S, + \rangle$ $\langle \text{BEL-PART}_{\text{great}}, S, + \rangle$	e. $\langle \text{INT}, r^*, 0 \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, - \rangle$	f. $\langle \text{DES}_{\text{great}}, S, + \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, - \rangle$
	<i>haza kell-ett \swarrowmen-ni(e) / menni \swarrow</i>		<i>haza kell-ett vol-na \swarrowmen-ni(e) / menni \swarrow</i>	
kell 'must'	c. $\langle \text{INT}_{\text{MAX}}, r^*, + \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, + \rangle$	d. $\langle \text{BEL}_{\text{amax}}, S, + \rangle$ $\langle \text{BEL-PART}_{\text{MAX}}, S, + \rangle$	g. $\langle \text{INT}_{\text{MAX}}, r^*, + \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, - \rangle$	h. $\langle \text{DES}_{\text{amax}}, S, + \rangle$ $\langle \text{BEL}_{\text{MAX}}, S, - \rangle$

Fig. 1. Mood and Modality in Past Tense in Hungarian

Intensional factors of aspect can be modelled in our approach alike. Let us consider, for instance, the progressive Hungarian sentence below in (9b). Owing to this aspect, there emerges a phenomenon called the Imperfective Paradox [7:147]: the truth value of (9b) cannot be calculated exclusively on the basis of external facts. What is to be checked externally is the period of time until 17.10 on the day in question, that is, only a part of the entire eventuality e (9d). The ideal speaker holds it likely, but not sure, that e had been successfully accomplished (9c). For the period of time from 17.10, “internal” factors become relevant: preferably the subject's intention concerning his traveling home (9e). It is worth mentioning that (one of the meanings of) Future Tense (9f) can be characterized by the same intensional pattern (9c-d). Past Progressive, thus, is practically some “Future in the Past”.

Example 9. HUNGARIAN PROGRESSIVE ASPECT (AND THE IMPERFECTIVE PARADOX)

- ‘What was Peter doing at 5.10 pm on May 4, 2003?’
- Utazott (éppen) haza. ‘He was traveling home.’
travel-PAST-3SG (just) home
- $\Gamma_{\text{e:travel}} = \{ \langle \langle \text{BEL}_{\text{MAX}}, S, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}_{\text{great}}, S, \tau, + \rangle \rangle \}$

- d. $\langle\langle \text{BEL-PART}_{\text{MAX}}, s, \tau, + \rangle\rangle$
 e. $\langle\langle \text{INT}, r_{\text{Peter}}, \tau, + \rangle\rangle \}$
 f. Péter haza fog menni. ‘Peter will travel home.’
 Peter home will go.INF

5 Embedding Information in Interpreter’s Information State

We have reviewed the impact of the intensional character of different kinds of lexical items on the final output of dynamic interpretation. There are also pragmatic impacts.

It is obvious, for instance, that *irony* will simply reverse the polarity label of a certain worldlet ($\pi=-$). In other cases it may be figured out that the speaker is bluffing; the corresponding worldlet label, hence, should be $\pi=0$. Have we obtained no information? That is not the case: we could obtain information concerning, instead of the external world, the speaker’s suspicious intentions...

It is also quite easy, due to our reciprocal and lifelong theoretical basis, to model an interpreter’s *reliability* or the reliability of pieces of information. What is to be compared are different interpreters’ intensional worldlet patterns pertaining to one and the same eventuality and/or fixed interpreters’ intensional worldlet patterns pertaining to different eventualities. The simplest principle to be considered is that the same information from independent sources is more reliable, and a coincidence like this can enable us to qualify these sources as more reliable. Such principles are to control the percolation of information in the ideal interpreter’s partially ordered worldlet structure and the calculation of the deviation of the “sources” from the default picture of “ideal speakers”, often mentioned in Sections 2-4.

As *ReALIS* is a “reciprocal” and “lifelong” multi-agent system of communicating interpreters, we can implement an intensional model of different question types. A simplified model of wh-questions is illustrated in (10a-e) below. Referent r^* is defined in (10b) as Pál’s wife at the relevant moment t . Then e^* (10c), which is also about r^* , is to be placed in the worldlet pattern defined in (10d), with an unanchored (unidentified) predicate p^* . Because of the interrogative form, e^* is such that (10d) the speaker cannot decide its truth value, but intends to do that, and she holds the hearer to be likely to know its truth value, and she hopes that the hearer will be prepared for giving her the relevant piece of information. Predicate p^* has been said to be unanchored in (10c); it is the pragmatic principle of “Maximize Discourse Coherence” [6] that presses the hearer to anchor p^* to a lexical item as profitably as possible. The profitability of an answer can be calculated on the basis of the increment of the speaker’s information state. Answer1 in (10e) is obviously the least favorite answer as it yields no increment. Answer3 is better than answer2 if, and only if, the speaker knows the person mentioned, because finding the referent of an identified person makes available all information that has been linked to it so far.

Example 10. THE QUESTION OF QUESTION

- a. Ki volt Pál felesége akkoriban? ‘Who was Paul’s wife at that time?’
 who was Paul wife.POSSG3 THEN
 b. $e: p_{\text{wife}} t r^* r_{\text{Pál}}$

- c. e^* : $p^* t^* r^*$
- d. $\Gamma_{e^*} = \{ \langle \langle \text{BEL}_{\text{MAX}}, s, \tau, 0 \rangle \rangle, \langle \langle \text{INT}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, s, \tau, + \rangle \rangle, \langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, h, \tau, + \rangle \rangle, \langle \langle \text{DES}, s, \tau, + \rangle, \langle \text{INT}, h, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, s, \tau, + \rangle \rangle \}$
- e. Answer1. ‘A woman.’
 Answer2. ‘A waitress of our favorite Indian restaurant.’
 Answer3. ‘The admirable Shabana Singh.’
- f. Ki *is* volt Pál felesége akkoriban?
 ‘Who was Paul’s wife at that time, again?’
- g. $\Gamma_{e^*}^+ = \{ \langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, s, \tau, + \rangle \rangle, \langle \langle \text{BEL}_{\text{amax}}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, h, \tau, + \rangle \rangle, \langle \langle \text{BEL}_{\text{great}}, s, \tau, + \rangle, \langle \text{BEL}_{\text{amax}}, h, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, s, \tau, + \rangle \rangle \}$
- h. Tunteeko Pekka Marjan / Marjaa?
 know-SG3-Q Peter Mary-ACC / Mary-PART
 ‘Does Peter know Mary?’ (e: $p_{\text{know}} t_{\text{Peter}} r_{\text{Mary}}$)
- i. $\Gamma_e = \langle \langle \text{BEL}_{\text{MAX}}, s, \tau, 0 \rangle \rangle, \langle \langle \text{BEL}_{\text{great}}, s, \tau, + / - \rangle \rangle, \langle \langle \text{INT}, s, \tau, + \rangle, \langle \text{BEL}_{\text{MAX}}, s, \tau, + / - \rangle \rangle$

(10f) shows a special form of question with *is* ‘also’, which is used in this construction as a peculiar discourse particle. Its intensional contribution is described in (10g): the speaker is sure that she used to know e^* ($\tau'' < \tau$), and she is almost sure that the hearer still knows it; and, preferably, she is also quite sure that the hearer knows that she used to know e^* .

A yes/no question is an indication that the speaker is sure neither that a certain e is true nor that it is false, and she wants to know the truth. The specialty of the Finnish example in (10h) is that the case marking of the object —Accusative or Partitive— indicates if the speaker expects a positive or a negative answer, respectively.

6 Implementation of λ

Taking everything into account that we stated above, one can conclude that the implementation of λ is fairly simple. To do this, the appropriate values of Γ_r are stored for each referent r , and they are used for determining the actual value of λ or calculating its value for any new referent.

We use Prolog as the primary programming language, as well as Contralog, an extension to Prolog which makes easier to implement the theory of $\mathfrak{R}eALIS$ by using forward chaining, being under development [18]. Experimental implementation (including $\mathfrak{R}eALIS$ ’ predecessor, the GeLexi project [4]) and study programs [18] were made for the formula-building function σ , too.

For λ we just use the list handling mechanism of Prolog to create new worldlets, if needed, for any new referent. In an earlier period of our work, even in [16], only one γ index was permitted for each referent but this turned out to be inadequate when we tried to describe sentences/pragmatic situations like the ones shown in (3) in Section 2, where we do not use α to connect the two instances of e , e' , r_{girl} etc. at all (r_{girl} is just α -anchored to the external entity of u_{girl} and not to the other instance of r_{girl}). Instead, we use doubly-embedded lists to represent the trees of Γ_r as shown in the examples below.

Second, our predicate `lambda` should have a feature to describe entities and even *infons* [2, 1.1.7] of the outer world. As stated in [2, 1.2.4], every referent has its *direct owner*. Entities do not have one and a set of numbers (e.g. the negative numbers) will be reserved for them. The level label list belonging to any entity in the outer world is declared empty. What is, indeed, still a question, is the representation of the *root* worldlet of any interpreter, just like the existence of polarity and time in root worldlets. But if any of these exists, we would need multiple root worldlets and this is what we should avoid because of psycholinguistic reasons. Therefore, one should conclude that the Γ -structure representing a root worldlet of any interpreter *i* is also empty – with no time or polarity. Instead, the direct owner of each referent should be included in `lambda` as a third argument.

Therefore, λ is represented as a set of `lambda/3` facts. The first and second arguments are the identifiers of the referent and the owner, respectively. The third argument is a doubly embedded list of γ -series of labels, but taking the potential complexity of Γ into account, we are considering introducing a tree-like structure (of embedded Prolog lists) with no level limit.

So the Prolog equivalents of the λ -levels describing (3) now look like this:

```
ref(1,r,'guest'). ref(2,r,'porter'). %guest's referents
ref(11,p,'wife'). ref(12,r,'girl'). ref(13,e,'11(12,1)').

ref(21,r,'guest'). ref(22,r,'porter'). %porter's referents
ref(31,p,'wife'). ref(32,r,'girl'). ref(33,e,'31(32,21)').
```

Note: referents are constructed as a result of grammatical analysis. Their σ - and α -relationships (which are less relevant for now) are also determined on-the-fly where function `alpha` describes the (permanent or temporary) identification of referents and entities.

```
lambda(1,1,[]). %the guest is the interpreter itself
alpha(1,-1). %entity -1 is e.g. 'John Peterson'
lambda(11,1,[[[sub,des,med,1,_T,+1],[sub,bel,med,2,_T,+1]],
[[sub,bel,max,1,_T,-1]]]).
lambda(12,1,[[[sub,des,med,1,_T,+1],[sub,bel,med,2,_T,+1]],
[[sub,bel,max,1,_T,-1]]]).
lambda(13,1,[[[sub,des,med,1,_T,+1],[sub,bel,med,2,_T,+1]],
[[sub,bel,max,1,_T,-1]]]).
alpha(12,-2). %2 is 'Nathalie Cardiff'
alpha(2,-22). %the porter in John's mind is also 'Kevin Johnson'

lambda(22,22,[]). %the porter as an interpreter
alpha(22,-22). %entity -22 is e.g. 'Kevin Johnson'
lambda(31,22,[[[sub,bel,med,22,_T,+1],[sub,des,med,21,_T,+1]],
[sub,bel,max,22,_T,+1]], [[sub,bel,max,22,_T,-1]]]).
lambda(32,22,[[[sub,bel,med,22,_T,+1],[sub,des,med,21,_T,+1]],
[sub,bel,max,22,_T,+1]], [[sub,bel,max,22,_T,-1]]]).
lambda(33,22,[[[sub,bel,med,22,_T,+1],
[sub,des,med,21,_T,+1],[sub,bel,max,22,_T,+1]],
[[sub,bel,max,22,_T,-1]]]).
alpha(32,-2). %also 'Nathalie Cardiff'
alpha(21,-1). %also 'John Peterson'
```

Modal verbs and verbs with a modal meaning (like *to desire*), adverbs (*perhaps*), adjectives (*alleged*) and morphemes (Conditional) set the values of λ facts inherently. To determine the exact way of this process (and to determine possible pragmatic relations also described by λ) is mostly ongoing and future work, although with results which are partially published [15]. Most importantly, there are *level-changing* and *level-keeping* words and morphemes [2,1.2.4.1], all waiting for an accurate description in the frames of \Re ALIS, along with their implementation. For instance, *to desire* (temporarily) creates a new modal level, thereby extending Γ .

In addition to earlier applications of \Re ALIS-based systems in computational linguistics (sophisticated machine translation and NL-based decision supporting systems, high-level grammar checkers, checking consistency of human-made translations), we intend to handle linguistic data from less frequently used languages, many of which simply have too few speakers to build a corpus, e.g. the Sámi languages (a branch of the Finno-Ugric language family). What makes \Re ALIS suitable for this task is its “totally lexicalist” approach, according to which any linguistic analysis is based on lexical rules. A similar system (based on transducers and also approaching languages in a strongly lexicalist way) is being built by Moshagen and his colleagues [10] [12] and will be able to analyze different Sámi dialects among other languages. An important feature of Moshagen’s system is that it solves the problem of analyzing composite words. \Re ALIS, indeed, is based on semantics and pragmatics and does not stick to the Chomskyan view of the Autonomy of Syntax. Syntactic features will be calculated from semantic ones (if our target is to generate NL output) and backwards (if we analyze source-language texts).

Therefore, in the long run, a \Re ALIS-based system might be able to produce a more precise analysis of NL input than any other rule-based NLP system.

References

1. Alberti, G.: Accessible referents in “opaque” belief contexts. In: Ditmarsch, H., Herzig, A. (eds.) Proc’s of 9th ESSLLI Belief Revision and Dynamic Logic Workshop, pp. 1–7 (2005)
2. Alberti, G.: \Re ALIS: An Interpretation System which is Reciprocal and Lifelong. In: Workshop ‘Focus on Discourse and Context-Dependence’. Center for Language and Comm., Amsterdam (2009), <http://www.hum.uva.nl/acllc/events.cfm/C2B8E596-1321-B0BE-6825998CFA642DB2>, <http://lingua.btk.pt.e.hu/realispapers>
3. Alberti, G.: \Re ALIS: Interpreters in the World, Worlds in the Interpreter. Academic Press, Budapest (2011) (in Hungarian)
4. Alberti, G., Kleiber, J.: The *GeLexi* MT Project. In: Hutchins, J. (ed.) Proceedings of EAMT 2004 Workshop, pp. 1–10. Univ. of Malta, Malta (2004)
5. Alberti, G., Kleiber, J.: The Grammar of \Re ALIS and the Implementation of its Dynamic Interpretation. *Informatica* 34/1, 103–110 (2010)
6. Asher, N., Lascarides, A.: *Logics of Conversation*. Cambridge Univ. Press (2003)
7. Dowty, D.R.: *Word Meaning and Montague Grammar*. Reidel, Dordrecht (1979)
8. Dowty, D.R., Wall, R.E., Peters, S.: *Introduction to Montague Semantics*. D. Reidel Publishing Company, Dordrecht (1981)

9. Fauconnier, G., Turner, M.: Conceptual integration networks. *Cognitive Science* 22(2), 133–187 (1998)
10. Gaup, B., Moshagen, S., Omma, T., Palismaa, M., Pieski, T., Trosterud, T.: From Xerox to Aspell: A First Prototype of a North Sámi Speller Based on TWOL Technology. In: Yli-Jyrä, A., Karttunen, L., Karhumäki, J. (eds.) *FSMNLP 2005. LNCS (LNAI)*, vol. 4002, pp. 306–307. Springer, Heidelberg (2006)
11. Grice, H.P.: Logic and Conversation. In: Cole, P., Morgan, J.L. (eds.) *Syntax and Semantics. Speech Acts*, vol. 3, pp. 41–58. Academic Press, New York (1975)
12. Huhmarniemi, S., Moshagen, S., Trosterud, T.: Usage of XSL Stylesheets for the Annotation of the Sámi Language Corpora. In: *Proc. of the Linguistic Annotation Workshop*, pp. 45–48. ACL, Prague
13. Kálmán, L.: Deferred Information: The Semantics of Commitment. In: Kálmán, L., Pólos, L. (eds.) *Papers from the 2nd Symp. on Logic and Lang.*, pp. 125–157. Akadémiai, Bp (1990)
14. Kamp, H., van Genabith, J., Reyle, U.: Discourse Representation Theory. In: Gabbay, D., Guenther, F. (eds.) *Handbook of Phil. Logic*, vol. 15, pp. 125–394. Springer, Berlin (2011)
15. Károly, M.: Interpretáció és modalitás – avagy a $\mathfrak{R}eALIS$ lambda függvényének implementációja felé. In: Tanács, A., Vincze, V. (eds.) *MSzNy 2011*, pp. 284–296. Departments of Informatics, Szeged (2011)
16. Károly, M., Alberti, G.: The Implemented Human Interpreter as a Database. In: Cordeiro, J., Virvou, M. (eds.) *Proceedings of IC3K the 5th Int. Conf. on Software and Data Technologies*, vol. 2, pp. 468–474. SciTePress, Funchal (2011)
17. Kiefer, F.: *Jelentélmélet [Semantics]*, Corvina, Budapest (2000)
18. Kilián, I.: Tárgymodell változatok a $\mathfrak{R}eALIS$ nyelvi elemzéséhez. In: Tanács, A., Vincze, V. (eds.) *MSzNy 2011*, pp. 276–283. Departments of Informatics, Szeged (2011) (in Hungarian)
19. Kilián, I.: Tárgymodell változatok a Roberts, C.: Anaphora in Intensional Contexts. In: Lappin, S. (ed.) *The Handbook of Contemporary Semantic Theory*, pp. 215–246. Blackwell, Oxford (1996)

Inferring the Scope of Negation in Biomedical Documents

Miguel Ballesteros¹, Virginia Francisco²,
Alberto Díaz¹, Jesús Herrera¹, and Pablo Gervás²

¹ Departamento de Ingeniería del Software e Inteligencia Artificial

² Instituto de Tecnología del Conocimiento

Universidad Complutense de Madrid

C/ Profesor José García Santesmases, s/n

E-28040 Madrid, Spain

{miballes,virginia,albertodiaz,jesus.herrera}@fdi.ucm.es,

pgervas@sip.ucm.es

Abstract. In the last few years negation detection systems for biomedical texts have been developed successfully. In this paper we present a system that finds and annotates the scope of negation in English sentences. It infers which words are affected by negations by browsing dependency syntactic structures. Thus, firstly a greedy algorithm detects negation cues [1], like *no* or *not*. And secondly the scope of these negation cues is computed. We tested the system over the Bioscope corpus, annotated with negation, obtaining competitive results. The system presented in this paper can be accessed via web [2].

1 Introduction

Generally speaking, negation turns an affirmative statement into negative (*I want / I do not want*) and indicates if it is a statement or not. It is also a complex phenomenon in natural languages and it has been an active research topic for decades. Researchers have approached this topic from both linguistic and philosophical perspectives [2]. In most cases, negation involves a negation cue and a negated syntagma containing one or more words that are within the scope of negation. In the following example: “*There is no detectable effect on leg segmentation*”, ‘*no*’ is the negation cue used to denote that the following concept is negated, being “*detectable effect on leg segmentation*” the negated syntagma.

Nowadays negation detection is an emergent task in natural language processing. Detecting uncertain and negative assertions is essential in most text mining tasks where, in general, the aim is to derive factual knowledge from textual data. For instance, in text mining extracted information that is within the scope of negation should either be discarded or presented separately from factual information.

¹ A negation cue is defined as the lexical marker that expresses negation [1].

² <http://minerva.fdi.ucm.es:8888/ScopeTagger>

Negation is commonly seen in clinical documents and it is an important source of low precision in automated indexing systems [3]; as it is evidenced in Chapman’s work when querying large medical free-text databases, the presence of negations can yield numerous false-positive matches, because the medical personnel is trained to include pertinent negatives in their reports. In a search for *fracture* in a certain radiology reports database, 95 to 99 percent of the returned reports would state “*no signs of fracture*” or words to that effect. Therefore, to increase the utility of indexing medical documents, it is necessary to acknowledge whether words have been negated or not.

In this paper we present a system that annotates the scope of negation, making use of a simple technique: first a manually-defined set of keywords is matched, then an algorithm marks the range of the scope within the sentence using unlabelled dependency syntactic structures. Finally, we evaluated the system with an established corpus annotated with the scope of negations, Bioscope [4], with three different collections: (i) the Papers collection, (ii) the Clinical reports collection and (iii) the Abstracts collection, containing 12.70%, 13.55% and 13.45% of sentences containing negations respectively.

In Section 2 we discuss previous works on processing negation. In Section 3 we describe the algorithms that we propose for inferring the scope of negation. In Section 4 we discuss the evaluation performed and, finally, in Section 5 we give our conclusions and suggestions for future work.

2 Previous Work

Nowadays there are two main kinds of systems that work with negation: systems that detect wordforms affected by negations, and (more recent) systems that classify the whole scope of negations, which is a more difficult task. Our system is classified in the second kind of systems. There is also a trend on works that try to extract negated events, such as [5].

For the biomedical domain there is plenty of research studying negation and finding how to detect it. For instance, Chapman et al. [6] detected negations and identified medical terms affected, by means of a simple regular expression algorithm called NegEx. It achieves 84.5% precision and 77.8% overall recall over 400 randomly selected sentences. In a similar way Mutalik et al. [7] recognized negated patterns in biomedical texts by using a training set of 40 medical documents; the set was manually inspected and used to develop a rule-based system (Negfinder), able to recognize a set of negated patterns in texts. They showed very good evaluation results, verified by human interaction, yielding 95.7% recall and 91.8% precision. Also, Huang and Lowe [8] implemented a hybrid approach to an automated negation detection system. They combined regular expression matching with grammatical parsing, to check the limits in automatically detecting negations in clinical reports. Their approach identified negated phrases with 98.6% precision and 92.6% recall in a test set of 120 reports.

On the other hand, there are systems that infer the scope of negation. This is a more difficult problem, because it involves determining the words that are

within the scope of a negation cue, where to open the scope, and finally, where to close it. One of the main works has been carried out by Morante's team [9,10] in which a machine learning approach for the biomedical domain is shown. The system was evaluated with the Bioscope corpus and their results were: 80.11% overall precision and 78.44% recall in finding scopes of negation.

In 2010, a Workshop on Negation and Speculation in Natural Language Processing [11] was held in Uppsala, Sweden, bringing together researchers working on negation and speculation from any field related to computational language learning and processing. A specific goal was to describe the lexical aspects of negation to define how this phenomenon could be modelled for computational purposes, to explore techniques and to analyze how its treatment affects the efficiency of Natural Language Processing applications. Most of the approaches presented in the Workshop were in the biomedical domain, which is probably the most studied one in negation detection. An interesting paper for our work was presented in that workshop by Council et al. [12]. They used the Bioscope corpus to evaluate their scope finding system, that is based on dependency parsing, and their results were 78.2% recall and 81.9% precision.

Later, Zhu et al. [13] presented a unified framework for scope learning by means of shallow semantic parsing, evaluating it with the Bioscope corpus. They divided the process in three main steps and they carried out the evaluation considering *golden cues* (which means that their system does not need to find where the cue is and which one it is), and *golden trees* (which means that their system does not need to find how the correct tree is, because it is given). They also reported their results when the system should predict correctly the tree and the cue. Their work was focused on inferring the scope of negation, but also on speculation. Their results, without using golden cues were 72.53% recall and 72.24% precision. When the *golden cue* was given, they were notably higher. This means that such kind of system in synergy with an accurate cue classifier could be an interesting option to achieve very high results.

And finally, another interesting approach is presented by Agarwal and Yu [14]. They achieved an F1-score of 98% and 95% on detecting negation cue phrases and their scope in clinical notes, and an F1-score of 97% and 85% on detecting negation cue phrases and their scope in biological literature, they also included a website to test the system. ³

3 Negation Scope Finding

Our system consists of two algorithms: the first one is capable of inferring words affected by the negative operators (cues) traversing dependency trees and the second one is capable of annotating sentences within the scope of negations. This second algorithm consists of a set of rules that have been built making use of a developing set, which was extracted from the Papers collection of the Bioscope corpus, more concretely, this developing set was formed by the first 10% of the

³ <http://snake.ims.uwm.edu/negscope/index.php>

sentences that appear in the Bioscope Scientific Papers Collection (this set of sentences is removed in the Evaluation presented in Section 5).

In the first algorithm our system traverses a dependency tree, searching for negation cues to determine the correct scope over the tree. Our contribution lies in the identification of the scope, which is not explicit in the dependency tree. We selected Minipar parser [15] to develop the present experiment, which is a rule-based dependency parser capable of perform high accuracy unlabelled parsing (which is what we need for the present experiment). We selected Minipar because we only need unlabelled parsing and there is no need to train the system collecting annotated corpora.

Therefore, our system works as follows: a parse given by Minipar and a negation cue lexicon (as the one described in Section 3.1), are the inputs for the Affected Wordforms Detection Algorithm described in Section 3.2. Then, the Scope Finding Algorithm described in Section 3.3 acts on the set of negated nodes returning an annotated sentence with the scope of negation.

3.1 Negation Cue Lexicon

To determine the scope of negation, first of all a set of negation cues must be established. We considered the guidelines presented in [16] to set up our negation cue lexicon and also the work done by Mutalik et al. [7] in which some of the negation cues are presented.

The lexicon used to configure our system is shown in Table 1. This lexicon only contains the lemmas of each wordform, but our system is able to parse not only the lemma but all kind of verb forms.

Table 1. Lemmas of the Bioscope negation cues contained in our Negation Cue lexicon

not	no	neither..nor	none
discard	rule out	fail	avoid
absence	lack (v)	lack (n)	without
unable	rather than	absent	cannot

These negation cues are the ones selected to develop the present work, but similar analyses can be accomplished with a different set.

3.2 Affected Wordforms Detection Algorithm

We implemented an algorithm that takes the dependency tree for a sentence returned by the dependency parser and returns for each negation cue a set of words affected by the cue. It uses the lexicon of negation cues presented in the previous section.

The algorithm traverses the dependency tree of a sentence, and it carries out the following steps:

1. It detects all the nodes that are contained in the lexicon of negation cues.
 - If the negation cue is a verb, it is marked as a negation cue.
 - If the negation cue is not a verb, the algorithm marks the verb (if exists) affected by it as a negation cue. In this way, the words that depends on the verb are affected by the negation cue.
2. For the rest of nodes, if a node depends directly on any of the ones previously marked as a negation cue, the system marked it as negated. Moreover, the negation is propagated from the cue word through the dependency graph until finding terminals, so wordforms that are not directly related with the cue are detected too.

The algorithm generates finally a set of nodes containing the wordforms within the scope of negation cues involved in the sentence. It is worth to emphasize that our system uses the same process for all the different negation cues, that makes the system easy to adapt to different domains.

3.3 Scope Finding Algorithm

This second algorithm is implemented using as an example a subset of the Scientific Papers collection of the Bioscope corpus (first 10% of the sentences containing negations) in order to give shape to the rules involved in this algorithm. We selected the Scientific papers collection to come up with the rules because it contains much more variety of sentences.

It works with the set of words returned by the Affected Wordforms Detection Algorithm, described in the previous Subsection, and the dependency tree given by the dependency parser in order to annotate sentences with the scope of negation, inferring where the scope must be opened and where it must be closed.

It is worth to emphasize that when the scope is opened to the right of the negation cue, the scope of negation leaves the subject out. This correspond to sentences in active voice and they are the most frequent case. Additionally, there are some cases in which the scope is opened to the left of the negation cues. The most frequent one is the passive voice. As shown in [17], passive voice is an exception in the way of tagging sentences in Bioscope. In this case the subject is marked within the scope of negation, because if the sentence had been written in active voice, it would be the object of a transitive verb. Therefore, the first step is to decide where to open the scope of negation, which is related to the voice of the sentence: if the sentence is in passive voice the scope must be opened to the left of the negation cue and if the sentence is in active voice, the scope must be generally opened to the right of the cue.

Therefore, we considered that the Scope Finding Algorithm must be divided in two main processes: first, to detect if the sentence follows a negated passive voice structure or not, and second, to annotate the sentence with the scope of negation, or scopes if there is more than one (which is an usual situation).

A sentence is in passive voice if:

- It contains a transitive verb, such as, *show, consider, see, use, detect*, etc.
- It follows one of the patterns shown below:
 1. *modal verb + not + be + past participle.*
 2. *am/is/are/was/were + not + past participle.*
 3. *have/has been + not + past participle.*

Once our system has decided if the sentence is in passive voice or not, the Scope Finding Algorithm iterates the sentence, token by token and applies a set of rules about the scope opening and closing. The rules are applied in the order presented below and it only applies one rule for each token.

1. Scope opening:
 - a. If the token is contained in the set of nodes marked as negated by the Affected Wordforms Detection Algorithm and the scope for the cue involved is not open: the system opens the scope at the token and establishes that the scope for the cue involved is already opened.
 - b. If the token is a negation cue and the sentence is in passive voice: the system goes backward and opens the scope just before the subject of the sentence. The system opens and closes the cue at the token.
 - c. If the token is a negation cue and the sentence is not in passive voice: the system opens the scope just before the token. The system opens and closes the cue at the token.
2. Scope closing:
 - a. If the token is a punctuation symbol, followed by some wordforms that indicates another statement, such as *but*: the system closes the scope just after the token.
 - b. If the token is any wordform and all the nodes that are marked as negated for the negation cue are already included in the scope: the system closes the scope just before the token.
 - c. If the token is the end of sentence: the system closes the scope at the end of the sentence.
3. Adding words to the sentence: if none of the previous rules has been applied the token is added to the annotated sentence.

At this point, the system has computed the scope (scopes) of the negation (negations) for a given sentence, by inferring which nodes pertain to that scope (scopes) from the node (nodes) marked as negated. Figure 4 illustrates the processing of the following sentence: *The reason why the two other families were not detected is more complex.* In the figure, the potential usefulness of the syntactic structure to infer the scope of negation is evidenced.

4 Evaluation

In this Section we present the evaluation performed to test how good our system is. In Section 4.1 we present the design of the evaluation as well as the evaluation metrics, in Section 4.2 we show the results considering these metrics, and finally, in Section 4.3 we compared our results with other state of the art approaches.

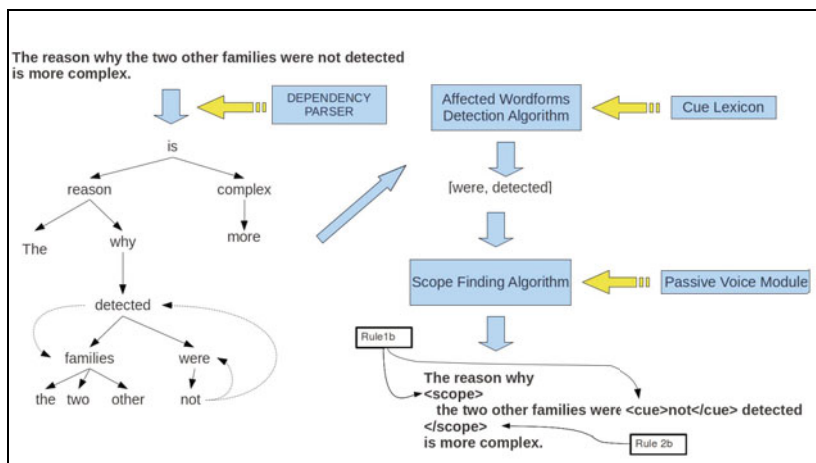


Fig. 1. The processing of a sentence by our system. As shown in the Figure, the rule applied to open the scope and open and closes the cue is *1b*. Finally, the rule applied to close the scope is *2b*, wherein the system closes the scope because there are no more wordforms marked as affected by the algorithm described in Section 3.2.

4.1 Evaluation Design

We selected Bioscope as the corpus for our evaluation. It is worth to emphasize that the evaluation is carried out over the output of the Scope Finding Algorithm, which is the output of the whole system and uses the Affected Wordforms Detection Algorithm.

Our first step in order to get the results was to select the sentences containing negations in the three collections of Bioscope, considering that for the Scientific Papers collection we must remove 10% of these sentences because this is the data set used to develop the rules of the algorithm presented in Section 3.3. Therefore, we evaluated our system with 100.0% of the clinical sentences containing negations, 90.0% of the papers sentences containing negations and 100% of the abstracts sentences containing negations.

The following evaluation measures were used:

$$P(\textit{Precision}) = \frac{\textit{Tokens correctly negated by our system}}{\textit{Tokens negated by the system}}$$

$$R(\textit{Recall}) = \frac{\textit{Tokens correctly negated by our system}}{\textit{Tokens negated in the collection}}$$

To balance the results in recall and precision, we used micro F1.

$$F1 = \frac{2PR}{P + R}$$

Additionally, we evaluate our system with the percentage of correct scopes (PCS), and the percentage of correct negation cues (PCNC).

$$PCS = \frac{\textit{Correct Scopes annotated by our system}}{\textit{Scopes annotated in the collection}}$$

$$PCNC = \frac{\textit{Correct negation Cues annotated by our system}}{\textit{Negation Cues annotated in the collection}}$$

By using all these measures we are considering not only a token-based evaluation but a whole scope classification measure, that really shows how good is a system that classifies the scope of negation in sentences.

4.2 Results and Discussion

When parsing the three collections of Bioscope, our system obtained the results given in Table 2.

It is worth to emphasize that we did scope identification with automatic cue recognition, so the input of our program, as shown in Section 3, is the sentence without any extra information.

Table 2. Results of our work, when evaluating it with the three collections of Bioscope

Collection	Precision	Recall	F1	PCS	PCNC
Papers	73.49%	80.70%	76.93%	56.43%	91.15%
Abstracts	84.92%	84.03%	84.48%	68.92%	95.56%
Clinical	95.83%	90.58%	93.13%	89.06%	94.82%

The different results could be explained as follows. The main reason for the better results that our system achieves when annotating the sentences from the Clinical Reports collection is that the sentences contained in the clinical reports collection followed very easy syntactic structures and most of them contain the scope to the right. The average sentence length in the clinical reports collection is 7.73 wordforms, while in the papers and abstracts collections is more than 26 wordforms [10]. Moreover, a lot of sentences in the abstracts and papers collections contain more than one negation cue, which are more difficult to parse than those having only one. Finding the scope of negation in the simpler sentences of the clinical reports collection is easier than finding it in the sentences of the abstracts and papers collections.

In a similar way, the results for the abstracts collection are better than the ones for the scientific papers collection. One possible reason of this is the simplicity that usually characterizes abstract sentences. Sentences in an abstract are usually easy to understand, the writer commonly shows the main ideas of what is explained below with more simple syntactic structures.

Analyzing the sentences with mistakes, we found that there are two main reasons for these errors:

- Minipar, as any other dependency parser, is not error free, being able to cover about 79% of the dependency relations. If the dependency tree returned by

Minipar is not correct, our system is not able to infer accurately the scope of negation in the sentence. In some cases we found that the tree is incomplete, and some information is missed. In these rare cases our system does not have enough information to decide how far the scope must be annotated. In most of these cases the cue is not correctly found and if it is, the scope probably closes incorrectly. As a suggestion for further work, we could replace Minipar with other dependency parsers that perform better the task of analyzing negation.

- There are some negation cues that are not always considered as negation cues, such as *negative*. This fact is evidenced in Morante’s work [16]. Due to the characteristics of our system, we must define the negation cue lexicon at the beginning. A semantic module is the obvious suggestion to tackle these special cases, because we could be able to infer the negative semantics of a single word that is not always acting as a negation signal avoiding the noise produced by an initial static decision.

4.3 Comparison with Other State-of-the-Art Systems

In this Section, we show an approximate comparison with some of the systems of the state of the art. We compare our results with the machine learning approach of Morante and Daelemans [10], the shallow semantic parsing approach of Zhu et al. [13] and the dependency system of Councill et al. [12]. The main comparison is shown in Table 3 where we show the precision, recall, F1, PCS and PCNC with other state of the art approaches. As evidenced in the results, our system performs very good for all the results with the exception of PCNC, in which the effect of our static and small lexicon of cues causes noise in the performance.

It is important to notice that in this table we show the results of Zhu’s and Morante’s systems when using automatic cue recognition, as we did in our system. Therefore, we are not reporting their results when using neither *golden cues* nor *golden trees*, which are much higher. In addition, for Councill’s system only results for the scientific papers collection are shown because it is the only collection in which they published results.

Morante’s system is based on machine-learning. In contrast, our system was constructed using as development set a subset of the sentences presented in the papers collection, as it is described in Section 3.3. Thus, while we tested our system with the Bioscope corpus (with the exception of the first 10% of the developing set of Papers), Morante et al. performed 10-fold cross validation experiments with the abstracts collection. And, for the other 2 collections, they trained with the abstracts set and tested with the corresponding collection. This fact affect the results, but we tried to make the results as comparable as possible.

This is why the Morante et al. results are much more comparable to ours in the case of the Abstracts collection. In the same way, Councill et al. results carrying out the experiment only with papers is much more directly comparable to our results because they get papers sentences to carry out the training. As shown in Table 3, we can observe how these 2 cases produced similar results to ours. Councill et al. system seems to be very competitive because the original

Table 3. Results of our work, evaluated with the three collections of Bioscope and compared with the systems of Morante et Al., Zhu et al. and Council et Al.

Collection	System	Precision	Recall	F1	PCS	PCNC
Papers	Our Results	73.49%	80.70%	76.93%	56.43%	91.15%
	Morante et Al.	72.21%	69.72%	70.94%	41.00%	92.15%
	Zhu et Al.	56.27%	58.20%	57.22%	–	–
	Council et Al.	80.80%	70.80%	75.50%	53.70%	–
Abstracts	Our Results	84.92%	84.03%	84.48%	68.92%	95.56%
	Morante et Al.	81.76%	83.45%	82.60%	66.07%	95.09%
	Zhu et Al.	78.24%	78.77%	78.50%	–	–
Clinical	Our Results	95.83%	90.58%	93.13%	89.06%	94.82%
	Morante et Al.	86.38%	82.14%	84.20%	70.75%	97.72%
	Zhu et Al.	82.22%	80.62%	81.41%	–	–

Table 4. PCS per negation cue for negation cues that occur 10 or more than 10 times in one of the subcorpus and appear in our lexicon of negation cues. The column # shows the number of appearances for each case; the column **Our** shows our system values and Morante’s system values are given in column **Mor.**

	Abstracts			Papers			Clinical		
	#	Mor.	Our	#	Mor.	Our	#	Mor.	Our
absence	57	56.14	71.93	–	–	–	–	–	–
absent	13	15.38	38.46	–	–	–	–	–	–
cannot	28	42.85	28.57	16	50.00	50.00	–	–	–
fail	57	63.15	85.97	13	38.46	53.84	–	–	–
lack	85	57.64	52.94	20	45.00	50.00	–	–	–
neither	33	51.51	72.72	–	–	–	–	–	–
no	207	73.42	81.64	44	50.00	54.54	673	73.10	89.60
none	–	–	–	10	0.00	71.42	–	–	–
not	1036	69.40	66.41	200	39.50	64.50	57	50.87	66.66
rather than	20	65.00	65.00	12	41.66	25.00	–	–	–
unable	30	40.00	73.33	–	–	–	–	–	–
without	82	89.02	79.27	24	58.33	70.83	–	–	–

task of this work was to annotate sentences to solve a sentiment analysis task, nevertheless, their results with Bioscope were very good.

In Table 4 we show the percentage of correct scopes (PCS) per negation cue, for negation cues that occur 10 or more than 10 times in each collection present in Bioscope. We compare our results with the ones published by Morante and Daelemans [10], which is the same system studied in Table 3. Negation cues with a lower PCS have a higher percentage of scopes to the left (*absent*, *unable*). In this case we consider all the test set including the data seen (in the algorithm construction) in the Papers collection, that conforms the 100.0% of the papers sentences containing negations in order to obtain the same numbers as Morante et al. system, therefore, it is worth to mention that this column of data should be observed under this perspective.

In this depicted table of results we can see how the Morante’s machine learning approach is not able to cover negation signals with a very low frequency in the

training set but a higher frequency in the test, as we can see in the results for the negation cue *none* in the papers collection (it is worth to remind that they used the abstract collection for training and carried out the testing with the papers collection, in this case). Nonetheless, our system classifies the scope of this signal with higher accuracy.

Considering the most frequent negation cues, *not* and *no* (this cues are the ones with the strongest effect on the accuracy at the end), our system beats the results of Morante et Al. in clinical reports and papers collection. However, they beat our results for the cue *not* in the abstracts collection.

Finally, we did not include the Agarwal and Yu's work [14] in the comparison, which achieves an F1-score of 98% and 95% on detecting negation cue phrases and their scope in clinical notes, and an F1-score of 97% and 85% on detecting negation cue phrases and their scope in biological literature. This approach using conditional random fields present very high results, but as discussed in the Tutorial Given at the IJCNLP 2011 conference at Chiang Mai, Thailand,⁴ the corpus partitions and the evaluation measures are different. Thus, the systems are, at least, not directly comparable, which shows that there are other ways to evaluate this task. Nevertheless, we consider important to mention this work, that includes a website in which is possible to test the system and shows a very interesting approach, as we described in Section 2.

5 Conclusions and Future Work

In this paper we presented a high performance system able to infer the scope of negations. From the results of our experiments we can conclude that dependency parsing is a valuable auxiliary technique for negation detection, at least in the particular case of English.

As a suggestion for future work, we consider that the scope of negation must not always be annotated as continuous. In Bioscope, the scope of negation leaves the subject out, with the exception of passive voice sentences. Nonetheless we consider that the subject must always be considered as a part of the scope. Moreover, when there is an affirmative sentence that affects the subject of a negative passive voice sentence, it is difficult to infer automatically which subject is considered. For instance, in the following sentence "*Therefore, TNF-alpha mRNA induction by PMA, like its induction by virus and LPS, [is not primarily mediated by NF-kappa B], but rather is mediated through other sequences and protein factors.*", the scope of negation is in passive voice but the subject is implicit by the word *is*, and it is not directly included in the scope of negation if we follow the annotation guidelines of Bioscope, as it is done in the present work. Thus, we suggest that the scope must be discontinuous in the way of considering other wordforms that in Bioscope are out of the scope, but are directly affected by the negation cue. It can be achieved using a tabular format for the corpus, instead of plain sentences annotated with XML language. As we show in the present work, we decided to evaluate our system with Bioscope, thus our system

⁴ http://www.ijcnlp2011.org/ijcnlp2011/downloads/tutorial/tu3_present.pdf

annotates the sentences in the same way as it is done in that corpus. For further work we are going to consider other approaches to evaluate our system, some of them are introduced in [1].

Observing the results shown in Table 4, where we show the PCS per negation cue, an interesting idea for future work could be a system that uses in synergy the results of both systems, our system and Morante et Al. Combining what is presented in this paper with the fact that a common approach to fault-tolerant systems is the implementation of a voting system that select the best output for two systems that perform the same computational task, in this case depending on the cue involved.

Finally, for future work, we are going to study different dependency syntactic parsers in order to find which one is the best for inferring the scope of negation. We are also thinking to test our system into the opinion mining domain, as it is done in [18], in which the effect of negation is studied for sentiment analysis in product reviewing.

Acknowledgments. This research is funded by the Spanish Ministry of Education and Science (TIN2009-14659-C03-01 Project), Universidad Complutense de Madrid and Banco Santander Central Hispano (GR58/08 Research Group Grant).

References

1. Morante, R., Schrauwen, S., Daelemans, W.: Corpus-based approaches to processing the scope of negation cues: an evaluation of the state of the art. In: Bos, J., Pulman, S. (eds.) *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, pp. 350–354 (2011)
2. Horn, L.R.: *A natural history of negation*. University of Chicago Press, Chicago (1989)
3. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G.: Evaluation of negation phrases in narrative clinical reports (2002)
4. Vincze, V., Szarvas, G., Farkas, R., Mora, G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics* 9, S9 (2008)
5. Sarafraz, F., Nenadic, G.: Identification of negated regulation events in the literature: exploring the feature space. In: *Semantic Mining in Biomedicine* (2010)
6. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G.: A simple algorithm for identifying negated findings and diseases in discharge summaries. *J. Biomed. Inform.*, 301–310 (2001)
7. Mutalik, P.G., Deshpande, A., Nadkarni, P.M.: Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study using the UMLS. *Journal of the American Medical Informatics Association* 8, 598–609 (2001)
8. Huang, Y., Lowe, H.J.: A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association* 14, 304–311 (2007)

9. Morante, R., Liekens, A., Daelemans, W.: Learning the scope of negation in biomedical texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 715–724. Association for Computational Linguistics, Stroudsburg (2008)
10. Morante, R., Daelemans, W.: A metalearning approach to processing the scope of negation. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, pp. 21–29. Association for Computational Linguistics, Stroudsburg (2009)
11. Morante, R., Sporleder, C. (eds.): Proceedings of the Workshop on Negation and Speculation in Natural Language Processing. University of Antwerp, Uppsala (2010)
12. Councill, I.G., McDonald, R., Velikovich, L.: What’s great and what’s not: learning to classify the scope of negation for improved sentiment analysis. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, NeSp-NLP 2010, pp. 51–59. Association for Computational Linguistics, Stroudsburg (2010)
13. Zhu, Q., Li, J., Wang, H., Zhou, G.: A unified framework for scope learning via simplified shallow semantic parsing. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 714–724. Association for Computational Linguistics, Stroudsburg (2010)
14. Agarwal, S., Yu, H.: Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Associations* (2010)
15. Lin, D.: Dependency-based evaluation of MINIPAR. In: Proc. Workshop on the Evaluation of Parsing Systems, Granada (1998)
16. Morante, R.: Descriptive analysis of negation cues in biomedical texts. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), European Language Resources Association (ELRA), Valletta (2010)
17. Szarvas, G., Vincze, V., Farkas, R., Csirik, J.: The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, BioNLP 2008, pp. 38–45. Association for Computational Linguistics, Stroudsburg (2008)
18. de Albornoz, J.C., Plaza, L., Gervás, P., Díaz, A.: A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 55–66. Springer, Heidelberg (2011)

LDA-Frames: An Unsupervised Approach to Generating Semantic Frames

Jiří Materna

Centre for Natural Language Processing
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
xmaterna@fi.muni.cz
<http://nlp.fi.muni.cz>

Abstract. In this paper we introduce a novel approach to identifying semantic frames from semantically unlabelled text corpora. There are many frame formalisms but most of them suffer from the problem that all frames must be created manually and the set of semantic roles must be predefined. The LDA-Frames approach, based on the Latent Dirichlet Allocation, avoids both these problems by employing statistics on a syntactically tagged corpus. The only information that must be given is a number of semantic frames and a number of semantic roles to be identified. The power of LDA-Frames is first shown on a small sample corpus and then on the British National Corpus.

Keywords: LDA-Frames, semantic frame, Latent Dirichlet Allocation.

1 Introduction

Semantic frames and valency lexicons are very useful sources of knowledge. They capture semantic roles (selectional preferences) valid for a set of lexical units (predicates). The structures of linked semantic roles are called semantic frames. Linguists are using them for their ability to describe an interface between syntax and semantics. In practical natural language processing applications, they can be used, for instance, for the word sense disambiguation or syntactic parsing. One of the largest source of semantic frames for English, based on so-called Frame Semantics, is a well-known database called FrameNet [1].

The linguistic basis of the FrameNet project and the Frame Semantics can be found in the theory of Case Grammar, beginning with the work by Charles J. Fillmore in 1968 [2]. Main goal of the FrameNet project is to extract the information about the linked semantic and syntactic properties of English words from a large electronic text corpora, using both manual and automatic procedures.

One of the most important drawbacks of FrameNet and other similar databases of semantic frames is that they are very expensive to create. The reason is that they have to be created mainly manually by trained linguists. Moreover, there is a problem with objectivity of the acquired frames since the notion of semantic classes and frames is subjectively biased when the frames are created

manually. These are strong arguments for an attempt to create a database of semantic frames automatically.

On the one hand, previous work on automatic generating of semantic frames is limited and, to our knowledge, there is no fully automated system. On the other hand, there is a lot of previous work on automatic detection of selectional preferences and semantic roles. Their identification is a good starting point for automatic creating semantic frames.

The work on selectional preferences identification can be divided into two basic categories: approaches with and without a predefined set of semantic roles. The former ones use a fixed set of roles, which are produced manually. An example of such a database of roles is WordNet [8] or semantic roles from FrameNet [4]. The roles are assigned to the predicate arguments by measuring some kind of semantic overlap. These techniques produce human-interpretable output, but often suffer from a poor coverage and quality due to an incoherent taxonomy.

In the later category, the semantic roles are usually represented by latent class variables connected with probability distributions over the set of possible argument realizations. These classes and distributions are automatically learned from data. This retains the class-based flavour of the problem, without the limitations of the manually selected sets of semantic roles. Among the first works that use such probabilistic approaches belongs the model by [10]. In their work they use the Expectation-Maximization algorithm to learn the parameters of the model. Another and more recent work is described in [9]. It uses the LinkLDA framework in which each predicate argument is modelled using the Latent Dirichlet Allocation (LDA) [1], and the inference is performed by collapsed Gibbs sampling. Similar work, also based on LDA, has been performed by [12]. In his work, he proposed a couple of LDA models for the task of modelling selectional preferences for selected grammatical relations.

The LDA-Frames method described in this paper extends such probabilistic approaches by incorporating them into the semantic frame framework. The basic idea behind the LDA-Frames is that each lexical unit (predicate) can be associated with a probability distribution over latent variables representing semantic frames. These semantic frames correspond to tuples of semantic roles (other latent variables), which are associated with probability distributions over semantic role realizations. We can simply compare probability distribution over semantic frames and group lexical units with similar meaning.

The paper is structured as follows. Section 2 is dedicated to the description of the LDA as was proposed by David Blei. In section 3, we will introduce the theory of LDA-Frames in detail. Section 4 will describe two experiments, which show the power of the LDA-Frames. The first experiment is performed on a small, artificially created set of testing data, and shows the abilities of the LDA-Frames to reconstruct hidden frames. The second one, carried out on the British National Corpus, shows the relation to the Corpus Pattern Analysis [6]. The conclusion and future work is summarized in section 5.

2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) proposed in [1] is a generative probabilistic model for collections of discrete data, such as text corpora, usually used for topic modelling. The basic idea behind the LDA is to assume that words in a text document are generated by a mixture of topics, where each topic is represented as a multinomial distribution over words. The mixing coefficients of topics for each document and the word-topic distributions are hidden and are learned from the observed data using unsupervised learning techniques. For learning the hidden parameters of LDA, Blei et al. [2] introduced a variational inference algorithm. Subsequently, Griffiths and Steyvers [5] developed an algorithm based on collapsed Gibbs sampling, which generates a sequence of samples from the joint probability distribution of the LDA model. Whereas variational inference is faster (it only needs to see the observed data once), the collapsed Gibbs sampling gives more accurate results.

More formally, LDA assumes the following generative process for each document i in a corpus of M documents:

1. Choose θ_i from $\text{Dir}(\alpha)$, where $i \in 1, \dots, M$.
2. For each word position (i, j) in the document i , where $j \in 1, \dots, N_j$:
 - (a) Choose a topic z_{ij} from $\text{Multinomial}(\theta_i)$.
 - (b) Choose a word w_{ij} from $\text{Multinomial}(\varphi_{z_{ij}})$ with a Dirichlet prior $\text{Dir}(\beta)$.

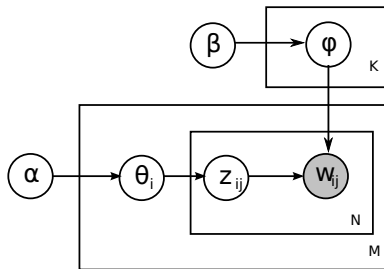


Fig. 1. Graphical model for LDA

The graphical model for LDA is shown in figure 1. In this simple model, several parameters are needed to be given. First, the dimensionality K of the Dirichlet distribution (and thus the range of the topic variables \mathbf{z}) is assumed known and fixed. Second, the topic distributions θ are parametrized by a constant vector α , and the word probability distributions φ by a constant vector β . These parameters can be set by hand or can be estimated from the data by the variational methods.

Given the observed words $\mathbf{w} = \{w_{ij}\}$, the task of the Bayesian inference is to compute the posterior distribution over the latent topics $\mathbf{z} = \{z_{ij}\}$. The posterior distribution of the model is given by

$$P(\mathbf{w}, \mathbf{z}, \theta, \varphi | \alpha, \beta) = \prod_{k=1}^K \text{Dir}(\varphi_k | \beta) \prod_{i=1}^M \text{Dir}(\theta_i | \alpha) \prod_{j=1}^N \text{Mult}(z_{ij} | \theta) \text{Mult}(w_{i,j} | \varphi_{z_{ij}}) \quad (1)$$

In the collapsed Gibbs sampling inference method, the θ and φ distributions are marginalized out, and only the latent topics $\mathbf{z} \in \{1, \dots, K\}$ are sampled. After having the topic variables inferred for all words in the corpus, the topic-document distribution θ and the word-topic distribution φ can be easily computed.

3 LDA-Frames

The LDA-Frames are semantic frames automatically generated from a syntactically tagged corpus, motivated by so called Word Sketches [7] and the Frame Semantics [3]. Word sketches are automatic, corpus-based summaries of a word's grammatical and collocational behaviour, which takes as input a corpus of any language and corresponding grammar patterns. The resulting summaries are produced in the form of a ranked list of common word realizations for a grammatical relation of a given target word. The LDA frames combine the Word Sketch based grammar patterns with the idea of semantic frames using the LDA method.

In the LDA-Frames, a frame is represented as a tuple of semantic roles, each of them connected with a grammatical relation (subject, object, modifier, etc.). These frames are connected with a predicate (in the Frame Semantic called lexical unit) via a probability distribution. For each lexical unit, we are given a set of corpus realizations of selected grammatical relations acquired from the Word Sketch grammar patterns, and our goal is to identify generalizations of such realizations in the form of LDA frames.

Let us look at an example of grammatical relation realizations of lexical units *eat*, *drink* and *teach* in table 1. In this example we use two grammatical relations – subject and object, whose corpus realizations are shown in columns two and three. The fourth column shows the frames we would like to identify automatically. In the LDA-Frames formalism, the set of inferred frames and semantic roles is fixed and shared through all lexical units. Here, the set of semantic roles is given by $R = \{Person, Food, Animal, Drink\}$ and the set of frames by $F = \{(Person, Food), (Animal, Food), (Person, Drink), (Person, Person), (Person, Animal)\}$. We can see that the semantic roles have human-interpretable labels. In the fact, the LDA-Frames framework is not able to identify such labels. It uses integers instead of them. If the user wants to replace these numbers by labels, one has to assign them manually based on word distributions. Nevertheless, the labels are not needed in many practical NLP application, and moreover, thanks to this feature, the only input the LDA-Frames formalism requires is the set of corpus realizations, number of frames and number of semantic roles.

Table 1. Example of grammatical relation realizations

Lexical unit	subject	object	frame
eat	John	food	(Person, Food)
	Mike	pizza	
	man	cake	
	dog	meat	(Animal, Food)
	mouse	cheese	
drink	Jane	coffee	(Person, Drink)
	Mike	tee	
teach	teacher	student	(Person, Person)
	professor	Mike	
	Peter	dog	(Person, Animal)

3.1 Generative Process

The method of automatic identification of semantic frames is based on the probabilistic generative process. We treat each grammatical relation realization as generated from a given semantic frame according to the word distribution of the corresponding semantic role. Formally, let $G = \{g_1, g_2, \dots, g_S\}$ be the list of selected grammatical relations, $LU = \{lu_1, lu_2, \dots, lu_U\}$ list of lexical units, and

$$\mathbf{w} = \left\{ \begin{aligned} &\{(w_{1,1,1}, w_{1,1,2}, \dots, w_{1,1,S}), \dots, (w_{1,T,1}, w_{1,T,2}, \dots, w_{1,T,S})\}, \\ &\{(w_{2,1,1}, w_{2,1,2}, \dots, w_{2,1,S}), \dots, (w_{2,T,1}, w_{2,T,2}, \dots, w_{2,T,S})\}, \\ &\dots \\ &\{(w_{U,1,1}, w_{U,1,2}, \dots, w_{U,1,S}), \dots, (w_{U,T,1}, w_{U,T,2}, \dots, w_{U,T,S})\} \end{aligned} \right\}$$

grammatical relation realizations of all lexical units. For simplicity, we assume that each lexical unit has T realizations in the corpus. This simplification can be easily avoided by indexing T_1, T_2, \dots, T_U . The number of frames is given by the parameter F and the number of semantic roles by R . The realizations are generated by the LDA-Frames as follows.

For each lexical unit $u \in \{1, 2, \dots, U\}$:

1. Choose a frame distribution φ_u from $\text{Dir}(\alpha)$.
2. For each lexical unit realization $t \in \{1, 2, \dots, T\}$ choose a frame f_{ut} from $\text{Multinomial}(\varphi_u)$, where $f_{ut} \in \{1, 2, \dots, F\}$:
3. For each slot $s \in \{1, 2, \dots, S\}$ of the frame f_{ut}
 - (a) look up the corresponding semantic role r_{uts} from $\rho_{f_{ut}s}$, where $r_{uts} \in \{1, 2, \dots, R\}$.
 - (b) generate a grammatical realization w_{uts} from $\text{Multinomial}(\theta_{r_{uts}})$

The graphical model for LDA-Frames is shown in figure 2. In this model, ρ_{fs} is a projection $(f, s) \mapsto r$, which assigns a semantic role to each slot s of a frame f . This projection is global for all lexical units. The multinomial distribution of

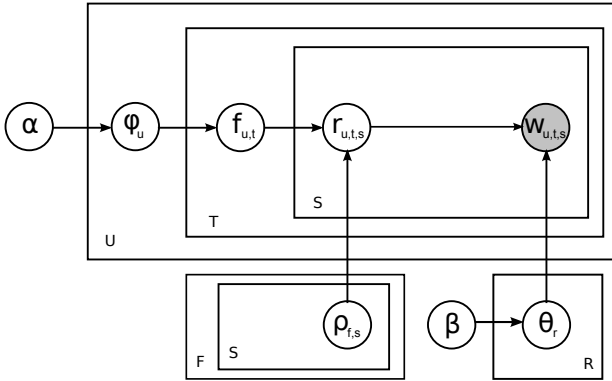


Fig. 2. Graphical model for LDA-Frames

words θ_r for a semantic role r is from Dirichlet(β). The model is parametrized by F (number of frames), R (number of semantic roles), and by hyperparameters of prior distributions α and β . These hyperparameters are usually set by hand to a value between 0.01 – 0.1.

Given the observed grammatical relation realizations \mathbf{w} , the task of the Bayesian inference is to compute the posterior distribution over the latent variables $\mathbf{f} = \{f_{ut}\}$ and $\mathbf{r} = \{r_{uts}\}$. The posterior distribution of the model is given by

$$\begin{aligned}
 P(\mathbf{w}, \mathbf{f}, \mathbf{r}, \varphi, \theta, \rho | \alpha, \beta) = & \\
 & \prod_{i=1}^R \text{Dir}(\beta) \prod_{i=1}^F \prod_{j=1}^S \text{Categorical}\left(\frac{1}{R}\right) \times \\
 & \prod_{u=1}^U \text{Dir}(\alpha) \prod_{t=1}^T \text{Mult}(\varphi_u | \alpha) \prod_{s=1}^S \text{Mult}(r_{uts} | \rho_{f_{uts}}) \text{Mult}(w_{uts} | \theta_{r_{uts}}) \quad (2)
 \end{aligned}$$

For the inference we use collapsed Gibbs sampling, where the θ , ρ and φ distribution are marginalized out. After having all topic variables \mathbf{f} and \mathbf{r} inferred, we can proceed to computing the lexical unit–frame distribution and the semantic role–word distribution. Let C_{uf}^φ be the count of cases where frame f is assigned to lexical unit u , C_{rw}^θ is the count of cases where word w is assigned to semantic role r and V is the size of vocabulary. The φ and θ distributions are computed using the following formulas:

$$\varphi_u = \frac{C_{uf}^\varphi + \alpha}{\sum_f C_{uf}^\varphi + F\alpha} \quad (3)$$

$$\theta_r = \frac{C_{rw}^\theta + \beta}{\sum_w C_{rw}^\theta + V\beta} \quad (4)$$

3.2 Semantically Related Lexical Units

The semantic frames generated by the LDA-Frames formalism are an interesting source of information about selectional preferences, but they can even be used for grouping semantically related lexical units. Separated semantic frames can hardly capture the whole semantic information about a lexical unit. Nevertheless, the LDA-Frames provide an information about the relatedness to every semantic frame we have inferred. After the inference process, each lexical unit u is connected with a probability distribution over semantic frames φ_u . Thus, we can group lexical units with similar probability distributions together to make a semantic cluster.

There are several methods how to compare probability distributions. We use the Hellinger Distance, which measures the divergence of two probability distributions, and is a symmetric modification of the Kullback-Leibner divergence. For two probability distributions φ_a, φ_b , where $P(f|x)$ is the probability of frame f being generated by lexical unit x , the Hellinger Distance is defined as follows:

$$H(a, b) = \sqrt{\frac{1}{2} \sum_{f=1}^F \left(\sqrt{P(f|a)} - \sqrt{P(f|b)} \right)^2} \quad (5)$$

4 Experiments

We have performed two experiments to assess the quality of generated LDA frames. The first task shows how well is the algorithm able to reconstruct frames that were used for generating data, and are hidden during the inference process. This test is performed on a small, artificially created set of frames. In the second task, we have employed the LDA-Frames algorithm on the British National Corpus, and compared the resulting frames with frame patterns built in the Corpus Pattern Analysis project [6].

4.1 Reconstructing Abilities of LDA-Frames

The LDA-Frames algorithm requires several parameters and hyperparameters to be given. While the number of frames and roles should be determined by the application, the hyperparameters α and β , and the number of iteration are unknown. In the experiment described in this section, we will show how to set these unknown variables to achieve the best quality of generated frames on an artificially created data.

In the test, we have created a set of four semantic roles, each of them with several lexical realizations. The set of roles is given by table 2. Notice, that some of the realizations (for instance *fish* or *jenny*) are assigned to more than one semantic role. This makes the semantic roles more realistic and the realization–role assignment ambiguous.

Next, we have selected eleven lexical units, and for each of them several semantic frames, corresponding to *subject* and *object* grammatical relations. These

Table 2. Semantic roles and their lexical realizations

Semantic roles	realizations
Animal	dog, cat, mouse, fish, chicken, jenny
Food	cake, fish, lunch, chicken, dinner
Institution	school, state, university, police
Person	people, man, woman, john, jenny

Table 3. Lexical units and their semantic frames

Lexical unit	Semantic frames
boil	(Person, Food)
buy	(Person, Food), (Person, Animal)
cook	(Person, Food), (Institution, Food)
eat	(Person, Food), (Animal, Food)
love	(Person, Person)
pay	(Institution, Person)
sell	(Institution, Food)
sue	(Institution, Institution), (Person, Institution)
teach	(Person, Person), (Institution, Person)
warn	(Person, Person), (Institution, Person)
work for	(Person, Institution), (Person, Person)

frames are only composed of the roles defined in table 2, and therefore have no ambition to be complete (in the sense that there are no more frames for a lexical unit). The selected lexical units with corresponding frames are showed in table 3.

Using the semantic roles and frames from tables 2, 3, we have generated a dataset of frame realizations according to the following algorithm. For every lexical unit u :

1. Choose a number of corpus realizations $N_u \in \{10, \dots, 100\}$ from the uniform distribution.
2. For each realization $n_u \in \{1, \dots, N_u\}$, among all permitted frames for lexical unit u , choose a semantic frame f_{n_u} from the uniform distribution.
3. For each frame f_{n_u} generate a realization of all its roles according to table 2 from the uniform distribution.

As the result of the algorithm we was given between 10 and 100 tuples of frame realizations for each lexical unit from the table 3. This dataset takes the same form as the input data for the LDA-Frames algorithm. It is known which frame is responsible for generating a particular realization, but this information is not presented in the generated data.

In order to evaluate reconstructing abilities of the LDA-Frames, we employed the LDA-Frames algorithm on the generated data to infer new frames and frame assignments from unlabelled data, and compared these frames with the original ones. Since the inferred semantic roles are represented by integers, the map-

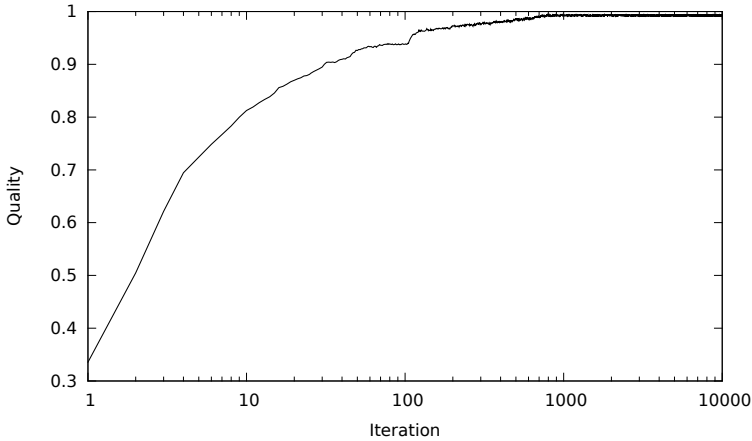


Fig. 3. Graph of the quality

ping onto the original semantic roles is ambiguous (in this case, there are $4!$ combinations). In order to disambiguate the assignment, we compared word distributions of the original semantic roles with the generated ones, and used the most likely mapping. Specifically, the similarity has been measured by the Hellinger distance as was defined in section 3.2.

The quality of the generated semantic frames has been measured as the percentage of the correctly assigned frames for all frame realization in the data. A graph of the quality for the number of sampling iteration between 1 and 10^4 is shown in figure 3 (the horizontal axis uses a logarithmic scale). The experiment has been performed 50 times with different initial values of hidden variables and the qualities for each iteration have been averaged. It turns out that the quality reaches its maximum after approximately 1000 iteration. The reached solution was the best one, which correctly recovered all the hidden variables. We tried to set the hyperparameters α and β to various values between 0.01 and 0.5 but, for the given data, the differences have not been significant.

4.2 Comparison with CPA

The previous experiment showed how good the LDA-Frames algorithm performs on small artificial data. In this experiment, the power of LDA-Frames has been shown on the British National Corpus data. The generated frames were then compared with the Corpus Pattern Analysis (CPA) frames [6]. The frames in CPA are very similar in structure to the frames generated by LDA-Frames, moreover, the CPA frames are created and linked to corpus uses by linguists. This is the reason why we compared LDA-Frames with CPA in order to evaluate the quality of LDA-Frames.

First, we have selected all verbs from CPA and found all their realizations in the British National Corpus. For each realization we also identified the *subject*

Table 4. Semantic frames for lexical unit *consume*

Semantic frame	(37, 52) 0.82308		(37, 58) 0.09153		(130, 52) 0.05374	
realizations	subject	object	subject	object	subject	object
	people	food	people	time	dog	food
	child	meal	child	energy	fish	meal
	man	meat	man	night	animal	meat
	woman	fruit	woman	year	snake	fruit
	family	fish	family	minute	cat	fish
	girl	dinner	girl	month	cow	dinner
	person	breakfast	person	thing	sheep	breakfast
	mother	lunch	mother	resource	predator	lunch
	baby	bread	baby	way	rabbit	bread
	boy	sandwich	boy	lot	horse	sandwich

Table 5. The most similar lexical units to *consume*

Lexical unit	drink	eat	ingest	gulp	smoke	sip	devour	slurp
Distance	0.619	0.622	0.658	0.661	0.666	0.681	0.691	0.691

and *object* grammatical relation realizations using the Word Sketch grammar patterns. This data has been used to generate semantic frames using the LDA-Frames in 1000 sampling iterations. The parameters have been set in accordance with CPA frames to $F = 817$, $R = 200$. The hyperparameters have been set to $\alpha = \beta = 0.01$. An example of resulting frames, specifically for the lexical unit *consume*, is shown in table 4.

In the table, there are three most frequent frames sorted by probability in descending order from left to right. In the first row of the table, there are semantic frames represented as tuples of integers, where each number corresponds to one of 200 semantic roles, along with their probabilities. Below each frame, there is a list of first ten role realizations sorted by their probability in descending order from top to down.

As was mentioned in section 3.2, we can group lexical units with similar frames distributions. In table 5, there are listed the most similar lexical units to the verb *consume* along with their distance scores.

In the CPA project, the lexical unit–frame assignment is annotated in the corpus, but there is no information about the semantic roles realizations. Thus, we have not been able to perform the same test as in the previous experiment. In order to test the quality of the generated frames, we have compared the generated and CPA frames for 300 most frequent lexical units from CPA. For the experiment, it has been necessary to map the integer representations of semantic roles in LDA-Frames to semantic roles in CPA. This mapping has been carried out manually so that at least one CPA role had to be assigned to each semantic role in LDA-Frames. Since the roles in LDA-Frames are flat (they have

no hierarchical structure), some of them had to be assigned to more than one semantic role in CPA.

The quality has been measured as follows. First, for each lexical unit we have selected only those frames that have corpus coverage greater than 10 %. A lexical unit has 1.48 such CPA frames on average. The same selection has been carried out on the LDA-Frames. The average number of such frames for the LDA-Frames is 1.71. The quality has been measured by two different methods. First, as the percentage of the LDA-Frames that are presented in the set of CPA frames for each lexical unit, and second as the percentage of lexical units, where the most frequent frame is same in CPA and LDA-Frames. The results are showed in table 6.

Table 6. Quality of LDA-Frames in comparison with CPA frames

Method	Quality
All frames	52.8 %
Most frequent frames	61.8 %

The results show that although the LDA-Frames are created by a different method, they result in similar frames as in CPA. The greatest advantage of the LDA-Frames is that they do not need any human annotators, and are theoretically language independent on the assumption that we have a Word Sketch grammar patterns for that language.

5 Conclusions

We have presented a method for automatic identification of semantic frames using topic modelling similar to the Latent Dirichlet Allocation. This method is called LDA-Frames. It computes statistics on a syntactically tagged corpus using the Word Sketch grammar patterns, and provides a set of semantic frames along with the probability distributions over frames for lexical units and probability distribution over corpus realizations for the set of semantic roles. The only information that must be given is a number of semantic frames and a number of semantic roles to be identified. Using the probability distribution over frames, it is possible to easily group lexical units with similar meanings together.

The quality of generated frames have been measured by two experiments. In the first one, we have measured its abilities to reconstruct frames that were used for generating data, and were hidden during the inference process. In the second task, we have employed the LDA-Frames algorithm on the British National Corpus, and compared the resulting frames with frames built in the Corpus Pattern Analysis project. Both the tests showed that the algorithm works really well. In the experiments, we have used only *subject* and *object* grammatical relations, but the method is general and can be used with any other relations.

Since the LDA-Frames provide a complete probabilistic model for the given data, its results are applicable to many NLP tasks such as word sense disambiguation, information extraction, or disambiguation for syntactic parsing without coverage limitations.

Acknowledgements. This work has been partly supported by the Ministry of Education of Czech Republic under the projects LC526 and LINDAT-Clarin LM2010013.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3, 993–1022 (2003)
2. Fillmore, C.J.: *The Case for Case*. In: *Universals in Linguistic Theory*. Holt, Rinehart and Winston, New York (1968)
3. Fillmore, C.J.: *Frame Semantics*. In: *Linguistics in the Morning Calm*, pp. 111–137. Hanshin Publishing Co., Seoul (1982)
4. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. *Computational Linguistic* 28, 245–288 (2002)
5. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. *Proceedings of the National Academy of Sciences of the United States of America*, 5228–5235 (2004)
6. Hanks, P., Pustejovsky, J.: *A Pattern Dictionary for Natural Language Processing*. In: *Revue Francaise de Langue Appliquée*. Brandeis University (2005)
7. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. In: *Proceedings of the Eleventh EURALEX International Congress*, Lorient, France, pp. 205–116 (2004)
8. Resnik, P.: Selectional Constraints: an Information-Theoretic Model and Its Computational Realization. *Cognition* 61, 127–159 (1996)
9. Ritter, A., Mausam, Etzioni, O.: A Latent Dirichlet Allocation Method for Selectional Preferences. In: *Proceedings of the 48th Annual Meeting of the ACL*, pp. 424–434. Association for Computational Linguistics (2010)
10. Rooth, M., Riezler, S., Prescher, D., Carroll, G., Beil, F.: Inducing a Semantically Annotated Lexicon via EM-based Clustering. In: *Proceedings of the 37th Annual Meeting of the ACL*, pp. 104–111. Association for Computational Linguistics (1999)
11. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J.: *FrameNet II: Extended Theory and Practice* (2006), <http://www.icsi.berkeley.edu/framenet>
12. Séaghdha, D.Ó.: Latent Variable Models of Selectional Preference. In: *Proceedings of the 48th Annual Meeting of the ACL*, pp. 435–444. Association for Computational Linguistics (2010)

Unsupervised Acquisition of Axioms to Paraphrase Noun Compounds and Genitives

Anselmo Peñas¹ and Ekaterina Ovchinnikova²

¹ NLP & IR Group, UNED
Juan del Rosal, 16
28040 Madrid, Spain
anselmo@lsi.uned.es

² USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292 USA
katya@isi.edu

Abstract. A predicate is usually omitted from text when it is highly predictable from the context. This omission is due to the effort optimization that humans perform during the language generation process. Authors omit the information that they know the addressee is able to recover effortlessly. Most noun-noun structures including genitives and compounds are result of this process. The goal of this work is to generate automatically and without supervision the paraphrases that make explicit the omitted predicate in these noun-noun structures. The method is general enough to address also the cases where components are Named Entities. The resulting paraphrasing axioms are necessary for recovering the semantics of a text, and therefore, useful for applications such as Question Answering.

Keywords: Paraphrasing, Background Knowledge Acquisition, Noun compounds, Proposition Stores.

1 Introduction

Humans optimize the effort of language generation and they omit the pieces of information that they know addressees are able to recover. In particular, when a predicate is highly predictable from the context, it can be omitted from the discourse (see [1] for a detailed study of implicit predicates). From a computational perspective, the recovering of the omitted information is crucial if we want to build a semantic representation of the text.

Some transparent noun compounds (e.g., “*morning coffee*”) and genitives (e.g., “*Shakespeare’s tragedy*”) are result of this process. Their interpretation requires making explicit the relation between the nouns. For example, the noun compound “*morning coffee*” is most probably interpreted as a “*coffee drunk in the morning*”, while a possible interpretation of “*morning newspaper*” is a “*paper read in the morning*”. Analogously, the genitive “*Shakespeare’s tragedy*” is most probably interpreted as a “*tragedy written by Shakespeare*”.

Experimental studies show that noun-noun constructions are very common and the compounding process is extremely productive in English [2]. Interpretation of noun-noun dependencies is important for many NLP applications, especially those like Question Answering or Textual Entailment that require the consideration of paraphrases.

The ultimate goal of this work is to automatize the process of making explicit the omitted predicate in noun-noun structures. Our approach relies on the use of Proposition Stores in the style of KNEXT [3], DART [4], and BKB [5]. Proposition Stores contain lexico-syntactic structures with their frequency in a large corpus. For example, “*quarterback throw(s) pass*” is a proposition of type NVN (subject-verb-object); “*bomb explode(s) in attack*” is a proposition of type NVPN (subject-verb-preposition-complement), etc. Thus, for example, they are candidate paraphrases to make more explicit the meaning of compounds such “*bomb attack*” or “*Favre pass*” (if we gather previously that “*Favre is a quarterback*”).

The latter example shows the need of selecting the most relevant semantic classes for a given Named Entity (NE). This is necessary to enable access to the most relevant portions of background knowledge (propositions that paraphrase noun-noun structures in our particular case). For example, given the excerpt “*Dan Marino’s pass*”, if the most relevant class for *Marino* is *player* then we expect *pass* to have the meaning of *passing play*. However, if the most relevant class for *Marino* is *person* rather than *player*, then we must consider *pass* to have the meaning of *passport*. Different portions of background knowledge are enabled in each case. Thus, for the former case a relevant paraphrase would be “*Dan Marino throw(s) pass*” and for the latter a relevant paraphrase would be “*Dan Marino show(s) pass*”.

Usual approaches to Named Entity recognition work with predefined sets of entity classes and turn the problem into a classification one. In that approach, the finer the grain of the classes is, the harder is its correct identification, and the bigger is the amount of training data required. Thus, for example, if we fall into American football domain we can expect that the predefined class for entities such as “*Dan Marino*” is “*player*”. However, this decision is being made arbitrarily. Is “*player*” the appropriate level of abstraction able to retrieve the most relevant portions of background knowledge in the domain? If we ask a background knowledge base what a “*player*” does with a “*pass*”, then the category “*player*” is still too coarse to recover useful background knowledge: they can *throw*, they can *catch*, they can *drop* the ball, etc. If we were able to categorize “*Dan Marino*” as “*quarterback*”, then we would recover different but more accurate actions: the fact that “*quarterbacks*” usually *throw* or *complete passes* more than *catch* or *drop* them.

The good news is that in most cases, the relevant semantic classes we need to consider are pointed out by the text through very well-known structures (such as appositions, copulative verbs, preposition “*as*”, etc.). We can consider these structures to work with an unbounded number of NE semantic classes. One of the main contributions of this work is to leverage the information about semantic classes to provide better paraphrasing predicates for a given noun compound, in particular when one of the components is a named entity no matter if previously unsighted.

Summing up, we present a completely unsupervised method to paraphrase noun-noun structures using different types of *propositions*. The method is general enough to consider not only noun compounds but also genitives, not only common nouns but also proper nouns, and not only recover subject-verb-object predicate structures but also consider other structures including, for example, prepositional attachments.

2 Previous Work

There are two main conceptions for the task of making the meaning of noun-noun dependencies more explicit: The first one assumes that relations between the component nouns can be mapped to a finite set of predefined semantic relations [6]. The second one assumes there are an unbound number of possible relations between component nouns [7] [8].

There are several approaches to state a predefined set of semantic relations that can make more explicit the semantics of a noun compound. In some cases, the relations are made explicit by actual words in language. In the case of [6], the resulting finite list contain predicates (verbs and prepositions) such as CAUSE, HAVE, MAKE, USE, BE, IN, FOR, FROM, ABOUT. Lauer [9] opted for the use of prepositions like OF, FOR, IN, ON, AT, FROM, WITH, ABOUT.

In the context of knowledge based systems, the target ontology is giving the finite set of possible interpretations to map the noun compounds [10] [11] [12].

The third option is to predefine a finite set of abstract semantic relations such as POSSESSION, LOCATION, PURPOSE, etc. There are several proposals in this direction [13] [14] [15] [16] including SemEval-2 Task 8 [17]. This approach has the drawback of having to select a priori the finite set of relations. Different authors propose different sets of relations and, ultimately, this decision has to be taken in consideration of what machine classifiers are able to discriminate.

Related to the principle that there is an unbound set of possible relations between nouns, the option that is taking more strength is to approximate their meaning by using paraphrases [18]. Following their example, the meaning of “*malaria mosquito*” could be approximated (among many other options) by the clause “*mosquito that carries malaria*”. This can be seen as the retrieval of the predicate that is implicit in the compound. For example, [18] use the pattern *N1 THAT * N2* to search in the web the verbs that can express the meaning of the compound *N1 N2*. Thus, the result is a resource of candidate predicates for each noun compound.

In this tradition, most of the approaches have been evaluated towards pairs of nouns isolated from the context. In a common scenario, the system under evaluation is supposed to generate a set of paraphrases for each two nouns and score the paraphrases, so that the scores correlate well with human judgments. For example, participant systems in SemEval-2 Task 9 [19] had to assess the relevance of the proposed verbs in order to make explicit the compound meaning.

We propose in this work a unified method to paraphrase noun compounds and genitives including those containing Named Entities. Our method to acquire paraphrasing axioms is completely unsupervised and the set of expected semantic classes for the proposition arguments is completely unrestricted.

3 Unsupervised Acquisition of Paraphrasing Axioms

We work with the hypothesis that the meaning of noun-noun dependencies can be made more explicit by means of predicates related to certain lexico-syntactic structures (propositions). This procedure depends on whether the components are common nouns or instances (proper nouns or Named Entities). We distinguish the following cases:

1. Common-noun as head of a common-noun
 - a. $nn(\text{common-noun}, \text{common-noun})$, e.g. *bomb attack*
 - b. $poss(\text{common-noun}, \text{common-noun})$, e.g. *officer's response*
 - c. $of(\text{common-noun}, \text{common-noun})$, e.g. *door of a house*
2. Proper-noun as head of a proper noun
 - a. $nn(\text{proper-noun}, \text{proper-noun})$, e.g. *Hong Kong Disneyland*
 - b. $poss(\text{proper-noun}, \text{proper-noun})$, e.g. *Vikings' Culpepper*
 - c. $of(\text{proper-noun}, \text{proper-noun})$, e.g. *England of the Tudors*
3. Common-noun as head of a proper-noun
 - a. $nn(\text{proper-noun}, \text{common-noun})$, e.g. *England fear*
 - b. $poss(\text{proper-noun}, \text{common-noun})$, e.g. *John's book*
 - c. $of(\text{common-noun}, \text{proper-noun})$, e.g. *house of John*
4. Proper-noun as head of a common noun
 - a. $nn(\text{common-noun}, \text{proper-noun})$, e.g. *quarterback Culpepper*
 - b. $poss(\text{common-noun}, \text{proper-noun})$, e.g. *people's England*
 - c. $of(\text{proper-noun}, \text{common-noun})$, e.g. *England of aristocrats*

In the case (1) we can use as candidates the propositions found directly in the collection that have both nouns among its arguments. In cases (2), (3) and (4) a proper noun is involved in the dependency and we need additional processing. Ideally, we would need to consider a set of possible semantic classes for each proper noun, and then find the propositions that paraphrase the noun-noun structure according to these classes. This is particularly true for case (2) where, for example, there is no clue of the appropriate paraphrasing of “*Vikings' Culpepper*” unless we know that “*Vikings*” is a “*team*” or “*Culpepper*” is a “*player*” or both. Knowing this, we can find the paraphrase “*Culpepper plays for Vikings*”. Notice that we don't really need a set of classes for the two proper nouns. One could be enough to constrain the retrieval of candidate propositions. This is, in fact, what occurs in case (3). In case (4) only subcase (4a) has some frequency in the source text collections. In this case (4a), the compound is using the common noun as an attribute or a class of the proper noun. We state in these cases that the implicit predicate is *to be* (e.g. *Culpepper is quarterback*). Therefore, we don't search for other candidate paraphrases in this case.

3.1 System Overview

The input to the system proposed is a list of arguments a , and the task is to find the predicates p with highest probability $P(p|a)$. Propositions are tuples $\langle p, a \rangle$ that we

extract previously from large document collections. Some of the arguments in a can be instances (Named Entities), and thus we can't expect that we have observed p with a before. We will bridge this gap by considering the possible classes c for the instances in a . For this purpose, we need to estimate the probabilities $P(c|a)$ and $P(p|c)$. Summing up, in first place we build the Proposition Store; second, we estimate $P(c|a)$ and $P(p|c)$; then we combine them to calculate $P(p|a)$; and finally, we return the predicates p ranked by probability.

3.2 Building the Proposition Store

The Proposition Store contains *class-instance* relations in one hand, and propositions in the other. We obtain the *class-instance* relations in a very easy way using three general patterns over syntactic dependency trees:

1. A dependency where a common noun (candidate class) is a modifier of the instance (e.g. *quarterback Brett Favre*).
2. An apposition dependency between a noun and an instance in any direction (e.g. *Brett Favre, quarterback, ...*).
3. A dependency between a noun and an instance due to a copulative verb in any direction (e.g. *Brett Favre is the quarterback...*).

Finding all possible *class-instance* relations is not the goal of these patterns, but to recover enough evidence for further probability estimation over classes and propositions. As a matter of example, these are the most frequent *class-instance* relations in the experiment collection, sorted by frequency:

```
has_instance(leader, 'Yasir': 'Arafat'):1491.
has_instance(amod:[leader:n, palestinian:a], 'Yasir': 'Arafat'):1187.
has_instance(spokesman, 'Marlin': 'Fitzwater'):1001.
has_instance(leader, 'Mikhail': 'S.': 'Gorbachev'):980.
has_instance(amod:[leader:n,
soviet:a], 'Mikhail': 'S.': 'Gorbachev'):901.
has_instance(chairman, 'Yasir': 'Arafat'):756.
has_instance(agency, 'Tass'):637.
has_instance(leader, 'Radovan': 'Karadzic'):611.
has_instance(adviser, 'Condoleezza': 'Rice'):590.
has_instance(nn:[adviser:n, security:n], 'Condoleezza': 'Rice'):569.
...
```

The main advantage of this approach is that we are not predefining the set of possible classes, but letting the texts to point them out. As discussed in the introduction, each entity may require different semantic labels in different contexts to enable the interpretation, so we permit more than one semantic class per instance.

With respect to propositions, they are extracted as lexical instantiations of a predefined set of syntactic structures (see [20] for more details). The most productive propositions for noun compound paraphrasing are of type NVN (lexical instantiations of subject-verb-object structure), type NVPN (lexical instantiations of subject-verb-preposition-complement structure), and type NPN (lexical instantiations of noun-preposition-noun structure). Our Proposition Store supports queries about any proposition part and any proposition type. For example, the query “*propositions that include bomb as noun and attack as noun, sorted by frequency*” produces:

```

nvn:[bomb:n, in:in, attack:n]:13.
nvpn:[bomb:n, explode:v, in:in, attack:n]:11.
nvnvn:[bomb:n, kill:v, people:n, in:in, attack:n]:8.
nvn:[attack:n, with:in, bomb:n]:8.

```

...

Some propositions contain instances and we will use them to infer the association between predicates and classes. For example, from the *class-instance* relation we know that *Yasir Arafat* is a *leader*. We will infer the things a leader can do from the things *Yasir Arafat* (and other leader) do:

```

nvn:[renounce:v]:['NAME', terrorism:n]:['Yasir':'Arafat']:9.
nvn:[condemn:v]:['NAME', bombing:n]:['Yasir':'Arafat']:7.
nvn:[tell:v]:['NAME', reporter:n]:['Yasir':'Arafat']:6.

```

...

3.3 Probability of a Predicate Given the Classes of Its Arguments

The estimation of $P(p|c)$ has two sources of evidence: propositions previously observed that take the list of classes $c = \langle c_1, \dots, c_n \rangle$ as arguments, and propositions that take as arguments instances that belong to a class in c . We use the first source of evidence for the queries that involve only classes, and the second source for the queries that involve at least one instance. In the first case, we can estimate $P(p|c)$ directly as:

$$P(p|c) = \frac{|p \text{ has_args } c|}{|* \text{ has_args } c|}$$

In the second case, we consider only the propositions with at least one instance in the following way: Let p be a predicate with n arguments and we have observed it instantiated with different n -tuples of instances $i = \langle i_1, \dots, i_n \rangle$. Now, we look for the most probable n -tuples of classes $c = \langle c_1, \dots, c_n \rangle$ as arguments of p considering the observed n -tuples of instances i and their possible classes c given by the *class-instance* relation. The assumptions we make are:

1. Each individual instance i_k in the tuple i has a set of possible classes independently of the other instances in i .

$$P(i|c) = \prod_{k=1}^n P(i_k|c_k)$$

2. Arguments of the predicate restrict each other, that is, arguments i are independently conditioned by p .

$$P(p|i) = \frac{|p \text{ has_args } i|}{|* \text{ has_args } i|}$$

Thus, we formulate the general problem in terms of probability as follows:

$$P(p|c) = \sum_i P(i|c) \cdot P(p|i)$$

This model considers tuples of arguments instead of arguments independently. In this way, we capture the interdependence between arguments.

In most cases, at least one argument is not an instance (proper noun), but a class (common noun). For this reason we distinguish three cases for the probability of an instance given a class:

$$P(i_k|c_k) = \begin{cases} 1 & \text{if } i_k = c_k \\ \frac{|c_k \text{ has_instance } i_k|}{|c_k \text{ has_instance } *|} & \text{if } |c_k \text{ has_instance } i_k| > 0 \\ \frac{|* \text{ has_instance } i_k|}{|* \text{ has_instance } *|} & \text{if } i_k \text{ is instance, } |c_k \text{ has_instance } i_k| = 0 \\ 0 & \text{otherwise} \end{cases}$$

The first case corresponds to the arguments that are classes. The second and third cases correspond to instances observed before. The fourth case, again, corresponds to an argument that is neither an instance nor the given class. The development of this case by estimating a relatedness probability between classes is left for future work.

3.4 Probability of a Predicate Given a List of Arguments

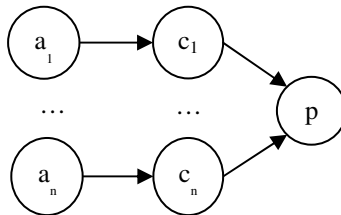
Noun compound paraphrasing axioms have the generic form: $a \rightarrow p \mid P(p|a)$, where a is the list of arguments for predicate p and $P(p|a)$ is the conditioned probability for the axiom. For the estimation of $P(p|a)$ we make the same assumptions we made in the previous step:

1. Each argument a_k in the tuple a has a set of possible classes independently of the other arguments.

$$P(c|a) = \prod_{k=1}^n P(c_k|a_k)$$

2. Arguments of the predicate restrict each other. In this case, we will assume that the arguments classes are independently conditioned by p . This probability $P(p|c)$ has been estimated in the previous step.

Graphically, we are modeling the problem as:



$$P(p|a) = \sum_c P(c|a) \cdot P(p|c)$$

For the probability of a particular class given the input argument we distinguish these cases:

$$P(c_k|a_k) = \begin{cases} 1 & \text{if } c_k = a_k \\ \frac{|c_k \text{ has_instance } a_k|}{|* \text{ has_instance } a_k|} & \text{if } |c_k \text{ has_instance } a_k| > 0 \\ \frac{|c_k \text{ has_instance } *|}{|* \text{ has_instance } *|} & \text{if } a_k \text{ is instance, } |c_k \text{ has_instance } a_k| = 0 \\ 0 & \text{otherwise} \end{cases}$$

Again, the first case corresponds to the situation where the argument is a class itself. The second case corresponds to instances (proper nouns) seen before in the class-instance relations after processing the source collection. The third case corresponds to instances not seen before. The fourth corresponds to the cases where a_k is a common noun expressing a class different than c_k . We leave for future work the estimation of a relatedness probability to be used in this case.

4 Evaluation

The evaluation aims at measuring the implicit predicates that could be recovered with the fully automatized procedure we have presented so far. For this purpose, we have checked the data sets of the second Recognizing Textual Entailment (RTE-2) looking for the textual entailment pairs containing noun-noun dependencies that are relevant for inferring entailment.

For the purposes of our experimentation, we built a Proposition Store from a collection of 216,303 New York Times articles categorized as World News. Around 7,800,000 sentences from this collection have been parsed using Stanford parser [21] to produce typed dependency graphs [22]. Over these graphs, we applied the patterns that recover our propositions according to the corresponding syntactic structures.

4.1 Experimental Data

For creating a test set, we have manually investigated the 1600 entailment pairs from the RTE-2 Challenge development and test sets [23]. Only those noun compounds and genitives which are responsible for the entailment inference have been considered. As an example, consider the following pair taken from the RTE-2 development set:

Text: Muslims make up some 3.2 million of Germany's 82 million people...

Hypothesis: 82 million people live in Germany.

In this pair, interpreting the possessive “*Germany's 82 million people*” as “*82 million people live in Germany*” is crucial for inferring entailment. In contrast to it, the possessive “*city's airport*” in the following pair does not contribute to the entailment inference:

T: Two weeks ago, China became the first nation to operate a maglev railway commercially, when officials inaugurated a 30-kilometer-long line between downtown Shanghai and the city's airport.

H: Maglev is commercially used.

Noun-noun constructions which need not be expanded for inferring entailment were disregarded as well. For example, in the pair below there is no need to expand the possessive “*John Lennon's widow*” in order to infer entailment, because it occurs both in Text and in Hypothesis:

T: John Lennon's widow, Yoko Ono...

H: Yoko Ono is John Lennon's widow.

The noun-noun constructions containing anaphora were also ignored, for example:

T: Some 55 percent of the German public are opposed to the euro, less than 150 days before its introduction...

H: The introduction of the euro has been opposed.

Since without an anaphora resolution procedure application of axioms to such phrases is useless, we did not consider them in the evaluation.

In total, 83 textual entailment pairs containing noun-noun dependencies relevant for inferring entailment have been found in the RTE-2 development (48) and test (35) collections. Since some pairs share exactly the same paraphrase we ended with 77 different paraphrases.

Table 1. Distribution of noun-noun constructions in the RTE-2 development and test sets

Relation type	nn	poss	of	Total
Common-noun as head of a common-noun	12	-	2	14
Proper-noun as head of a proper noun	7	4	1	12
Common-noun as head of a proper-noun	15	24	9	48
Proper-noun as head of a common noun	-	3	-	3
Total	34	31	12	77

Table 1 shows the distribution of the noun-noun construction types (listed in Section 3) involved in entailment. Only 18% of paraphrases involve noun-noun dependencies between common nouns. This number shows the importance of developing automatic paraphrasing methods for noun-noun dependencies involving named entities (and instances in general). In fact, the most frequent noun-noun construction in our dataset is a common noun as governor of a proper noun (62% of paraphrasing cases).

Table 2 shows the type of propositions needed to paraphrase the noun-noun dependencies.

Table 2. Distribution of proposition types needed to paraphrase the noun-noun dependencies

	NVN	NVPN	NPN	NVNP	location	of	has	has-instance
Total	15	11	14	3	9	21	2	2

As we could expect, a relevant proportion of noun-noun dependencies are paraphrased with the preposition *of* (up to 27%). These cases are trivial to solve. The same for *has* and *has-instance* relations which are default interpretations of the noun-noun constructions. We will discuss the results obtained for the rest of paraphrasing cases.

4.2 Results and Discussion

Up to 10 paraphrases (13%) involve an additional lexical inference besides the structural one (e.g. *owner* → *head*, *leader* → *member*, *marijuana* → *drug*, etc.) The combination of the structural and lexical paraphrases is out of the scope of this work, and thus, we ignored the need of this lexical inference.

In 9 cases, the paraphrase is making explicit that one component is the location of the other component (e.g. *Hong Kong Disneyland*, *New York club*, etc...). The expressions used to make this relation explicit in the RTE pairs are: *base in*, *locate in*, *be in*, *in*, and *situate in*. For most cases we are unable to capture the specific expression used in each case to express the relationship of *location*. In many cases, the information about location is always implicit. For example, asking about “*China maker*” what we get are paraphrases such as “*maker produce in China*”, but not a generic one such as “*maker is based in China*”. However, among the paraphrasing candidates that we obtain automatically, we find the proposition of type NVPN “*be in*” in 78% of the *location* cases. Thus, the location case is solved classifying the entity as location with a standard NE recognizer, and then checking whether the proposition “*N be in LOCATION*” is among the candidates.

From the rest 43 non trivial cases, we find 63% of paraphrases (see Table 3) and lose the others (Table 4).

Interestingly, in some cases, not finding the paraphrase is the correct behavior. The clearest example is paraphrase 16 (that comes from pair 485 of RTE-2 development collection). In this case, paraphrasing axioms for “*produce vehicle*” are expecting classes such as “*world exporter*”, “*automaker*” or “*carmaker*”. The following are the first axioms sorted by $P(p|c)$:

```

1 | [named:world_exporter, vehicle] -> nvn:[produce:v]
1 | [named:japanese_company, num:[vehicle]] -> nvn:[produce:v]
0.2380952380 | [named:financial_center, vehicle] -> nvn:[produce:v]
0.1666666667 | [named:japanese_company, vehicle] -> nvn:[produce:v]
0.1470588235 | [named:automaker, num:[vehicle]] -> nvn:[produce:v]
0.1190476190 | [named:auto_maker, num:[vehicle]] -> nvn:[produce:v]
0.1142857142 | [named:carmaker, num:[vehicle]] -> nvn:[produce:v]
...

```

However, none of these classes are suitable for the instance “*United Nations*”, whose most frequent classes in our collection are:

```
has_instance(organization, 'United': 'Nations'):26.
has_instance(forum, 'United': 'Nations'):12.
has_instance(target, 'United': 'Nations'):11.
has_instance(body, 'United': 'Nations'):9.
has_instance(agency, 'United': 'Nations'):9.
has_instance(instrument, 'United': 'Nations'):8.
has_instance(group, 'United': 'Nations'):8.
...
```

This example is showing how our procedure is considering the classes discovered automatically attached to the given instances: if instead of “*United Nations vehicle*” we query for “*Toyota vehicle*” then “*Toyota produce vehicle*” appears as a paraphrase among the given candidates (sorted by $P(\text{pli})$):

```
[named:'Toyota', vehicle] ->
-> in:[] | 0.093882216343849
-> nvn:[develop:v] | 0.0771943521830898
-> nvn:[build:v] | 0.0692048562887729
-> nvn:[sell:v] | 0.0676145829704216
-> i_has_n:[] | 0.0654674378035677
-> nvn:[produce:v] | 0.0138156885695286
-> nvn:[make:v] | 0.00941897686698851
-> nvn:[export:v] | 0.00765968688431721
-> nvn:[recall:v] | 0.00449006137810559
-> nvn:[assemble:v] | 0.00389949527020733
...
```

5 Conclusions and Future Work

The data set shows that there are cases of textual entailment that require the paraphrasing of noun compounds and genitives by recovering their implicit predicate. As observed in the dataset, a significant proportion of these noun-noun dependencies include proper nouns. We have presented a unified methodology to paraphrase noun-noun dependencies able to cope with Named Entities. This methodology relies in the use of Proposition Stores for the identification of candidate paraphrases.

We have observed that some noun-noun dependencies don’t require the retrieval of implicit predicates. In particular, when one of the dependents is a location, instead of retrieve implicit predicates, what we need is to assess this locative relation. A similar case is the *class-instance* relation denoted by noun-noun dependencies governed by a Named Entity.

We have tested if Proposition Stores can provide the paraphrasing candidates needed for the interpretation of a particular text. A Proposition Store built from a

collection of 200,000 documents provides 63% of the non-trivial paraphrases in the dataset. This fact rises the question about the size of the collection. Obviously, larger collections will provide more paraphrases, but we can't expect they will provide all. We will need to leverage the context of the noun-noun dependency in the document not only for the final selection of the most appropriate paraphrase, but also to find the relevant candidates in some cases. For example, now we are ignoring that the relevant class of an instance might be pointed out in the context of the noun-noun dependency we want to paraphrase: to paraphrase the noun-noun dependencies involving the instance we could use directly its class given by the context.

Another alternative that deserves attention is the possibility to consider a relatedness probability between classes. Certainly this could increase the number of candidate paraphrases, but we introduce the risk of adding more noise. In the setting presented we assume that the different classes are unrelated which is not always true. However, the framework is ready to explore the inclusion of probabilities for class-class relatedness.

Finally, we leave for future work the development of an inference procedure that, considering the dependency context, selects the paraphrases that provide better interpretations of the whole text.

Table 3. Paraphrases found among candidates (ignoring lexical paraphrasing)

NPN	Agency Chairman ↔ head of Agency case against Jackson ↔ Jackson trial population for Missouri ↔ Missouri's population output in the Gulf of Mexico ↔ the Gulf of Mexico's output council to Rome ↔ Rome's council Berlin's landmark ↔ landmark in Berlin party in Spain ↔ Spain's party trade ban ↔ ban on trade trade in ivory ↔ ivory trade problem with engine ↔ engine problem ivory ban ↔ ban on ivory trade increase of activity ↔ increase in activity
NVN	U.S. Ambassador ↔ Ambassador represents the U.S. ETA bombing ↔ ETA carried_out bombing Gaza's inhabitant ↔ Israel has inhabitant Nicholas Cage marry wife ↔ Nicholas Cage's wife comments made by Wells ↔ David Wells' comments Microsoft's monopoly ↔ Microsoft holds a monopoly
NVNP	wife of Joseph Wilson ↔ wife is married to Joseph Wilson
NVPP	Vietnam veteran ↔ veteran comes from Vietnam Shapiro work in office ↔ Shapiro's office people live in Germany ↔ Germany's people McNabb throwing for 357 yards ↔ McNabb's 357 yards group led by Abu Musab al-Zarqawi ↔ Abu Musab al-Zarqawi's group director of the Council ↔ director works for the Council city of Olympic Games ↔ Olympic Games hold in city wife of Sean Carr ↔ wife married to Sean Carr

Table 4. Paraphrases not found

1	brainchild of Dave McCool ↔ Dave McCool is inventor of brainchild
2	semi-final on Friday ↔ Friday's semi-final
3	counter-demonstration of Palestinians ↔ counter-demonstration by Palestinians
4	U.N. Javier Perez ↔ Javier Perez is employed by the U.N.
5	Fernandez of FEMA ↔ Fernandez works for FEMA
6	Wilson's "Two Trains Running" ↔ "Two Trains Running" was written by Wilson
7	Aki Kaurismaki's 'Juha' ↔ Kaurismaki directed 'Juha'
8	Kaspersky Labs branded the spyware ↔ Kaspersky Labs' software
9	employee of Bush ↔ Bush appointed employee
10	head of branch ↔ head commands branch
11	Court Judge ↔ Judge occupies a post at the Court
12	browser called Mosaic ↔ browser Mosaic
13	Tunguska meteorite ↔ meteorite fell in Tunguska
14	Gerhard Schroeder's Social Democrats ↔ Schroeder belongs to the Social Democrats
15	native of Beckemeyer ↔ native resides in Beckemeyer
16	United Nations vehicle ↔ The United Nations produces vehicles

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science and Innovation through the project Holopedia (TIN2010-21128-C02).

References

1. Pustejovsky, J.: The Generative Lexicon. *Computational Linguistics* 17(4) (1991)
2. Lapata, M., Lascarides, A.: A probabilistic account of logical metonymy. *Computational Linguistics* 29(2) (2003)
3. Van Durme, B., Schubert, L.: Open Knowledge Extraction through Compositional Language Processing. In: *Symposium on Semantics in Systems for Text Processing* (2008)
4. Clark, P., Harrison, P.: Large-scale extraction and use of knowledge from text. In: *The Fifth International Conference on Knowledge Capture, K-CAP 2009* (2009)
5. Peñas, A., Hovy, E.: Semantic Enrichment of Text with Background Knowledge. In: *1st International FAM-LbR Workshop, NAACL 2010* (2010)
6. Levi, J.N.: *The Syntax and Semantics of Complex Nominals*. Academic Press (1978)
7. Downing, P.: On the creation and use of English compound nouns. *Language* 53(4) (1977)
8. Finin, T.: *The Semantic Interpretation of Compound Nominals*. Ph.D. thesis, University of Illinois at Urbana Champaign (1980)
9. Lauer, M.: Corpus statistics meet the compound noun. In: *Proceedings of the 33rd Meeting of the Association for Computational Linguistics* (1995)
10. McDonald, D.B.: *Understanding Noun Compounds*. Tech. rep., CMU Technical Report CS-82-102 (1982)
11. Alshawi, H.: *Memory and Context for Language Interpretation*. Cambridge University Press, Cambridge (1987)
12. Fan, J., Barker, K., Porter, B.W.: The knowledge required to interpret noun compounds. In: *18th International Joint Conference on Artificial Intelligence* (2003)

13. Warren, B.: *Semantic Patterns of Noun-Noun Compounds*. Acta Universitatis Gothoburgensis, Goteborg (1978)
14. Barker, K., Szpakowicz, S.: Semi-automatic recognition of noun modifier relationships. In: *Proceedings of the 36th Annual Meeting of the ACL* (1998)
15. Girju, R., Moldovan, D., Tatu, M., Antohe, D.: On the semantics of noun compounds. *Computer Speech and Language* 19(4) (2005)
16. Tratz, S., Hovy, E.: Taxonomy, Dataset, and Classifier for Automatic Noun Compound Interpretation. In: *Proceedings of the 48th Annual Meeting of the ACL* (2010)
17. Hendrickx, I., et al.: *SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals*. In: *Proceedings of SemEval 2010* (2010)
18. Nakov, P., Hearst, M.A.: Using Verbs to Characterize Noun-Noun Relations. In: Euzenat, J., Domingue, J. (eds.) *AIMSA 2006. LNCS (LNAI)*, vol. 4183, pp. 233–244. Springer, Heidelberg (2006)
19. Butnariu, C., Kim, S.N., Nakov, P., Seaghdha, D.O., Szpakowicz, S., Veale, T.: Semeval-2 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In: *Proceedings of the 5th International Workshop on Semantic Evaluation* (2010)
20. Peñas, A., Hovy, E.: Filling Knowledge Gaps in Text for Machine Reading. In: *23rd International Conference on Computational Linguistics, COLING 2010* (2010)
21. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423–430 (2003)
22. Marneffe, M., Manning, C.D.: The Stanford typed dependencies representation. In: *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation* (2008)
23. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The second PASCAL Recognising Textual Entailment challenge. In: *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment* (2006)

Age-Related Temporal Phrases in Spanish and Italian

Sofía N. Galicia-Haro¹ and Alexander Gelbukh²

¹ Faculty of Sciences UNAM University City, Mexico City, Mexico
sngh@ciencias.unam.mx

² Center for Computing Research, National Polytechnic Institute, Mexico
www.Gelbukh.com

Abstract. This paper reports research on temporal expressions. The analyzed phrases include a common temporal expression for a period of years reinforced by an adverb of time. Some of these phrases are age-related expressions. We analyzed samples of this type obtained from the Internet for Spanish and Italian to determine appropriate annotations for marking up text and possible translations. We present the results of comparison for four selected classes.

Keywords: temporal expressions, people's age, multilingual comparison.

1 Introduction

Some words or whole sequences of words in a text are temporal expressions: for example, *today*, *Tuesday 28*, *three weeks*, *about a year and a half*; each refers to a certain period of time. Such words or sequences of words mainly share a noun or an adverb of time: *today*, *month*, *year*. Syntactic variety in temporal expressions has enabled a diversity of human communication. This diversity means that, in natural language processing, temporal expression software recognition copes with the problem of deciding whether a word or a sequence is a temporal expression.

A person's age is the amount of time since that person was born and different syntactic compounds are employed to describe it. In Spanish there are phrases recognized by an initial adverb: for example, *around*, *still*; they end with the noun of time, for example *year*, and they describe a person's age. For example: *aún a sus 90 años* "even at his 90 years", *ahora a mis 19 años* "now at my age of 19 years." These phrases are very interesting since they reinforce the meaning of time in different forms and they are the phrases that we target in this work.

Since temporal expression recognition is an important part in machine translation, we analyzed the translations of such temporal expressions that we are interested in. Due to the similarity of construction in Spanish and Italian, we supposed that these phrases would be similar in both languages. However we found that the automatic Italian translation¹ "*ancora ai suoi 26 anni*" for the phrase *aún a sus 26 años* is not common in Italian.

¹ <http://www.online-translator.com/Default.aspx/Text>

Automatic recognition of expressions of time was introduced in the Named Entity Recognition task of the Message Understanding Conferences,² where temporal entities were tagged as “TIMEX.” Since then, temporal annotation schemes have been developed; for example, [1] and [2] for English, [3] for Italian, and [4] for Spanish. The authors in [1] produced a guideline intended to support a variety of applications in the performance of some useful tasks. As they pointed out, the guideline was not intended to represent all the varieties of temporal information.

The guidelines consider temporal expressions that reference calendar dates, times of day, durations or sets. They considered lexical triggers to identify temporal expressions. A lexical trigger is a word or numeric expression whose meaning conveys a temporal unit or concept. To be a trigger, the referent must be able to be oriented on a timeline, or at least oriented with relation to a time (past, present, future). However, the phrases of interest to us do not fulfill the described trigger characteristics and were not considered in those annotation guidelines.

In this article, we present a web-based analysis carried out to compare such Spanish temporal expressions with age-related temporal phrases in Italian, with the objective of determining appropriate annotations for marking up text and possible translations relating to them. In section 2, we present the characteristics of the Spanish phrases and the method we applied to obtain the materials for the comparison. In section 3, we describe the Italian phrases obtained with a similar method. In section 4, we present the comparison of such phrases. Section 5 concludes.

2 Age-Related Temporal Expressions in Spanish

Usually a person’s age is described by Spanish temporal expressions including the time noun *años* “years.” They can be recognized in the following ways:

- with the string *de edad* (lit. “of age”) after the word *años*
Example: *un vendedor de 38 años de edad* (a seller who is 38 years old)
- with the preposition: *de* (of) after an animated noun or a person’s name and before the number of years, sometimes delimited by commas
Example: *la niña de 11 años* (the 11-year-old girl)
- with the strings: *la edad de* (lit. “the age of”) before the number of years.
Such phrases can be preceded by different prepositions: *a, con, desde, hasta*
Example: *falleció ayer a la edad de 95 años* (died yesterday at 95 years old)

There are, however, other temporal expressions that describe people’s ages: for example, *aún a sus 65 años*, lit “still at his 65 years”, *de alrededor de 20 años*, lit. “of about 20 years.” These temporal phrases denote a point in the timeline of a person; it could be a point in the timeline of the events related in the sentence or a point in a tangential timeline.

In [5], we analyze the context for Spanish temporal phrases that begin with an adverb of time (AdvT) and end with a noun of time (TimeN) expressing a person’s

² <http://timexportal.wikidot.com/timexmuc6>

age. We can observe the relation between the groups of words in the following examples:

1. *A sus 30 años Juan se comporta como niño*
2. *Aún a sus 30 años Juan se comporta como niño*
3. *Hoy a sus 30 años Juan se comporta como niño*

The sentences describe the same main fact: *John, who is 30 years old, behaves like a child*, but they tell us something else when we introduce a modifier (*aún* “still,” *hoy* “today”) in each one: they argue for different conclusions.

- Even at 30 years old \Rightarrow in spite of his age he behaves as if he were a child
- Today, at 30 years old \Rightarrow today he behaves like a child.

The adverbs “even” and “today” make such conclusions obligatory and reinforce the meaning of time in different forms. Both adverbs are related to time duration; one strict reading refers to 24 hours and the other to a longer period of time, but they also imply a direct judgment on the perception of the speaker, on the behavior of the subject or on both.

2.1 Material Acquisition

The method we presented in [5] first allows the manual selection of examples representing the class being considered: a different combination of an adverb and a preposition before the number of years. Then a program retrieves from the Internet a more representative group of examples for such classes.

1. Acquisition of classes

For class acquisitions we used a collection of texts compiled from a Mexican newspaper that is published daily on the Web. The texts correspond to diverse sections, the economy, politics, culture, sport, and so forth, from 1998 to 2002 [6].

We wrote a program to extract the sentences matching the pattern: AdvT–something–TimeN, where: *something* corresponds to a sequence of up to six words³ without punctuation marks, verbs or conjunctions; TimeN corresponds to *año*, *años* “year, years”; and AdvT to adverbs of time, 51 elements from a dictionary.⁴

We manually selected one arbitrary example representing a class, the five resulting classes were: *aún a*, *aún con*, *actualmente de*, *alrededor de*, *ahora de*.

2. Acquisition of examples from classes

Since the newspaper text collection contains a subset of all possible temporal phrases expressing the ages of people, we analyzed methods to obtain a more representative group of phrases and we chose to look on the Internet for examples. This option allowed us to find phrases generated by native speakers more quickly, including the more common collocations. Nevertheless, it is known that searching the Internet has drawbacks but we decided to do so on the basis that we did not know how the results

³ A larger quantity of words does not guarantee any relation between the AdvT and the TimeN.

⁴ DRAE, Real Academia Española. (1995): *Diccionario de la Real Academia Española*, 21 edición (CD-ROM), Espasa, Calpe.

were classified [7]. The quantity of pages automatically obtained was limited to 50, to obtain 500 snippets.

SEARCH(C)

For each phrase of type ADV-**-NounT* or string-**-NounT* in C

(1) Obtain 100 examples from the Internet

(1.1) $D = \{\text{examples excepting such instances where } * \text{ includes verbs or}$

(1.2) Print D

(2) Classify them according to such words retrieved by *

(3) For each group of phrases sharing words retrieved by *, assign a class D_i

(3.1) $F = \text{class } D_i$

(3.2) SEARCH(F)

UNTIL no new elements are obtained

Fig. 1. Algorithm to obtain variants of temporal expressions

The main idea of obtaining more examples from the Internet is based on obtaining a few examples from the newspaper texts, simplifying them (eliminating determinants, adjectives, etc.) and searching for variants by including Google's asterisk facility [8]. The whole procedure is shown in Figure 1. For example: for the phrase *aún con sus jóvenes 48 años*, the string when simplified becomes "*aún con año*" and the search is "*aún con * años*", using the Google search engine tool limited to the Spanish language, where the asterisk substitutes for the eliminated words. Google returns hits where there is a string of words initiated by "*aún con*" and then a sequence of words, ending with "*años*."

The process is repeated several times until no new repeated phrases are obtained, determining the sequences of words that appear with higher frequency. In addition, some phrases not corresponding to the specific temporal phrases were picked up. These phrases were eliminated in the manual identification at the end of each cycle to reduce the quantity of Google searches.

After this compilation of examples, 18 classes were manually selected and these appear in the first column of Table 1, where NUM treats numbers represented by digits or letters. Some of the 18 classes obtained from the Internet seem to preserve their meaning independently of the context and others require some form of words in context to denote the age of a person.

The second column shows the number of examples obtained, after the elimination of phrases where there is no relation between the AdvT and TimeN. Since the examples were automatically obtained from the snippet, some of them were not considered because of the lack of text when the sentences were split and the context omitted around the searched phrase. Fewer than 10% are errors because of the short snippet. Column 3 shows the results after manual syntactic and semantic analyses of the context.

Many studies focused on having a corpus that models the whole language. However, for inducing information for annotation and translation of the phrases of interest to us, we collected just a particular subset of language, the one that corresponds to them. Thus, the research we report here refers to aspects related to the collection that has been skewed by design.

Table 1. Selected phrases for age-related temporal phrases in Spanish

Type of phrase	# examples	% age-related
aún a tus NUM años	7	100
aún con mis pocos NUM años	1	100
aún con mis cortos NUM años	2	100
aún con sus escasos NUM años	4	100
aún con tus casi NUM años	1	100
ahora a mis NUM años	182	99
aún a sus NUM años	293	96
aún hoy a sus NUM años	38	92
aún con sus NUM años	109	86
ahora de NUM años	352	86
alrededor de los NUM años	355	84
actualmente con NUM años	270	80
ahora con casi NUM años	118	67
aún con sus casi NUM años	7	57
ahora con más de NUM años	90	46
actualmente de NUM años	28	36
ahora a los NUM años	132	36
actualmente de unos NUM años	16	19

3 Italian

For the analysis of Italian temporal expressions reinforced by an adverb of time, we used one collection of texts compiled from Italian newspapers [9]. The collection contains 16,247 sentences from *La Stampa* (20/10/1989), *La Stampa* (21/10/1989), *Il Mattino* (20/10/1989), *Il Mattino* (20/10/1989), *La Repubblica* (21/10/1989) and *Corriere della Sera* (21/10/1989).

We found that the most common Italian phrases expressing age can be recognized in the following ways:

— with the word “di” before the number of years.

Example: signora di 52 anni (woman 52 years old), Soggetto di sesso maschile di 72 anni di età “Subject of male sex of 72 years of age”

— with the word “a” before the number of years.

Example: Graziella Gaeta, morta a 19 anni (Graziella Gaeta, died at 19 years old)

— a number and the word “*anni*” after the person’s name, delimited by commas or parentheses.

Example: Rose Cordero, 19 anni: top model e quasi mamma (Rosa Cordero, 19 years old, top model and mother)

— including in addition the string: *all’età, di età* (lit. “the age”), before and after the number of years.

Example: entrato nel Pcus all’età di 21 anni, (entering the PCUS at 21 years old)

To acquire Italian temporal phrases expressing age similar to the Spanish phrases detailed in section 2, we applied the same method described in 2.1. We first applied the method described in Acquisition of classes but no results of the type Adv-something-Time were obtained in the small collection of newspaper texts.

We have access to La Repubblica Corpus⁵ online and we manually selected the following classes: *adesso * anni, anche * anni, ancora * anni, attualmente * anni, intorno * anni*, and *oggi * anni*.

We then applied the method described in 2.1, Acquisition of examples from classes, to access the Google search engine tuned to the Italian language. The program obtained 3,452 examples but eliminated the phrases not corresponding to the temporal phrases of interest to us. The results obtained from the Internet produced the examples detailed in Table 2, for such classes with person’s age meaning higher than 10%. The columns correspond to the same parameters as the previous table. We observed that only three types of classes correspond to more than 80% of temporal expressions with age-related meanings.

4 Comparison

According to [10], in many instances, statistically equivalent constructions are not semanto-syntactically equivalent. Considering quantitative data, a pair of phrases of two different languages could be non-equivalent on at least two counts: intralinguistic and interlinguistic (contrastive). The intralinguistic results can also be obtained cross-linguistically and more directly by looking at various translations of a given construction into another language. Statistically, comparisons can be conducted both in texts which are attested as translations and on texts which are not translations but are comparable on account of being written on a similar topic, by similarly qualified authors using similar registers, and so forth. For this reason, although our materials do not come from translations they are comparable and they possess qualities in common, that is *tertium comparationis*.

We note that although Spanish and Italian exhibit a substantial degree of similar realizations of age phrases, there are some divergences. Some of them correspond to prepositions and the incorporation of determinants in Spanish, which are not present in the same correspondent phrases in Italian. The elements to be compared in the age phrases for both languages are: 1) adverbs, 2) surface structure of the phrase between

⁵ <http://sslmitdev-online.sslmit.unibo.it/corpora/how-to.php?path=&name=Repubblica>

adverb and noun *years*, and 3) adjective presence. For this comparison we divided the results into four groups corresponding to the adverbs: *ahora/adesso* “now”, *alrededor/intorno* “around”, *actualmente/attualmente* “at present”, *aún/ancora* “still”.

We matched the above described groups in Spanish and Italian, considering first the adverb, then the age meaning and finally the percentage of phrases with age-related meaning. These matched groups are shown in Tables 3 to 6. We also considered as stated by [10] that two linguistic items across languages are statistically equivalent if, in comparison with other synonymous constructions, they have maximally similar frequencies of occurrence in the relevant texts. In general, we compared classifying examples as identical, different in some respects, and with no equivalent.

Table 2. Selected phrases for age-related temporal phrases in Italian

Type of phrase	# examples	% age-related
ancora adesso a quasi NUM anni	1	100
oggi con i suoi NUM anni	21	100
anche a NUM anni	215	81
intorno agli NUM anni	186	77
ancora adesso a NUM anni	86	77
intorno a NUM anni	50	74
adesso a NUM anni	111	73
intorno ai NUM anni	370	71
ancora con i suoi NUM anni	6	67
ancora a quasi NUM anni	40	60
ancora a NUM anni	91	59
ancora oggi a NUM anni	80	51
anche sui NUM anni	87	44
attualmente a NUM anni	38	39
oggi a quasi NUM anni	45	38
anche di NUM anni	95	33
anche fino a(i) NUM anni	13	31
attualmente di NUM anni	134	29
oggi a NUM anni	107	28
adesso di NUM anni	86	22
attualmente intorno ai NUM anni	15	13
anche ai NUM anni	85	11

The number of groups including adjectives is higher in Spanish than in Italian. However, the examples obtained from the Internet are fewer compared with the examples obtained for groups without adjectives. We also observed that the syntactic structure is rather similar but the percentage of phrases with age-related meaning is very different. The most productive group for age meaning in Italian is the group for the *ancora* “still” adverb.

The group associated with the adverb “now” (see Table 3) contains one class classified as identical in the first row, one classified as different in some respects in the third row, and three classes classified as with no equivalent, two for Spanish and

one for Italian. The groups which showed differences in some respects differ in the article (*los*) included in Spanish and the percentage of age-related meaning, where the Italian group is more productive. The classes in the first row differ in the percentage of age-related meaning which is contrary to the previous case: the Spanish group is more productive. Productive similarity is important since it could be considered as a basis for preferring a specific structure for a possible translation.

The three classes classified as having no equivalent differ in structure. The Spanish classes include an adjective which gives a slight difference in meaning: one considering the speaker referent and the other because of a sense of approximation and a different preposition. The Italian class has a different preposition and very low productivity.

Table 3. Group of phrases related to a person's age initiated by the adverb *ahora/adesso*

Type of phrase	# examples/ % age-related	Type of phrase	# examples/ % age-related
ahora de NUM años	352/86	adesso di NUM anni	86/22
		adesso da NUM anni	69/1
ahora a los NUM años	132/36	adesso a NUM anni	111/73
ahora a mis NUM años	182/99		
ahora con casi NUM años	118/67		

The group of phrases initiated by the adverb “around” (see Table 4) is the group with fewer variants. The structure is rather similar and we classified them as different in some respects because of the difference in prepositions. The three Italian classes differ in the absence or inclusion of an article, the inclusion has two variants according to whether the number of years begins with a consonant or a vowel. The Spanish counterpart includes the same article. All the classes are highly productive for the age-related meaning.

Table 4. Group of phrases related to a person's age initiated by the adverb *alrededor/intorno*

Type of phrase	# examples/ % age-related	Type of phrase	# examples/ % age-related
alrededor de los NUM años	355/84	intorno ai NUM anni	370/71
		intorno agli NUM anni	186/77
		intorno a NUM anni	50/74

The group associated with the adverb “at present” (see Table 5) contains one class classified as identical in the second row, two classes classified as different in some respects in the first and last rows, and one class with no equivalent in the third row.

Table 5. Group of phrases related to a person's age initiated by the adverb *actualmente/attualmente*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
actualmente con NUM años	270/80	attualmente con NUM anni	12/8
actualmente de NUM años	28/36	attualmente di NUM anni	134/29
		attualmente a NUM anni	38/39
actualmente de unos NUM años	16/19	attualmente intorno ai NUM anni	15/13
		attualmente di circa NUM anni	29/7

The identical class (*actualmente de NUM años – attualmente di NUM anni*) has the same meaning and structure, and similar productivity. The first row shows classes with identical structures but a great difference in productivity for the age-related meaning and number of examples; we classified it as different in some respects to undertake a deeper analysis. The last row matches the three classes because of their meaning: an approximation to a number of years and around certain years, although the structure is different among them. In Spanish the around meaning is obtained by an indefinite article, in Italian by an adverb and an adverbial phrase.

The fourth group corresponds to the adverb “still” (see Table 6). It does not contain identical classes. There are three classes classified as different in some respects and eight with no equivalents. The three classes different in some respects are:

Table 6. Group of age's person phrases initiated by the adverb *aún/ancora*

Type of phrase	# ex/ % age	Type of phrase	# ex/ % age
aún con mis pocos NUM años	1/100		
aún con mis cortos NUM años	2/100		
aún con sus escasos NUM años	4/100		
aún a tus NUM años	7/100		
aún a sus NUM años	293/96		
		ancora a NUM anni	91/59
aún hoy a sus NUM años	38/92	ancora oggi a NUM anni	80/51
		ancora adesso a NUM anni	86/77
aún con sus NUM años	109/86	ancora con i suoi NUM anni	6/67
aún con tus casi NUM años	1/100	ancora a quasi NUM anni	40/60
aún con sus casi NUM años	7/57		
		ancora adesso a quasi NUM anni	1/100

1. aún hoy a sus NUM años – ancora oggi a NUM anni

In this match there is one difference in structure: the possessive adjective related to the person whose age is described in the Spanish class; and a big difference in the percentage of age-related meaning

2. *aún con tus/sus casi NUM años* – *ancora a quasi NUM anni*

There is a difference in structure: the possessive adjective related to the person whose age is described in the Spanish class. Also, they differ in preposition and the productivity of the Spanish classes is very low.

3. *aún con sus NUM años* – *ancora con i suoi NUM anni*

The only difference in structure is due to the inclusion of an article in the Italian class but the productivity of the Spanish class is rather low.

The no equivalent classes have the following discrepancies:

- The Spanish classes: *aún con mis pocos NUM años*, *aún con mis cortos NUM años*, *aún con sus escasos NUM años* correspond to very few examples including adjectives
- The Spanish classes: *aún a tus NUM años*, *aún a sus NUM años*, are very productive. They share structure with the Italian class with no equivalent *ancora a NUM anni* but productivity and the percentage of age-related meaning are rather different
- The Italian classes: *ancora adesso a NUM anni* and *ancora adesso a quasi NUM anni* include another adverb and there are no similar Spanish counterparts.

We consider that identical classes could be identically translated and the classes that differ in some respects could be translated by means of translation templates, as we describe below. The classes with well defined contexts for age-related meaning or with meanings independent of the context could be included in annotation guidelines as we propose in the following section.

4.1 Annotation

Annotation of temporal expressions in Spanish, developed in [4], considers those phrases expressing day times, dates and duration (TIMEX3). Annotation for temporal expressions in Italian, developed in [3, 11], includes the annotation of events as a primary objective. Generally, temporal expressions are annotated as entire constituents, typically noun phrases (e.g., 65 years). For more complex temporal expressions the annotation is divided, for example:

Ahora hace casi veinte años de la caída del muro de Berlín. “Now almost twenty years since the fall of the Berlin Wall”, it is annotated in [4] as follows:

- a. extent: *ahora* type:DATE value:PRESENT REF mod:--
 b. extent: *hace casi veinte años* type:DURATION value:P20Y mod:LESS THAN

Owing to the insertion of phrasal pronominal modifiers in the temporal phrases discussed above, temporal expressions denoting people’s age in Spanish and Italian should be annotated following the directions in [1] for the TLINK tag. TLINK is a temporal link that could represent the relation between two temporal elements: TIMEX3-TIMEX3.

We propose to include the type AGE in addition to DATE, TIME, DURATION and SET. The phrases we consider for this annotation are those, extracted from the examples, that are more productive with appropriate contexts. We include the

prepositional phrase in the extent of the temporal phrase since its value could be a mod attribute value. For example, we annotate the phrases *ahora a mis* NUM años in such proposed form:

ahora a los NUM años

```
<TIMEX3 tid="t1" type="DATE" value="PRESENT_REF" >ahora</TIMEX3>
<TIMEX3 tid="t2" type="AGE" value="PnumY" temporalFunction="TRUE"
anchorTimeID="t1"> a los NUM años </TIMEX3>
<TLINK timeID="t1" relatedToTime="t2" relType="SIMULTANEOUS"/>
```

4.2 Translation

The Spanish temporal expressions form a contiguous sequence given an appropriate context, and should be translated into an entire sequence as a multi-word unit. We propose their translation by specific templates as a preprocessing step applied before any other type of processing. Preprocessing is considered in different works, for example, in [12] the application of a rule-based language-independent chunker that contains constituency rules (they add a chunk boundary when two Part of Speech codes cannot occur in the same constituent), in [13] applying rule-based pre-processing for normalization of time and date phrases.

As considered in [14], the meaning of a lexical unit must be preserved in the target language, even though it may take different syntactic forms in the source and target languages. When translating a functional preposition, the identity of the source language preposition is thereby of less importance. In some cases, described above, the prepositions are different in the source and target languages and the binding of the prepositional phrases have a variant.

We present an example of translation by templates, the Spanish phrase *alrededor de los* NUM años could be translated to three forms in Italian: *intorno a* NUM anni, *intorno ai(agli)* NUM anni. This group was classified as different in some respects: one difference is the preposition; another difference is the Italian class without article (*intorno a* NUM anni).

We found two well defined types of context to define the templates:

1. The group with the string 'of age' in right or left context: "*de edad*", "*di età*"

This group contains 17% of Spanish examples with person's age meaning with the right context "*de edad*". There are 10% of Italian examples with such string in the left or right context for the group *intorno ai(agli)* NUM anni and only one example for the group *intorno a* NUM anni in the left context. See Fig. 2 where * means that the compounds "*ai*", "*agli*" appear according to the initial consonant or vowel in NUM.

2. The group where these temporal expressions initiate the sentences or they appear after punctuation signs.

This group contains 81 Italian examples and 61 Spanish examples. The Italian examples correspond to *intorno a* NUM anni (8%), *intorno ai* NUM anni (11%), *intorno agli* NUM anni (24%). The Spanish examples correspond to the 20% of the expressions with person's age meaning.

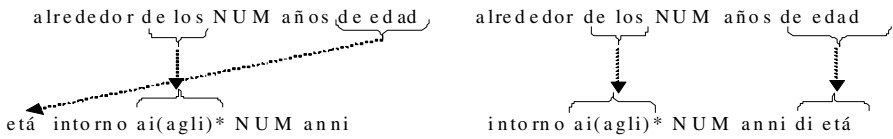


Fig. 2. Translation templates for the adverb “around” in the “of age” context

5 Conclusions

The variety in the structure of temporal expressions requires analysis of different combinations of classes of words. We analyzed temporal expressions including the noun of time *year* that are modified by an adverb of time and the whole phrase expressing a person’s age. These phrases are interesting since adverbs make some inferences obligatory and reinforce the meaning of time in different forms. Besides the time duration involved, they imply a direct judgment on the perception of the speaker, on the subject or on both.

We presented a method of automatically extracting classes of temporal phrases by means of newspaper texts for Spanish, and manually in an online access to newspaper texts for Italian. Then we present a method of increasing such classes when only a few examples are compiled. A more representative sample was compiled from the Internet for both languages.

Our study provides insights into the cross-lingual behavior of the temporal structure of person’s age expressions, particularly in the type AdvT–something–TimeN as realized in Spanish and Italian.

Acknowledgements. Work partially supported by Mexican Government (CONACYT, SNI), Government of Mexico City (ICYT project PICCO10-120), European project 269180 WIQ-EI, project “Answer Validation through Textual Entailment” CONACYTDST (India), and National Polytechnic Institute, Mexico (projects SIP 20111146, 20113295; COFAA).

References

1. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2005 Standard for the annotation of temporal expressions. MITRE Corporation (2005)
2. Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML annotation guidelines Version 1.2.1 (2006), http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf
3. Caselli, T.: It-TimeML: TimeML annotation scheme for Italian version 1.3.1 technical report (September 23, 2010), http://puma.isti.cnr.it/download.php?DocFile=2010-TR-002_0.pdf&langver=it&idcode=2010-TR-002&authority=cnr.ilc&collection=cnr.ilc&check=

4. Saurí, R., Saquete, E., Pustejovsky, J.: Annotating time expressions in Spanish. *TimeML Annotation Guidelines. Version TempEval 2010* (2010)
5. Galicia-Haro, S.N., Gelbukh, A.F.: Assessing Context for Age-Related Spanish Temporal Phrases. In: Coyle, L., Freyne, J. (eds.) *AICS 2009. LNCS*, vol. 6206, pp. 92–102. Springer, Heidelberg (2010)
6. Galicia-Haro, S.N.: Using electronic texts for an annotated corpus building. In: *4th Mexican International Conference on Computer Science, ENC 2003, Mexico*, pp. 26–33 (2003)
7. Kilgarriff, A.: Googleology is bad science. *Computational Linguistics* 33, 147–151 (2007)
8. Gelbukh, A.F., Bolshakov, I.A.: Internet, a true friend of translator: the Google wildcard operator. *International Journal of Translation* 18(1-2), 41–48 (2006)
9. Burr, E.: *Corpus of Italian Newspapers* (1993), <http://ota.ahds.ac.uk/headers/1723.xml>
10. Krzeszowski, T.: *Contrasting languages: The scope of contrastive linguistics*. Mouton De Gruyter, Berlin (1990)
11. Caselli, T.: *Time, events and temporal relations: An empirical model for temporal processing of Italian texts*. Tesi etd-04242009-113147. ILC- CNR, Pisa (2009)
12. Lefever, E., Macken, L., Hoste, V.: Language-independent bilingual terminology extraction from a multilingual parallel corpus. In: *Proceedings of the EACL 2009*, pp. 496–504 (2009)
13. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. *Computational Linguistics* 30, 417–449 (2004)
14. Gustavii, E.: Target language preposition selection—an experiment with transformation-based learning and aligned bilingual data. In: *Proceedings of EAMT 2005* (2005), http://stp.lingfil.uu.se/~ebbag/Target_Language_Preposition_Selection.pdf

Can Modern Statistical Parsers Lead to Better Natural Language Understanding for Education?

Umair Z. Ahmed*, Arpit Kumar, Monojit Choudhury, and Kalika Bali

Microsoft Research India,
“Vigyan”, #9, Lavelle Road, Bangalore 560 025, India
{t-umaira,t-arkum,monojitc,kalikab}@microsoft.com

Abstract. We use state-of-the-art parsing technology to build GeoSynth – a system that can automatically solve word problems in geometric constructions. Through our experiments we show that even though off-the-shelf parsers perform poorly on texts containing specialized vocabulary and long sentences, appropriate preprocessing of text before applying the parser and use of extensive domain knowledge while interpreting the parse tree can together help us circumvent parser errors and build robust domain specific natural language understanding modules useful for various educational applications.

Keywords: NLP for education, statistical parsers, evaluation of parsers, domain adaptation, domain ontology.

1 Introduction

A machine that can understand and solve a mathematical word problem has long been the holy-grail of Artificial Intelligence and computer-assisted tutoring systems. It is no surprise that there have been several attempts at building systems to solve word problems in various areas of mathematical and allied [1]. Any system that aims to solve word problems will have Natural Language Understanding (NLU) at its core, the primary aim of which is to map natural language description of a problem to certain machine-readable logical representation. This requires components that mirror a cognitive process: the ability to “parse” or understand the structure of the language itself, and the ability to use domain specific knowledge to arrive at logical representations that can then be used to infer what is being asked.

Over past few decades there has been significant advancement in natural language parsing technology with some of the best statistical parsers for English today having around 90% accuracy for phrase identification on a gold standard [2]. However, many systems for solving word problems use shallow syntactic processing tools such as parts-of-speech tagging or noun phrase identification [1]. This is primarily because a good parser for any educational technology has to meet very high accuracy criterion and deal with complex issues such as adjunct attachments, identification and coordination of complex NPs, denominalisation of verbs, and anaphora resolution [2]

* The work was done during the authors’ internship at Microsoft Research India.

to avoid a small parsing error quickly snowballing into a major system fault. Further, statistical parsers trained on general language data, usually from newswires, cannot be expected to perform well on a specialized vocabulary of a particular domain like mathematics, physics or chemistry. Any domain-adaptation for these parsers is unlikely to be trivial as most statistical parsers are typically trained on millions of words [3]. A great deal of work has been done on the comparison of parser performance on different text corpora; for example, Gildea [4] evaluates a statistical parser by training on one corpus (Wall Street Journal/Brown) and testing on another (Brown/Wall Street Journal), and observed significantly lower performance.

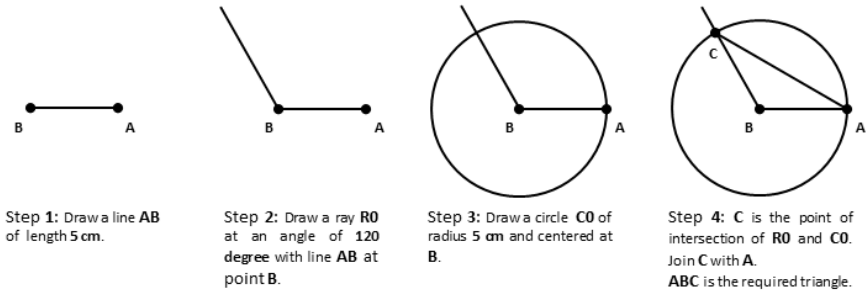


Fig. 1. Solution to the geometric construction problem discussed in text

In this paper, we evaluate off-the-shelf state-of-the-art parsers in developing NLU modules for GeoSynth – a system that solves word problems in geometric constructions. The NLU engine of GeoSynth uses parsing technology and domain ontology to understand high school geometry construction problems in natural language, and map it to a logical representation. These logical representations are then used for solving the problem either through pattern matching or program synthesis [5]. The steps of construction are presented as natural language output and its diagrammatic equivalents. Our evaluation shows that appropriate use of pre and post processing along with extensive domain knowledge can help overcome significant parser errors. In the next section we briefly describe the nature of geometry construction problems, followed by the GeoSynth system architecture in Section 4. The parsing experiment and results are presented in Section 5. Section 6 discusses main implications of the results and we conclude with some observations in Section 7.

2 Geometric Construction Problems

Cognitive development theories, for example, the van Hiel Framework [6, 7] for geometric thinking, have linked the progress of a student through different levels of mathematical reasoning with other cognitive skills. [8] argues that school students learn visual, verbal, drawing, logical and applied skills through geometry. The standards proposed by the US based National Council of Teachers for Mathematics also emphasizes geometric skills as a means for teaching and testing such other cognitive skills [9]. Therefore, geometry, and geometric construction in particular, is a vital component of any primary and high school curriculum.

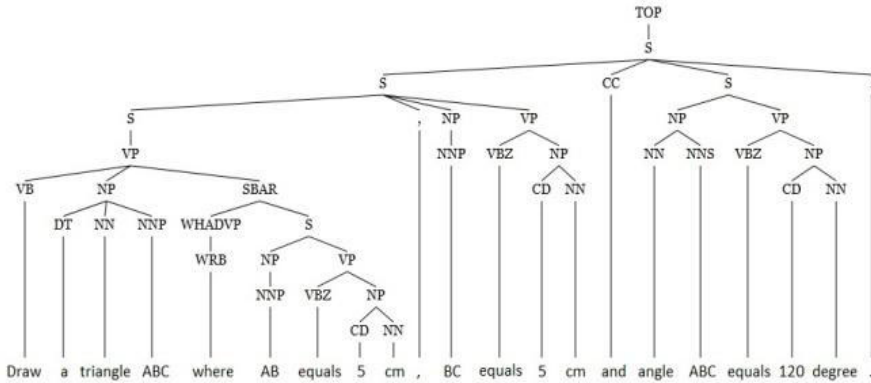


Fig. 2. Constituency Parse Tree for an example sentence generated by a parser

A typical geometric construction problem, such as “Draw a triangle ABC where $AB = 5\text{ cm}$, $BC = 5\text{ cm}$ and $\angle ABC = 120^\circ$ ”, has two main components. First there is a geometric entity, in this case “triangle ABC ”, that needs to be constructed. Second, a set of constraints that the entity must obey, such as “ $AB = 5\text{ cm}$ ” and “ $\angle ABC = 120^\circ$ ”. The solution to such a problem includes a diagram and a sequence of steps to construct the diagram. Figure 1 illustrates one possible solution for the above problem. Therefore, solving a geometric construction word problem, both by man and machine, involves two steps: (a) deduction of the entities to be constructed along with the associated constraints, and (b) coming up with a sequence of basic construction steps (using ruler and compass) that can generate the geometric entity described by (a). While the former calls for NLU, the latter requires, often a creative, application of geometric knowledge. Here, we shall discuss certain aspects of the former process.

Note that a geometric construction problem has four kinds of semantic/linguistic units: (1) *geometric entities* (e.g., “triangle”, “angle”), (2) *named entities* (e.g., “ ABC ”, “ AB ”) (3) *quantities* (e.g., “ 5 cm ”, “ 120° ”) and (4) *relationships* between the entities and quantities. Some of the common relationships are: **type-of** and **part-of** between two *geometric entities* (e.g., “triangle” is a **type-of** “polygon”, a “side” is a **part-of** a “polygon”); **has-name** between a *geometric entity* and a *named entity* (e.g., a particular “triangle” **has-name** “ ABC ”); **assignment** between a *geometric entity* and a *quantity* (e.g., “side AB ” is **assigned** “ 5 cm ”); other relations between *geometric entities* (e.g., *parallel*, *congruent*, etc.).

These relations are often implicit. For instance, in our running example, the facts that “one of the sides of triangle ABC ” **has-name** “ AB ”, or “side BC ” is **part-of** “triangle ABC ” are implicit. To complicate things further, the same geometric entity can be named variously (e.g., “triangle ABC ” can be referred to as “ BCA ”, “ CBA ”, etc.), and the same name can refer to multiple geometric entities (e.g., “triangle ABC ” and “angle ABC ”). Other well-known ambiguities associated with natural languages, such as scope of negation, prepositional phrase attachment and anaphora resolution are also present in geometric construction problems.

The aim of the NLU process is to analyse the natural language description of a problem and generate unique logical description, such as: (1) *triangle*(ABC)

(2) $triangle(ABC).angle(ABC).angleQ(B) = 120 \text{ degree}$

(3) $triangle(ABC).side(BC).lengthQ(l) = 5 \text{ cm}$

(4) $triangle(ABC).side(AB).lengthQ(l) = 5 \text{ cm}$

where, each line is a predicate: (1) represents the entity to be constructed, and (2), (3) and (4) represent the constraints.

3 GeoSynth: System Architecture

GeoSynth has four basic components: a *UI* for problem input and solution display, an *NLU* module that takes the problem statement and maps it to an equivalent logical representation, a *geometric construction solver* module that receives as input the logical representation and comes up with the steps of construction whenever possible, and a *natural language paraphrasing cum diagram generator* module that maps the solution from the solver module to a stepwise natural language description of the construction process and equivalent diagrammatic representations (e.g., see Figure 1). Due to paucity of space, here we will be able to discuss only the NLU and the solver modules in some details.

3.1 NLU Module

The input problem is first split into sentences using simple heuristics. The *constituency trees* for the normalized sentences are then generated by a parser (Figure 2 shows the constituency tree generated for our running example). The constituency tree is traversed bottom-up generating abstract geometric entities and constraints/relations between them by firing appropriate rules based on the *production rule* and lexical items associated with the nodes of the constituency tree. This process is similar to *syntax directed translation* used for code generation in compilers. Domain knowledge is extensively used to make inferences about implicit relations between the entities. Finally the partial logical forms generated from each sentence are merged by making use of the domain knowledge to obtain an equivalent and complete logical representation of the input problem statement.

3.1.1 Parsing

The intuition behind our approach is that if the constituency tree is correctly generated, then syntactic relations between the constituents can be efficiently exploited to arrive at the logical/semantic representation. Thus, the parsing module currently uses off-the-shelf English parsers like Stanford Parser [10] to generate parse trees. High error rates are expected from such a parser on geometry data because they have been trained on English sentences from newspapers reports, narratives and literature, which hardly contain any vocabulary from geometry. Domain adaptation for parsers is a possible solution in such cases, but requires a good amount of annotated data for sentences from the domain. This is often unavailable, as well as expensive and time-consuming to create. However, as it will be apparent from the following section, a high parsing accuracy is necessary for the NLU module to generate the relevant logical representations.

3.1.2 Ontology

Our system makes extensive use of domain ontology (Figure 3) that has been manually created. The domain ontology has *geometric entities*, *quantities* and

relations. Geometric entities are related by **type-of** (e.g., *square* is a **type-of** *rectangle*) and **part-of** (e.g., *side* is a **part-of** *triangle*) relations, they are associated with quantities (e.g., *side* has a *length*, length is measured in *in*, *cm*, etc.). In a **type-of** relation, the derived entity contains all the attributes of the base entity, plus it has its own attributes. Also, there are properties encoded in the Ontology which are of two types a) Initial - These are used to assign values to parameters of entities/quantities on creation. E.g., assigning “60” as value and “degree” as unit to the three *Angle* quantities of an *Equilateral Triangle* b) Triggered - The action of the property is performed on fulfilment of some prerequisite condition. E.g., when a *side* of a *Rectangle* entity is assigned a value (length), the *opposite side* of the *Rectangle* is assigned the same value automatically.

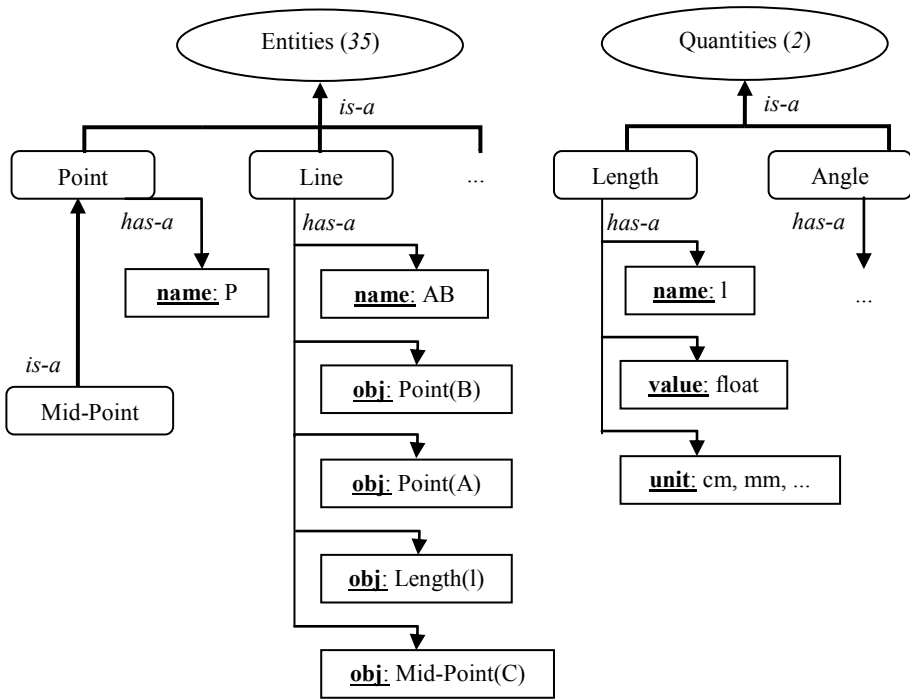


Fig. 3. Schematic of Ontology

Ontology also contains information about the standard naming conventions for geometric entities. For instance, a *triangle* is named by three capital letters, say *XYZ*, where *X*, *Y*, *Z* are the *vertices*, *XY*, *YZ*, *ZX* are the *sides*, and *YZX*, *ZXY*, ..., are alternative names of the same triangle. This rich knowledge of naming convention is extremely important for deducing the correct logical forms, and is a novel feature of our ontology.

3.1.3 Rules

We traverse the parse tree bottom-up and at every node of the tree a rule is fired based on the grammatical production rule and the actual lexical items associated with the

node. These rules, around a couple of hundreds, have been manually designed after studying the sentences in our development set on geometric construction problems. Nonetheless, most of these rules are not domain-specific.

For instance, for the production in Figure 2: $NP \rightarrow DT NN NNP$, which matches “*a triangle ABC*”, the following rule is fired:

1. Use ontology to identify if (a) the lexical item associated with NN, (e.g., “*triangle*”) is a valid *geometric entity*, and (b) the string associated with NE (i.e., “*ABC*”) is a valid *name* for the identified *geometric entity*.
2. If both of these conditions hold, create a new geometric entity of **type** *triangle* and **has-name** “*ABC*”.
3. Compare this entity against a *list* of geometric entities, and add to the *list* if and only if it is neither present in the list nor is it a **part-of** another entity already present in the list (e.g., if “*triangle ABC*” is already present in the list, then “*side AB*”, is added as a **part-of** “*triangle ABC*”).
4. Create an empty *list* of entities at the beginning of the process and update as stated above.

Similarly, for the right most production in the parse tree of Figure 2: $S \rightarrow NP VP$, the NP node will have an *Angle* entity with name ABC (got by applying the rule for $NP \rightarrow NN NNS$) and the VP node will have an *Angle* quantity (by applying the rule for $NP \rightarrow CD NN$ in it) and a dangling assignment of it due to the presence of *equals* (on execution of the rule $VP \rightarrow VBZ NP$). On applying the rule for this production ($S \rightarrow NP VP$), the *Angle* quantity will be assigned to the quantity *Angle* of *Angle* ABC entity. It can be seen from the figure that the parse tree generated by the parser is incorrect and this production ($S \rightarrow NP VP$) should have come inside the SBAR so as to relate to the *Triangle* entity created. Nonetheless, this issue is resolved at the topmost production $TOP \rightarrow S$, where a sanity check on the list of entities and quantities is performed, our system is able to figure that the *Angle* entity generated in this production with name ABC is a part of the *Triangle* entity with name ABC got from the left most production due to the presence of our Ontology and hence, does the corresponding assignment.

3.2 Solver

Given the logical representation of the entities to be constructed and their inter-relations, we take two different and complementary approaches to arrive at the steps of construction: (1) Program Synthesis and (2) Pattern Matching. Program synthesis [5] uses a library of around 20 primitive ruler and compass construction steps and synthesizes the solution as a sequence of combination of these steps which can generate the required entity. This is achieved by using a goal directed heuristic search.

In pattern matching, we maintain a database of patterns for different types of construction problems commonly taught in school (e.g., drawing a triangle given three sides (SSS), a triangle given 2 sides and 1 included angle (SAS) etc.) along with their solutions in the form of the steps of construction. Given a logical representation, we search the database of patterns to find an exact match, and if found, the solution is generated by appropriately modifying the names and quantities present in the solution pattern in the database.

While in principle, program synthesis can solve any geometric problem and pattern matching can solve only those problems for which the patterns exist in the

database. However, the output generated by program synthesis may or may not be the best or the desired way to solve a problem. The pattern matching approach provides a teacher with the flexibility to control the solution method and the granularity of the steps according to the prescribed syllabus or the level at which the solution is being taught and tested.

4 Experiment and Results

In this section we describe the evaluation experiment and the results, using three state-of-the-art English parsers for GeoSynth. For our experiments, we used 3 different state-of-the-art parsers - Charniak's parser [3], Stanford parser [10] and Collins's statistical parser [11] which was improved upon by Bikel [12].

All three parsers use statistical models trained on large amount of mainly news data. Charniak's Parser uses a context-free chart-parsing to generate the best probable parse tree for a sentence. The version used here [13] uses Markov models to generate and rank probable parses. Collin-Bikel Parser uses dynamic programming to generate parse probabilities in terms of dependency relations between head-words in a parse tree. The Stanford Parser also uses dynamic programming and context-free grammars to generate all possible parses for a given sentence, selecting the one with the highest probability.

4.1 Baseline

Our development set for the baseline consisted of 61 sentences from 39 textbook problems taken from online NCERT CBSE text books¹ on geometric construction. We tested our system on the test set comprising of 82 problems from 8th and 9th grade taken from [14] and [15] respectively. Note that the ontology, the pattern database and the rules have been constructed based on our observations on the development set.

Table 1. Parser accuracy on original sentences without text normalization

Parser	Parse Tree (%)			Logical Form (Recall / Precision %)		
	POS	LR/LP	Sentence level	Geometric Entity	Named Entity	Relation
Charniak	64.86	57.10/64.26	0	14.15/100	13.64/100	2.80/100
Stanford	69.83	53.37/53.63	2.15	13.20/96.55	6.06/100	1.60/75
Collins/ Bikel's	65.71	20.75/24.21	0	25/100	29.55/100	1.73/81.25

All the three parsers were evaluated on the test set against the gold standard created semi-automatically, for: a) the part-of-speech (POS) tagging accuracy, b) bracketing performance using *evalb* [16], and c) sentence level parsing accuracy, that is, the number of sentences which had a correct parse tree. We also noted the

¹ <http://ncertbooks.prashanthellina.com>

Precision/Recall accuracies of the logical forms, viz., Geometric entities, Named entities and Relations, generated by our system for the test set problems using the constituency trees generated by the three parsers. The results of this evaluation are tabulated in Table 1. It can be seen that all three parsers have a bracketing recall and POS tagging accuracy on geometry problems, of less than 58% and between 60-70% respectively. This is significantly lower than the reported accuracies on English literature of greater than 87% for bracketing recall and 92% for POS tagging [2]. The bracketing performance is extremely low with less than 3% of sentences having correct parse trees (0% for Charniak and Collins/Bikel's).

Hence, the Recall of geometric entities, named entities and relations is also very low for all 3 parsers with the Precision for relations dropping to 75-85% due to erroneous adjunct attachments for Stanford and Collins/Bikel's. A manual inspection revealed that the parsing errors were mainly due to a) incorrect named entity recognition - failing to mark the names of geometric entities, such as PQ, ABC, as proper nouns, b) prepositional phrase attachment and c) coordination ambiguities.

Table 2. Lower Cases

Parser	Parse Tree (%)			Logical Form (Recall / Precision %)		
	POS	LR/LP	Sentence level	Geometric Entity	Named Entity	Relation
Charniak	64.86	57.10/64.26	0	14.15/100	13.64/100	2.80/100
Stanford	76.17	66.77/62.45	8.60	29.25/98.41	9.85/100	4.13/88.57
Collins/Bikel's	73.26	19.78/23.32	0	40.57/98.85	23.86/100	2.13/100

Table 3. Lower Cases and Symbols

Parser	Parse Tree (%)			Logical Form (Recall / Precision %)		
	POS	LR/LP	Sentence level	Geometric Entity	Named Entity	Relation
Charniak	76.46	84.33/79.47	0	17.45/100	40.53/100	6.26/92.16
Stanford	87.31	69.89/68.63	8.60	24.06/100	45.45/100	17.84/91.16
Collins/Bikel's	81.60	26.38/27.11	0	45.75/98.98	28.41/100	9.45/98.61

For instance, in Figure 2 “and” has been incorrectly attached to the top *S*, where it should have been attached to *SBAR*.

4.2 Pre-processing

To address the high error rate in the baseline, we designed certain heuristics that modify the input sentence before sending it to the parser.

Lower and Upper Cases: We observed that Stanford and Collins/Bikel's performance increased significantly when proper cases were converted to lower cases (Ex: Draw \rightarrow draw) as seen in Table 2. Text-normalization: Also, since the parsers were not trained on geometry specific data, normalizing the text by replacing the symbols (e.g., =, \circ , \angle) by corresponding word forms ("equals", "degree" and "angle" respectively) greatly increased the POS tagging and bracketing accuracy, and the corresponding named entity and relation Recall/Precision (Table 3).

Pre-processing for POS labels: We also observed that the parser makes several errors in identifying adjectives, nouns, verbs and named entities specific to the geometry domain. This is the case even though these words are rather unambiguously used in geometry. For instance, "circle" or "square" are always used as nouns and not verbs. To resolve these issues we used a pre-processing step where we handcrafted a list of common geometric entities including multiword expressions such as "right angled triangle". These patterns were identified in the input text and replaced by more common and unambiguous English nouns such as "elephant" or "movie". As shown in Table 4, the sentence level accuracy along with Recall/Precision for Geometric entities and Relations improves greatly after this pre-processing step. Similarly, we designed regular expressions to identify named entities (e.g., *ABC* and *PQ*) and replace them by more common English named entities (e.g., *John* or *Mary*). Table 5 shows the results after this step.

Table 4. Lower Cases, Symbols and Geometric Entities

Parser	Parse Tree (%)			Logical Form (Recall / Precision %)		
	POS	LR/LP	Sentence level	Geometric Entity	Named Entity	Relation
Charniak	82.57	84.94/77.36	5.38	39.15/98.81	39.02/98.10	17.04/95.52
Stanford	91.77	71.36/68.67	13.98	39.15/100	52.27/100	35.02/94.95
Collins/ Bikel's	85.43	31.46/30.02	3.22	58.02/100	35.61/100	14.78/98.23

While the accuracy of the parser is significantly better on these modified sentences and the Precision/Recall for logical forms generated is highest on application of all of these heuristics the results are still lower than expected for the parsers [2]. Since, after the application of these pre-processing steps the geometric construction problems resembled the English sentences on which the parsers were trained on, we decided to proceed with the given performance and try to overcome the majority of the remaining errors with the help of Ontology to correctly deduce the logical representations.

4.3 Domain Knowledge and Ontology

Out of the 41 problems for which the system was able to generate the perfect logical representations, in all the 41 cases for original problems and 35 cases for the modified problems, the parse trees generated had at least one error, and often several and serious errors due to which the NLU module would generate incomplete/incorrect

logical representations rendering the solver incapable of solving the problems. However, use of domain knowledge during semantic interpretation of the constituent tree has helped us circumvent those errors in an effective way. For instance, in our running example, the parse tree as shown in Figure 2 is incorrect. According to the generated parse tree, *side BC* and *angle ABC* seem independent entities unrelated to *triangle ABC*. Nevertheless, during inference using the Ontology, our algorithm discovers that *BC* and *ABC*, by virtue of their names, are **part-of** *triangle ABC*. Thus, the missed syntactic connection is recovered at a semantic level during the logical deduction process. While it might not be possible to recover from any arbitrary parser error using ontology, the proposed method is effective only because the current off-the-shelf parsers have high accuracy in identifying local relations.

Table 5. Lower Cases, Symbols, Geometric Entities and Named Entities

Parser	Parse Tree (%)			Logical Form (Recall / Precision %)		
	POS	LR/LP	Sentence level	Geometric Entity	Named Entity	Relation
Charniak	89.43	83.11/72.47	7.53	51.89/99.10	40.53/100	24.37/91.96
Stanford	98.46	67.44/62.90	15.05	57.55/100	68.18/100	54.59/96.47
Collins/ Bikel's	93.37	34.46/28.95	4.30	68.40/97.97	53.41/100	18.51/92.05
Gold Standard	100	100/100	100	67/100	86.36/99.56	75.9/98.96

Table 6. Sentence Length vs. Accuracy after applying all modifications of lower cases, symbols, geometric entities and named entities

Sentence Length	Charniak (%)			Stanford (%)			Collins/Bikels' (%)		
	POS	LR/LP	Sent	POS	LR/LP	Sent	POS	LR/LP	Sent
1-8 (10.75 %)	95.89	90.63/ 92.06	60	97.26	90.62/ 92.06	70	95.89	59.36/ 58.46	40
9-16 (20.43 %)	87.07	75.96/ 75.24	5.26	97.84	82.21/ 81.43	37	92.24	28.85/ 28.30	0
17-24 (49.46 %)	89.41	85.90/ 72.77	0	98.61	59.77/ 55.24	0	93.58	34.22/ 27.31	0
25-32 (19.35 %)	89.61	80.28/ 68.80	0	98.63	71.63/ 65.56	0	93.14	34.00/ 28.94	0

5 Discussion

The labelled accuracy for the best parsers today (precision/recall of the order of 90%), drops drastically as the sentence length increases [2]. In our development and test sets, on an average a sentence had 15 and 16.9 words, respectively. Thus, the expected labelled precision/recall is between 70% and 80% [2], which implies that

almost every sentence has 3 to 4 errors. Indeed, as seen in Table 6, the sentence level parsing accuracy, along with POS tagging accuracy and bracketing Recall/Precision, drops with increase in sentence length. When the number of words in a sentence were between 1-8, the constituency tree for 40%, 60% and 70% of the sentences (sentence level accuracy) were generated accurately for Collins/Bikels', Charniak and Stanford respectively. On moving to the next bucket of 9-16 words per sentence, we observed that (Table 6) the sentence level accuracy fell greatly to 0%, 5.26% and 37% for Collins/Bikels', Charniak and Stanford respectively.

Table 7. F-Score for important phrases before any modification and after all modifications

Parser	No Modification				All Modifications			
	NP	VP	PP	ADJP	NP	VP	PP	ADJP
Charniak	0.61	0.57	0.58	0.35	0.79	0.92	0.74	0.25
Stanford	0.58	0.52	0.63	0.38	0.74	0.62	0.81	0.57
Collins/Bikels'	0.32	0.07	0.18	0	0.38	0.16	0.38	0

Table 8. Accuracies (%) for important POS before any and after all modifications

Parser	No Modification				All Modifications			
	NNP	NN	VB	JJ	NNP	NN	VB	JJ
Charniak	72.71	72.79	95.92	20.55	100	94.77	96	95.45
Stanford	79.58	79.24	100	13.73	99.62	97.19	100	100
Collins/Bikels'	72.62	87.26	93.33	53.33	96.97	97.62	97.94	100

Hempelmann et al. [2] also claim in their study that sentence length is the single most important factor that affects the robustness and the performance of parsers in educational/learning technology. Their results show no effect of genre or domain on parser performance independent of sentence length and certain common syntactic errors such as adjunct attachment, across the domains. While our results agree with [2] in that sentence length does effect parser accuracies over and above other factors, domain-specific vocabulary and structure plays a very important role in determining how well a parser works in a particular domain. In fact, our motivation for using certain heuristics as well as domain ontology for improving accuracies lies in the fact that domain-specific terminology hinders generation of correct parse-trees by general statistical parsers. From Table 7 and Table 8, it can be seen that the modifications we propose greatly help in improving the F-Score for bracketing performance and the accuracy of POS tagging respectively for some of the most important Phrase/POS required for solving a geometric problem.

6 Conclusion

In this work we explored the benefits of using state-of-the-art parsing technology in developing NLU components for domain specific tutoring systems. This study leads us to the following key observations:

(i) Accuracy of off-the-shelf English parsers falls drastically for long sentences and texts that use jargons and specialized vocabulary.

(ii) Domain adaptation of parsers requires good amount of annotated data, which is often unavailable and expensive to create.

(iii) However, simple heuristic based pre-processing of text that identifies domain specific vocabularies and replace them by more common words from the same lexical category can significantly improve the parser accuracies (in our experiments, we observe threefold improvement).

(iv) Furthermore, wide-coverage and in-depth domain ontology can be effectively used for extracting correct semantic relations even from incorrect parse trees.

This, in fact, parallels human cognitive process, where for interpretation of highly technical texts, domain knowledge is not only necessary, but also often can help circumvent mild general linguistic incompetence. Thus, we believe that one can appropriately reap the benefits of advancements in parsing technology for building applications in education, provided efforts are made to create extensive domain ontology.

Acknowledgement. We are grateful to Sumit Gulwani, Microsoft Research Redmond for providing us the program synthesis engine for solving geometric construction problems and for active participation in discussions during this work.

References

1. Mukherjee, Garain, U.: A review of methods for automatic understanding of natural language mathematical problems. *Artificial Intelligence Review* 29, 93–122 (2008)
2. Hempelmann, C.F., Rus, V., Graesser, A.C., McNamara, D.S.: Evaluating state-of-the-art Treebank-style parsers for Coh-Metrix and other learning technology environments. In: *Proc. of the 2nd Workshop on Building Educational Applications Using NLP*, pp. 69–76 (2005)
3. Charniak, E.: Statistical parsing with a context-free grammar and word statistics. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. AAAI Press/MIT Press, Menlo Park (1997)
4. Gildea, D.: Corpus Variation and Parser Performance. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Pittsburgh, PA (2001)
5. Gulwani, S., Korthikanti, V., Tiwari, A.: Synthesizing geometry constructions. In: *PLDI* (2011)
6. Senk, S.L.: Van Hiele levels and achievement in writing geometry proofs. *Journal for Research in Mathematics Education* 20(3), 309–321 (1989)
7. Hoffer, A.: Van Hiele-based research. In: Lesh, R., Landau, M. (eds.) *Acquisition of Mathematics Concepts and Processes*, pp. 205–227. Academic Press, New York (1983)
8. Hoffer, A.: Geometry is more than proof. *Mathematics Teacher* 74(1), 11–18 (1981)
9. National Council of Teachers of Mathematics: *Principles and standards for school mathematics*, Reston, VA (2000)
10. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp. 423–430 (2003)

11. Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proc. of the 35th Annual Meeting of the Association for Computational Linguistic, Madrid, Spain (1997)
12. Bikel, D.M.: Intricacies of Collins' parsing model. *Computational Linguistics* 30(4), 479–511 (2004)
13. Charniak, E.: A Maximum-Entropy-inspired parser. In: Proceedings of the North-American Chapter of Association for Computational Linguistics, Seattle, WA (2000)
14. Aggarwal, R.S.: Exercise: 36-B to 36-E. *Foundation Mathematics for Class 8* (2009)
15. Aggarwal, R.S.: Exercise: 10-D, 15-B. *Foundation Mathematics for Class 9* (2008)
16. Sekine, S., Collins, M.J.: Evalb. Tool for evaluating bracketing performance of a parser (2005), <http://nlp.cs.nyu.edu/evalb/> (accessed April 2011)

Exploring Classification Concept Drift on a Large News Text Corpus

Artur Šilić and Bojana Dalbelo Bašić

University of Zagreb,
Faculty of Electrical Engineering and Computing,
Unska 3, 10000 Zagreb, Croatia

Abstract. Concept drift has regained research interest during recent years as many applications use data sources that are changing over time. We study the classification task using logistic regression on a large news collection of 248K texts during a period of seven years. We present extrinsic methods of concept drift detection and quantification using training set formation with different windowing techniques. We characterize concept drift on a seven-year-long Le Monde news corpus and show the overestimation of classifier performance if it is neglected. We lay out paths for future work where we plan to refine extrinsic characterization methods and investigate the drifting of learning parameters when few examples are available.

Keywords: text classification, concept drift, logistic regression.

1 Introduction

Change of concept due to passing of time is called concept drift (CD). This ubiquitous phenomenon has regained research interest during recent years as many applications that rely on learning methods use data sources that are inherently changing over time. Such applications are used for monitoring and control, assistance, decision making, and robotics [19].

In machine learning, classifiers are constructed to discriminate examples belonging to a predefined concept. If the distribution of examples changes after the learning phase, negative effects might occur. First, the true error rate of a classifier might change substantially and it might not be well estimated during learning phase. Second, if assumptions about changes of the distribution are known, but unavailable to the learning algorithm, it might perform suboptimally. Research of CD is motivated with the goal of averting these two deficiencies.

In the area of text classification, research of CD is beneficial to applications of information filtering and organization. An information filtering system presents texts to users while their interests change over time. For example, the users might follow news stories, various documents, web pages, or simply wish to receive non-spam e-mails. On the other hand, an information organization system recognizes all texts of a given time-evolving topic while assuming that all information necessary for this task is contained within the texts. An information

organization system without the ability to handle CD may have significantly worse performance on time-changing data.

Substantial number of works contain both theoretical analysis and practical method design to handle the changes in data streams. Methods have been devised that solve detection, characterization, quantification, and modelling of CD. Also, they enable the construction of adaptive learners. Current publications lack a quantification of CD in large news corpora on a text classification task.

We present a study on a completely realistic data set where we show how much classifier performance is overestimated if CD is neglected. We have observed that a drift can be coarsely and extrinsically characterized by comparing the classifier performance using training set formation with different windowing techniques. Also, we have shown that if a dataset contains a concept drift that is neglected, then an overestimation of classifier performance may occur.

The work is structured as follows. In Section 2 related work is presented. In Section 3 experiment settings are explained and the results are discussed in Section 4. The article is concluded with Section 5.

2 Related Work

An excellent overview on CD research from Zliobaite [19] covers all of its aspects: definition, methods, applications, and fields of research. A shorter overview from Tsymbol is also available [16].

Most of existing research on CD in the domain of text classification concentrates on user interests. For example, in [3-5, 14, 17] the authors study simulated user interest changes in news. In [3], a Naive Bayes classifier is employed while time-based selection and weighting is applied to features. The authors use a 6K article subset of the 20 Newsgroups corpus. In the same work, drifting spam filtering was examined on the SpamAssassin collection of 9K emails. In [4, 5, 14], the authors use a subset of 2608 documents from the TREC corpus of English Business News, with simulated scenarios where the interest shifts from one topic to another. In [4, 5], a methodology very similar to ours is used. While we use logistic regression and time-based example selection, they combine SVM learning with time-based example selection and weighting in order to produce a classifier that better handles CD. In [14], a specific ensemble algorithm is constructed. In [17], a framework of wider purpose is designed that enables an existing algorithm to learn CD from fewer labeled data. They use the Reuters news corpus to simulate user interest changes. Lang [6] studies real user interests in news collected over a one-year period, but does not deal with CD. In contrast to user interests studied in above-mentioned works, we analyze intrinsic content change of news categories.

Concerning drifting text content classification, Liu and Lu [8] present an adaptive classifier based on updating feature weights and perform experiments on

web documents. Lebanon [7] models local likelihood of word appearance on a one-year corpus from Reuters (RCV1). Likewise, Forman [2] works on the RCV1 and addresses the Daily Classification Task assuming that only a portion of text documents are labeled each day, while the rest of the documents are unlabeled and require automatic labeling. To leverage the value of past data, Forman combines SVM with temporal inductive transfer (TIX). In essence, the TIX consists of augmenting each feature vector with classifications of past classifiers. In contrast, we work in a setting where the data is labeled only until a certain point in time and we use only the original features.

Close to our work is the research of [9, 11, 12] where CD of predefined text categories is analyzed on large collections of scientific (ACM Digital Library contains 30K articles over 23 years) and medical articles (Medline contains 861K articles over 16 years). In [9], temporal aspects of document collections are studied. In [11], the authors describe a strategy for example selection that handles CD. In [12], temporal weighting of examples and classification scores is incorporated to Rocchio, k-NN, and Naive Bayes learning algorithms.

In contrast to designing methods that intrinsically or extrinsically handle CD, some authors try to measure it. For example, Yeon et al. [18] introduce a measure of CD, which is defined as the angle between the estimated and the optimal weight vector obtained under no CD.

The conclusions of this work stress the importance of appropriate sampling, a well-known *modus operandi* in the domain of statistics. For example, Rakotomalala et al. [10] show that the sampling scheme that creates the learning set has to be accounted when building a predictive model.

3 Experiment Settings

3.1 Data Set and Preprocessing

We take news articles issued in the French daily Le Monde during seven years – from January 1995 to December 2001. Only the seven most numerous categories are considered: Culture (ART), Enterprise (ENT), France (FRA), Horizons (HOR), International (INT), Society (SOC), and Sport (SPO). Our corpus constructed in this way counts 248,581 news articles.

The very popular and the very basic text representation was used – the *bag-of-words* Vector Space Model [13]. Text was lowercased, and article structure was omitted resulting in all of its parts (title, lead, and body) being put in a single text chunk. Words were normalized using a version of Porter’s stemmer for French.¹ In order to reduce computational complexity, feature selection was employed where only features that appear in at least 10 different articles were retained. Such a simple feature selection is justified with the consensus that regularized methods do not need any advanced feature selection methods.

¹ <http://snowball.tartarus.org/algorithms/french/stemmer.html>, acc. 01/2012

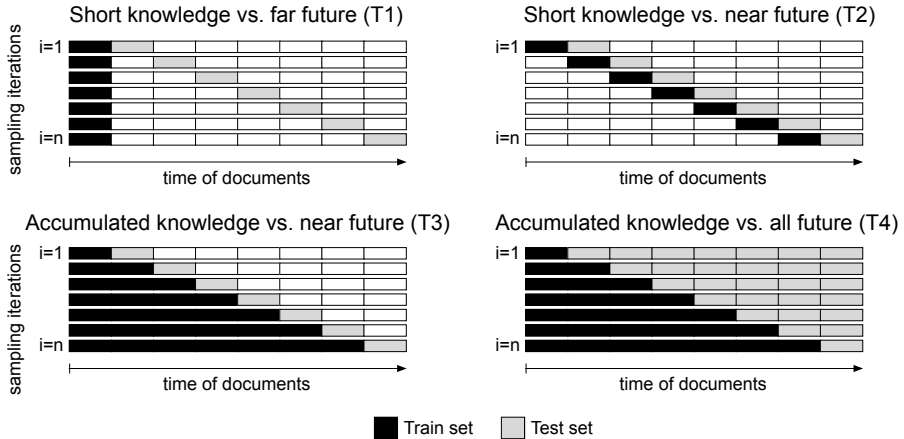


Fig. 1. Temporal sampling methods

3.2 Classification Algorithm

As a classification algorithm we used logistic regression implemented in the LIBLINEAR library [1]. This choice was motivated with method’s proven classification performance [2] and computational efficiency. The linear binary classifier is derived by solving the following unconstrained optimization problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) \equiv \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

Where \mathbf{x}_i is the i -th example and $y_i \in \{-1, 1\}$ is its classification label. Example \mathbf{x} is deemed having a positive classification $y = 1$ only when $\mathbf{w}^T \mathbf{x} > 0$. A single penalty parameter $C > 0$ controls the trade-off between margin width and accuracy on the training set. In our experiments, C is chosen from $\{0.0625, 0.125, 0.25, 0.5, 1, 4, 16, 64\}$. Classifiers were evaluated using the standard F1 measure, which was micro- and macro-averaged when calculated for many categories at once [15].

3.3 Temporal Sampling

Here we describe sampling methods that we used in experiments, which are based on timestamps of the news articles. The sampling methods are used to simulate classifiers that differently handle CD and to evaluate their performance. All sampling methods are illustrated on Fig. 1.

² See ICML’08 Workshop – PASCAL Large Scale Learning Challenge, <http://largescale.ml.tu-berlin.de/workshop/>, acc. 01/2012.

The first sampling method is the *short knowledge vs. far future* (T1), which anticipates that a classifier will be trained on the first interval (block of documents) and then utilized on the rest of the stream.

The second sampling method is the *short knowledge vs. near future* (T2)³ which anticipates that for each interval a classifier will be trained and then utilized only on the subsequent interval. By assuming an existence of a gradual CD, T2 should generally give better performance in comparison to T1 because near intervals tend to be of higher similarity in comparison to those further away. This assumption does not hold for seasonal content.

The third sampling method is the *accumulated knowledge vs. near future* (T3)⁴ If a gradual CD is not too strong or the target drifting concept has a constant subconcept, then a classifier trained with T3 sampling instead of T2 sampling is expected to have better performance due to more training data.

The fourth sampling method, *accumulated knowledge vs. all future* (T4), is motivated with classifier evaluation. T4 assumes that we train a classifier on all available intervals until a single time point, and test it on all subsequent intervals. T4 sampling is important for evaluation because it simulates a realistic situation when the classifier is unchanged after the initial training, exactly the way many classifiers are used in everyday applications.

According to [19], we employed the simplest possible CD learner based on training set formation using a window technique. As Rocha notes [11], varying the temporal window for the train set settles a tradeoff between two opposing effects. While increasing the window size too much might introduce texts that are out of the temporal context, decreasing the window size too much could result in too little data and overfitting might occur. In our case, these two extremes are analogous to situations where we employ T3 and T2 samplings, respectively. On the other hand, T1 and T4 samplings are not equal to any specific CD handling methods; they are rather created in order to show the effect on classifier performance (Q) if drift is not treated – either with specific methods or with simple retraining strategies. Whether $Q_{T3} > Q_{T2}$ holds, clearly depends on the type of CD present in the data stream – for example, in [5], two of three scenarios exhibit the opposite case ($Q_{T2} > Q_{T3}$), since the simulated CD is abrupt and very strong.

4 Results

4.1 Preliminary Experiments

The classification models were assessed using a simple *train vs. test* setting without validation. This was done in order to determine the theoretical maximum of classifier performance and its sensitivity to the single parameter C .

For start, T1 sampling was used suspecting that a CD exists in the given corpus. The classifiers were trained on first two months of the corpus, and then

³ T2 is equal to *no memory* sampling from [5].

⁴ T3 is equal to *full memory* sampling from [5].

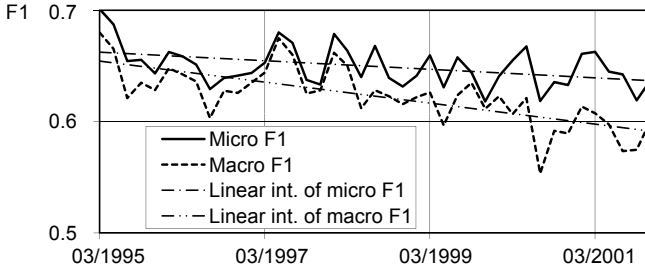


Fig. 2. Micro- and macro-averaged F1 classifier performance with T1 sampling trained on Jan-Feb 1995 over time; values on x-axis mark the beginnings of two-month intervals; the performance curves were also linearly interpolated to show a declining trend

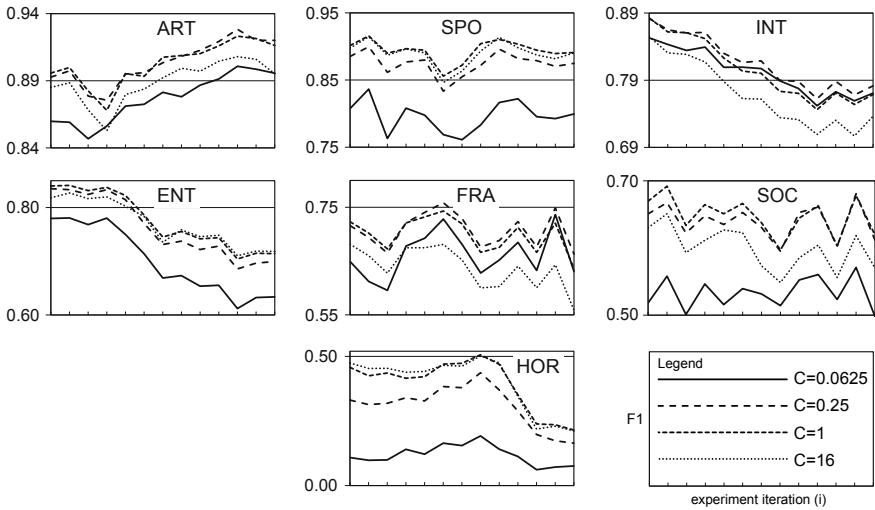


Fig. 3. F1 classifier performance with T1 sampling trained on Jan-Jun 1995 over time for each category and each choice of C ; ticks on the x-axis mark the beginnings of six-month intervals

tested on the remaining two-month intervals. With the exception of ART, all categories follow a performance decline confirming the existence of a CD, see Fig. 2.

Next, we divided the collection into six-month intervals and tested the impact of parameter C on classification performance, also using the T1 sampling. Again, on Fig. 3 we see that the most of the curves have a declining tendency over time. On Fig. 4 average performance is plotted against the choice of C for each category. Performance is greatly affected by the learning parameter C , but a mitigating circumstance is in that the optimal C from the first interval stays very near the optimum in all other intervals. Also, C is very stable in the neighborhood of its optimum, see Fig. 4. This means that the prediction of optimal C is going

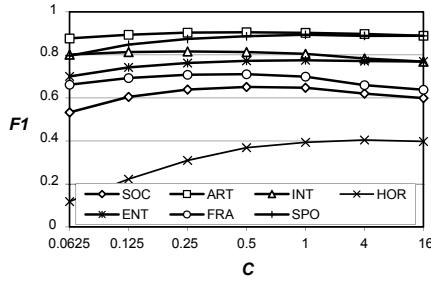


Fig. 4. Average F1 classifier performance with T1 sampling trained on Jan-Jun 1995, and tested on the remaining six-month intervals for each category depending on the parameter C

to be sufficiently accurate when inducing from the first interval. The described stability of C was observed for T2 and T3 samplings as well.

Another detail to be noted is that on Fig. 3 we see that classifiers on ART and SPO categories achieve relatively better results – around 90%, yet these are the only categories that do not exhibit a performance degradation with T1 sampling as other categories do. Further investigation of this observation is required.

4.2 Comparison of T1, T2, and T3 Samplings

We compared classifier performance with T1, T2, and T3 samplings, using the optimal C for each category. On Fig. 5 we see that for all intervals $Q_{T3} > Q_{T2}$ and $Q_{T3} > Q_{T1}$. The reason for this is because T3 has access to a larger number of past articles and because the drift is not too strong. Also, on most of the categories (HOR, INT, SPO, FRA, and ENT) we can see that the $Q_{T2} > Q_{T1}$. On these categories, the assumption that two temporally closer intervals are more related is correct.

Based on these results and according to the taxonomy of [19] we classify the CD present in our Le Monde corpus as a mild *incremental* drift in contrast to *sudden* or *reoccurring* drift types. We assume that the discussed drift is not *gradual* since a news text collection is comprised of very different events that have very distant texts.

4.3 Comparison of T4 and Random Samplings

T4 sampling is compared with random sampling on equally-sized training and test sets. For each iteration on a single category, 5-fold cross-validation was used to optimize the C parameter on training set and then the chosen parameter was used to train the classifier, which was tested on the test set comprised only of the remaining set of articles.

On Fig. 6 we see that, both for micro- and macro-averaging, random sampling overestimates by 3-5% in terms of the F1 measure. More detail is available on Fig. 7, where we see that random sampling gives a too optimistic performance

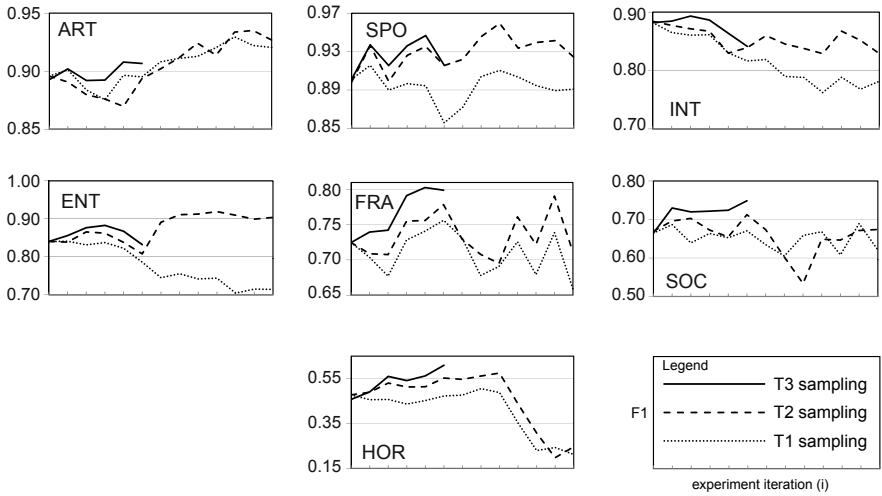


Fig. 5. Comparison of F1 classifier performance over intervals with T1, T2, and T3 sampling methods; starting interval for T1 sampling is Jan-Jun 1995; ticks on x-axis mark the beginnings of six-month intervals

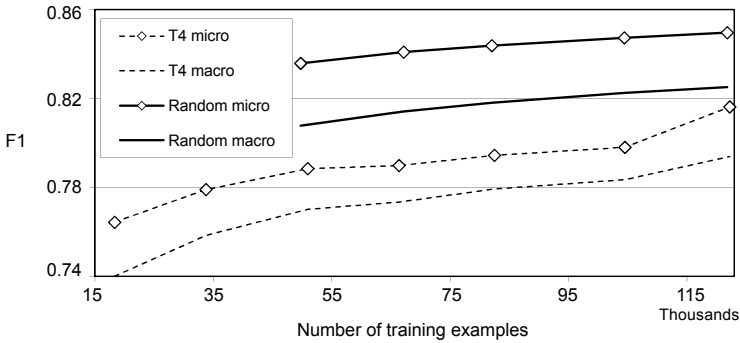


Fig. 6. Micro- and macro-averaged F1 classifier performance; random vs. T4 sampling for all categories; x-axis marks the size of the training set, while the rest is used for testing

estimation for each category. Comparing the random and T4 performance estimation, we see that on categories of ART and SPO differences are smaller than the ones on categories of ENT (max. 12%) and INT (max. 7%). This correlates with the result noted in Section 4.1, where INT and ENT performance plot with T1 was characterized with clearly falling curves, while ART and SPO had rising and constant curves. The working hypothesis is that if the T1 curve is falling more intensively, the drift is stronger and the overestimation of random sampling will be higher.

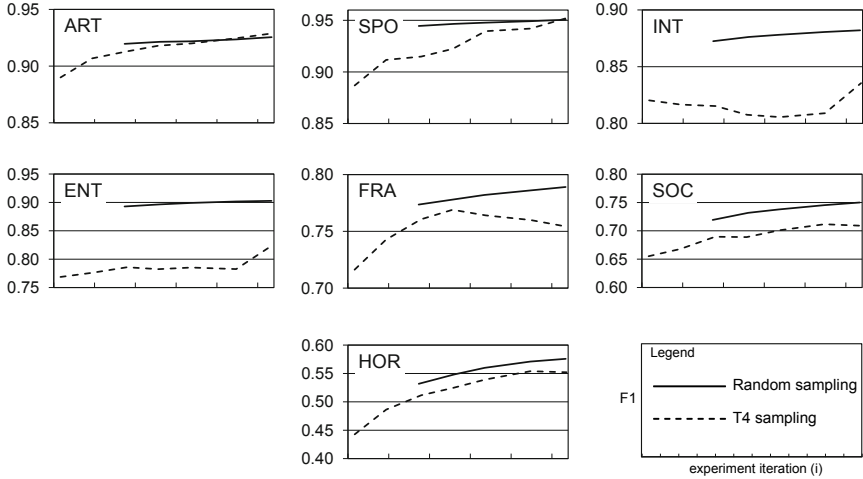


Fig. 7. Comparison of F1 classifier performance with random vs. T4 sampling for each category; x-axis marks the size of the training set, while the rest is used for testing

5 Conclusion

We have presented extrinsic methods of concept drift detection and quantification using training set formation with different windowing techniques. We have observed that a drift can coarsely be characterized by comparing the classifier performance using the described training vs. test set samplings. Using this methodology, we characterize broad categories of our Le Monde corpus as having an incremental and mild concept drift.

In the general case, we have shown that neglecting concept drift in data streams and using random sampling may lead to overestimation of classifier performance. If concept drift is not treated and if the time stamps of examples are available, they should be taken into account when constructing the training and test sets for classifier evaluation. Specifically, with respect to a chosen time point, training set should encompass only past data, while the test set should encompass only future data. In a realistic setting, by employing the logistic regression on a seven-year-long Le Monde news corpus, we observed that random sampling overestimates F1 about 3-5% on average.

Although the optimal learning parameter C was stable, future work will investigate the drift of C , which is suspected to exist in scenarios with fewer examples. Also, an analysis of a corpus with more categories will enable to test the strength of positive correlation between the decline of T1 performance curve and overestimation of classifier performance due to random sampling.

Acknowledgement. This research has been supported by the Croatian Ministry of Science, Education and Sports under the grant No. 036-1300646-1986. The authors would like to thank Stéphane Huet for preparing the Le Monde corpus.

References

1. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* 9, 1871–1874 (2008)
2. Forman, G.: Tackling concept drift by temporal inductive transfer. Technical Report HPL-2006-20R1, Hewlett Packard Laboratories (2006)
3. Katakis, I., Tsoumakas, G., Banos, E., Bassiliades, N., Vlahavas, I.P.: An adaptive personalized news dissemination system. *J. Intell. Inf. Syst.* 32(2), 191–212 (2009)
4. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.* 8(3), 281–300 (2004)
5. Klinkenberg, R., Rüping, S.: Concept drift and the importance of examples. In: *Text Mining – Theoretical Aspects and Applications*, pp. 55–78. Physica-Verlag (2003)
6. Lang, K.: Newsweeder: Learning to filter netnews. In: *Proc. 12th ICML*, pp. 331–339 (1995)
7. Lebanon, G., Zhao, Y.: Local likelihood modeling of temporal text streams. In: *Proc. 25th ICML*, pp. 552–559. ACM (2008)
8. Liu, R.-L., Lu, Y.-L.: Incremental context mining for adaptive document classification. In: *Proc. 8th KDD*, pp. 599–604. ACM (2002)
9. Mourão, F., da Rocha, L.C., Araújo, R.B., Couto, T., Gonçalves, M.A., Meira Jr., W.: Understanding temporal aspects in document classification. In: *WSDM*, pp. 159–170. ACM (2008)
10. Rakotomalala, R., Chauchat, J.-H., Pellegrino, F.: Accuracy estimation with clustered dataset. In: *Proc. 5th AusDM*, pp. 17–22. Australian Comp. Soc. (2006)
11. Rocha, L., Mourão, F., Pereira, A., Gonçalves, M.A., Meira Jr., W.: Exploiting temporal context in text classification. In: *Proc. 17th Conf. Information and Knowledge Management*. ACM (2008)
12. Salles, T., da Rocha, L.C., Pappa, G.L., Mourão, F., Meira Jr., W., Gonçalves, M.A.: Temporally-aware algorithms for document classification. In: *Proc. 33rd SIGIR*, pp. 307–314. ACM (2010)
13. Salton, G., Wong, A., Yang, A.C.S.: A vector space model for automatic indexing. *Communications of the ACM* 18, 229–237 (1975)
14. Scholz, M., Klinkenberg, R.: An ensemble classifier for drifting concepts. In: *Proc. 2nd Int. Wksh. on Knowledge Discovery in Data Streams*, pp. 53–64 (2005)
15. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* 34(1), 1–47 (2002)
16. Tsymbol, A.: The problem of concept drift: definitions and related work. Technical report, Trinity College Dublin (2004)
17. Widjantoro, D.H., Yen, J.: Relevant data expansion for learning concept drift from sparsely labeled data. *IEEE Trans. Knowl. Data Eng.* 17(3), 401–412 (2005)
18. Yeon, K., Song, M.S., Kim, Y., Choi, H., Park, C.: Model averaging via penalized regression for tracking concept drift. *J. Comput. Graph. Stat.* 19(2), 457–473 (2010)
19. Zliobaite, I.: Learning under concept drift: an overview. Technical report, Vilnius University (2010)

An Empirical Study of Recognizing Textual Entailment in Japanese Text

Quang Nhat Minh Pham, Le Minh Nguyen, and Akira Shimazu

Japan Advanced Institute of Science And Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan
{minhpqn,nguyenml,shimazu}@jaist.ac.jp

Abstract. Recognizing Textual Entailment (RTE) is a fundamental task in Natural Language Understanding. The task is to decide whether the meaning of a text can be inferred from the meaning of the other one. In this paper, we conduct an empirical study of the RTE task for Japanese, adopting a machine-learning-based approach. We quantitatively analyze the effects of various entailment features and the impact of RTE resources on the performance of a RTE system. This paper also investigates the use of Machine Translation for the RTE task and determines whether Machine Translation can be used to improve the performance of our RTE system. Experimental results achieved on benchmark data sets show that our machine-learning-based RTE system outperforms the baseline method based on lexical matching. The results also suggest that the Machine Translation component can be utilized to improve the performance of the RTE system.

Keywords: Textual Entailment, Machine Learning, Machine Translation.

1 Introduction

Recognizing Textual Entailment (RTE) is a fundamental task in Natural Language Understanding. It has been proposed with the aim of building a common applied semantic framework for modeling language variability [3]. Given two text portions T (text) and H (hypothesis), the task is to determine whether the meaning of H can be inferred from the meaning of T.

RTE can potentially be applied in many NLP tasks, such as Question Answering or Text Summarization. Applications of RTE have been reported in several studies: Question Answering [7], Information Extraction [16]. In these studies, RTE has been integrated as an important component. For instance, in Question Answering [7], a RTE component was used to determine if a candidate answer is the right answer for a question or not.

RTE task has received much attention in NLP research community, recently. There have been several RTE shared-tasks held by TAC conference [1], and many dedicated RTE workshops. However, to our knowledge, most published papers on RTE are for English, and there are only a very few of RTE studies for other

languages. It may be due to the fact that performance of state-of-the-art RTE systems significantly depends on RTE resources such as WordNet or database of inference rules, which have been mainly developed for English. Studies of RTE for other languages rather than English are useful, because for a specific language, there are language-specific linguistic phenomena which we need to take into account.

In this paper, we conduct an investigation of RTE task for Japanese, which has not been done before. We build a lightweight RTE system that is based on machine learning. RTE task is formalized as a binary classification problem and machine learning is applied to combine entailment features extracted from each pair of text T and hypothesis H . We evaluate our system using benchmark data sets from NTCIR9 RITE workshop [17] which is the first attempt of constructing a common benchmark for evaluating systems which automatically detect entailment, paraphrase, and contradiction in texts written in Japanese.

The use of bilingual corpora and Machine Translation for RTE task has been explored in the cross-lingual textual entailment recognition task [13,14] which is the task of recognizing textual entailment relationship between two text portions in different languages. Different from [13,14], in the current study, Machine Translation is used for monolingual RTE. In our system, a Machine Translation component is used to produce English translation of original Japanese data, and both original Japanese data and its translation are used to learn an entailment classifier. Our method is based on a reasonable assumption that if T entails H , then the translation of T should also entail the translation of H . We expect that English translation of the text and the hypothesis can provide more useful features for the RTE system to determine entailment relationship in the pair more correctly.

In short, the objectives of our paper are as follows.

- The paper investigates a machine-learning-based approach to recognizing textual entailment in Japanese. The effects of entailment features on the performance of our Japanese RTE system are analyzed in detail.
- We analyze the impact of various resources on the performance of our RTE system by conducting ablation tests.
- We determine whether Machine Translation can be used to improve the performance of our system. In order to do that, we integrated a Machine Translation component to the RTE system.

The remainder of the paper is organized as follows. Section 2 presents some related work to our research. In Section 3, we describe our machine-learning-based RTE system. In Section 4, we present experimental results achieved on two Japanese RTE data sets. Finally, Section 5 gives conclusions and some remarks.

2 Related Work

2.1 Machine-Learning-Based Approaches to RTE

In these methods, RTE task has been formulated as a classification problem. Multiple entailment features extracted from each pair T/H are combined using

machine learning methods [12]. Features may be similarity measures applied on the pair or other features such as polarity difference between T and H.

2.2 Using Bilingual Corpora and Machine Translation

Mehdad et al. [13] proposed the cross-lingual textual entailment (CLTE) task where text T and hypothesis H are written in different languages. A basic solution for CLTE task was proposed, in which a Machine Translation (MT) system is added to the front-end of an existing RTE engine. For instance, for a pair of English text and Spanish hypothesis, the hypothesis will be translated into English and then, the RTE engine will be run on the pair of the text and the translation of the hypothesis.

Mehdad et al. [14] proposed a new method for CLTE task, which takes advantages of bilingual parallel corpora by extracting information from the phrase-table to enrich inference and entailment rules, and using extracted rules for a distance-based entailment system. The use of bilingual parallel corpora for monolingual textual entailment was also explored. The main idea of that work is to increase the coverage of monolingual paraphrase tables by extracting paraphrases from bilingual parallel corpora and use extracted paraphrases for monolingual RTE.

Our approach makes use of Machine Translation for monolingual RTE. In our machine-learning-based RTE system, we combine both features extracted from data in original language and features extracted from translation data produced by a MT component to learn an entailment classifier.

3 Proposed Method

In our paper, we adopt the machine learning based approach to building a RTE system. RTE task is formulated as a binary classification problem in which each instance consists of a pair of a text T and a hypothesis H.

In this section, we describe our RTE system. The RTE system is divided into four main modules as shown in Figure 1: bilingual enrichment, preprocessing, feature extraction, and training.

First, each Japanese pair T/H is automatically translated into English using a MT engine. Then in preprocessing, both the Japanese pair and its associated translation pair are analyzed. After that, features which are extracted from the pair and its translation are input to an entailment classifier to determine if the entailment relationship exists in the pair or not. The entailment classifier is trained on the training set consisting of pairs T/H with their gold labels.

Our system uses Support Vector Machines (SVMs) [19], a robust method for classification problems, to train an entailment classifier which can determine whether the text T entails the hypothesis H for each pair T/H.

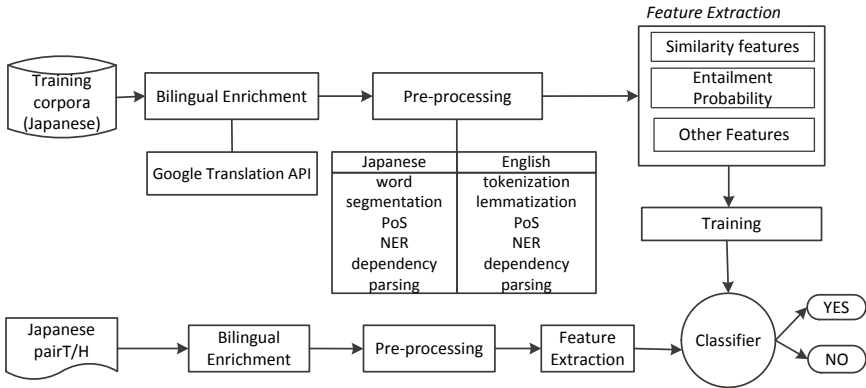


Fig. 1. Architecture of Japanese RTE System

3.1 Bilingual Enrichment

In order to make use of the bilingual constraint for RTE, original RTE corpus in Japanese is automatically translated into English, using Google Translator Toolkit¹.

3.2 Preprocessing

Japanese Pairs: We use Cabocha tool [10] for data preprocessing. For each pair, preprocessing process consists of tokenizing, chunking, named entity recognition, and dependency parsing. Parsed content of each sentence is represented in XML format.

English Pairs: Each Japanese T/H pair in our corpus is associated with its English translation. We use Stanford-CoreNLP tool to perform preprocessing for English pairs². Stanford-CoreNLP provides a set of fundamental natural language processing tools which can take raw English text input. At lexical level, we use the tool to perform tokenization, lemmatization, part-of-speech tagging, and named-entity recognition. At syntactic level, dependency parsing is done.

3.3 Entailment Classifier

In the system, we train an entailment classifier on the training set consisting of annotated pairs T/H. Each pair T/H is represented by a feature vector $\langle f_1, \dots, f_m \rangle$ which contains multiple similarity measures of the pair and some other features. For each training instance consisting of a pair T/H, features are

¹ Google Translator Toolkit: <http://translate.google.com/toolkit>

² Stanford CoreNLP is available on:

<http://nlp.stanford.edu/software/corenlp.shtml>

extracted from both the original pair in Japanese and its associated English translation pair. In the rest of this section, we describe features used in the entailment classifier.

Similarity Features: A large part of similarity features used in the entailment classifier is similar to features used in [12]. We use different kinds of text similarity/distance measures applied on the pair and its English translation. These measures capture how H is covered by T.

From each pair T/H (Japanese pair or English translation pair), text similarity/distance measures are applied on two pairs of strings:

- **Pair 1:** Two strings which consist of words of T and H in surface forms. Punctuations and special characters are removed. Stop words are removed for English pairs.
- **Pair 2:** Two strings which consist of base forms of words in T and H, respectively. Punctuations and special character are removed. Stop words are removed for English pairs.

We give a brief description of similarity features which are used to train the entailment classifier as follows.

a) Word overlap

Word-overlap feature captures lexical-based semantic overlap between T and H, which is a score based on matching each word in H with some words in T [4]. Japanese WordNet and English WordNet [5] are used in computing lexical matching. Matching criterion for two English words is the same as in [4]. For Japanese, a word h_w in H is considered as a matching word of a word t_w in T if they have the same surface or base form, or h_w is hypernym, meronym, or entailed word or of t_w .

b) Levenshtein distance

Levenshtein distance [12] of two strings is the minimum number of edit operations needed in order to transform a string to the other one. Allowable edit operations are deletion, insertion, or substitution of a single token. In our system, the edit distance from T to H is computed.

c) BLEU measures

BLEU score is a popular evaluation metric used in automatic machine translation [15]. It measures how a translation generated by a MT system is close to reference translations. The main idea is to compute n -gram matching between automatically generated translations and reference translations. In RTE problem, we used BLEU precision of H and T (T is cast as a reference translation) based on uni-gram, 2-gram, and 3-gram. Both baseline BLEU precision and modified n -gram precision are used.

d) Longest Common Subsequence String (LCS)

LCS feature computes the length of the longest common subsequence string between T and H [8]. The LCS feature is normalized by dividing by the length of H.

e) *Other similarity/distance measures*

We compute various similarity/distance measures which have been used for RTE: Jaccard coefficient, Mahatan distance, Euclidean distance, Jaro-Winkler distance, Cosine Similarity, and Dice Coefficient. For details of these measures, see [12].

Entailment Probability: The entailment probability that T entails H is computed based on the probabilistic entailment model in [6]. The main idea is as follows. The probability that the entailment relationship exists in the pair, $P(H|T)$ is computed via the probability that each individual word in H is entailed by T. The probability $P(H|T)$ is computed by the following equation:

$$P(H|T) = \prod_j P(h_j|T) \quad (1)$$

where the probability $P(h_j|T)$ is defined as the probability that the word h_j in H is entailed by T. The probability $P(h_j|T)$ is computed by:

$$P(h_j|T) = \max_i P(h_j|t_i) \quad (2)$$

In Equation 2, $P(h_j|t_i)$ can be interpreted as the lexical entailment score between words t_i and h_j . By this decomposition, the overall probability $P(H|T)$ is computed by the following equation.

$$P(H|T) = \prod_j \max_i P(h_j|t_i) \quad (3)$$

The lexical entailment score of two words w_1 and w_2 is computed by using the word similarity score between them. For English, lexical entailment scores are computed based on Levenshtein distance as in [11]

$$sim(w_1, w_2) = 1 - \frac{dist(w_1, w_2)}{max(length(w_1), length(w_2))} \quad (4)$$

For Japanese pair, we use the Japanese thesaurus, Nihongo goitaikei [9] to compute the similarity of two words.

Dependency-Parse-Based Features: Dependency relation overlap has been used in paraphrase identification [20]. For RTE task, we compute the dependency relation overlap of H and T by the following equation:

$$RelationOverlap = \frac{|relations(H) \cap relations(T)|}{|relations(H)|} \quad (5)$$

where $relations(s)$ denotes the set of head-modifier relations of the sentence s .

Table 1. Data statistics

Dataset	Y	N	Total
BC Subtask - Dev set	250	250	500
BC Subtask - Test set	250	250	500
Exam Subtask - Dev set	204	295	499
Exam Subtask - Test set	181	261	442

Named-Entity Mismatch: In a pair T/H, if the hypothesis contains a named-entity which does not occur in the text, the text may not entail the hypothesis. We use an indicator function π to compute the named-entity mismatch feature of T and H: $\pi(T, H) = 1$ if H contains a named-entity that does not occur in T and $\pi(T, H) = 0$, otherwise. We compute named-entity mismatch for both Japanese pairs and their associated English translation pairs.

Polarity Mismatch: The polarity mismatch in a pair T/H may indicate that T does not entail H. We compute polarity mismatch in a pair T/H using the Polarity Weighted Word List [18]. In that list, each Japanese word is associated with a weight that indicates whether the word has positive meaning or negative meaning. We use an indicator function to capture if words in the root nodes of dependency parses of T and H have opposite polarity. The polarity mismatch is applied only on Japanese pairs.

4 Experiments and Results

4.1 Data Sets

NTCIR9 RITE workshop [17] provides benchmark data for several Japanese RTE subtasks: binary-class subtask (BC subtask), multi-class subtask (MC subtask), Entrance Exam subtask (Exam), and RITE4QA subtask. In order to evaluate our system, we use data sets from BC subtask and Entrance Exam subtask. The BC subtask is the basic problem setting of RTE which is to determine whether the meaning of a hypothesis H can be inferred from the meaning of a text T. The Entrance Exam subtask is the same as BC subtask in terms of input and output, but data for Entrance Exam subtask is created from actual college-level entrance exams. Therefore, data of Entrance Exam subtask may be closer to real-world data than BC subtask.

Development set and test set are provided for BC subtask and Entrance Exam subtask. Each data set consists of pairs T/H along with their gold-standard labels “Y” or “N”. For each subtask, we train an entailment classifier on the development portion and evaluate the trained classifier on the test portion. Table 1 shows statistical information of each data set.

In experiments, classification accuracy is used to evaluate RTE methods. The accuracy is the percentage of pairs which are correctly predicted by a classifier.

Table 2. Experimental Results

Subtask	Baseline	SVM_mono	SVM_bi
BC	49% (245/500)	56.6% (283/500)	56.8% (284/500)
Exam	62.2% (275/442)	65.6% (290/442)	69.4% (307/442)

4.2 Machine Learning Tools

In training and testing, we used libSVM [2], an efficient machine learning tool for classification problems. In training, we use default initial parameters which are provided in the tool.

4.3 Baselines

A trivial baseline for the task is to randomly choose label for each pair. In experiments, we use a stronger baseline which is based on local lexical matching between T and H. Lexical matching score computed on each Japanese pair is compared with a threshold value tuned on the development portion of each data set.

4.4 Experimental Results

Experimental results achieved on the test sets of the two subtasks are shown in the Table 2. SVM_mono is the machine-learning-based system which uses monolingual features extracted from Japanese pairs to train the entailment classifier. SVM_bi is the system which uses all features extracted from both original Japanese pairs and their associated English translation pairs.

In BC subtask, both SVM_mono and SVM_bi method significantly outperform the baseline method (we use McNemar Test with $p < 0.05$), while the performance of SVM_mono and SVM_bi are almost the same.

In Exam subtask, both SVM_mono and SVM_bi method obtain better accuracy against the baseline, but only SVM_bi method significantly outperforms the baseline. The results also indicate that SVM_bi method achieves statistically significant improvement compared with SVM_mono method for Exam subtask.

4.5 Result Analysis

Table 3 compares the number of false-positive pairs and false-negative pairs predicted by the three above methods on the test set of each subtask. False-positive pairs are pairs which are predicted as “Y” pairs by a system while in gold standard, they are “N” pairs. False-negative pairs are pairs which are predicted as “N” pairs by a system while in gold standard, they are “Y” pairs.

Analyzing false-positive pairs predicted by methods SVM_bi and SVM_mono, we see that false-positive pairs mainly come from “N” pairs in which H is highly covered by T in terms of lexical, such as pair 15 in Figure 2. Among true-entailment pairs which our systems do not correctly classify, many pairs use

Table 3. Error Statistics

Methods (Subtask)	False-positive	False-negative
Baseline (BC)	164	91
SVM_mono (BC)	122	95
SVM_bi (BC)	131	85
Baseline (Exam)	88	79
SVM_mono (Exam)	46	106
SVM_bi (Exam)	44	91

# ID	Text	Hypothesis	Subtask	Entailed
23	バラック・オバマ氏。43歳。移民したケニア人を父に、カンザス州出身の白人女性を母に持つ「アフリカン・アメリカン」である。 Barrack Obama, 43 years old, whose father is a Kenyan immigrant and mother is a white woman from Kansas state, is an African American.	オバマ氏はアフリカ系アメリカ人である。 Obama is an African American.	BC	Y
12	戦車は、第一次世界大戦時に塹壕戦の突破を目的とした兵器として開発された。 In the World War I, tanks were developed as a war weapon to break through trench warfare.	第一次世界大戦では、新兵器として戦車（タンク）が用いられた。 In the World War I, tanks were used as a new war weapon.	Exam	Y
15	日本では多くの人が人生を仕事に費やしている。 In Japan, many people devote their life in work.	日本では多くの人が私生活を犠牲にしている。 In Japan, many people sacrifice their personal life	BC	N
148	主婦や求職中の人も2割いる。 20% of people are housewives and people who are seeking jobs.	2割が、「職場を持たない人」だ。 20% of people are people who do not have workplace.	BC	Y

Fig. 2. Example pairs in test sets. The meaning of text and hypothesis in English is shown for comprehension.

complex entailment or paraphrase rules, such as pair 148 shown in Figure 2. Therefore, a large paraphrase table of phrases and a database of entailment rules may be important in order to improve the classification accuracy of the system.

In BC subtask, the difference between SVM_mono and SVM_bi is not so significant, so it is not very clear whether the MT component is helpful or not. In Exam subtask, the MT component shows significant contribution to performance of the system. As shown in Table 3, the number of false-negative pairs predicted by SVM_bi is less than the number of false-negative pairs predicted by SVM_mono. It may indicate that the MT component used in SVM_bi provides more evidences for detecting entailment relationship in “Y” pairs through translation. Pair 23 in BC’s test set and pair 12 in Exam’s test set in Figure 2 shows two examples which SVM_bi correctly predicts their entailment labels while SVM_mono does not.

Table 4. Feature Analysis

Setting	BC	Exam
SVM_mono + LemmaSim	56.2% (-0.4)	65.1% (-0.5)
SVM_mono + SurSim	56.6% (+0)	64.5% (-1.1)
SVM_mono + SynSem	53.4% (-3.2)	64.0% (-1.6)
SVM_mono + LemmaSim + SurSim	56.8% (+0.2)	64.5% (-1.1)
SVM_mono + LemmaSim + SynSem	56.2% (-0.4)	65.6% (+0)
SVM_mono + SurSim + SynSem	56.0% (-0.6)	66.1% (+0.5)
SVM_mono + All Features	56.6%	65.6%
SVM_bi + LemmaSim	57.2% (+0.4)	68.1% (-1.3)
SVM_bi + SurSim	57.0% (+0.2)	65.8% (-3.6)
SVM_bi + SynSem	53.4% (-3.4)	65.6% (-3.8)
SVM_bi + LemmaSim + SurSim	58.2% (+1.4)	68.3% (-1.1)
SVM_bi + LemmaSim + SynSem	55.8% (-1.0)	69.2% (-0.2)
SVM_bi + SurSim + SynSem	56.2% (-0.6)	69.9% (+0.5)
SVM_bi + All Features	56.8%	69.4%

4.6 Feature Analysis

We conduct feature analysis in order to understand impacts of features on the performance of machine-learning-based RTE systems.

We divide features set into three categories as follows.

- **LemmaSim** consists of similarity features computed on base (lemma) form of each pair T/H.
- **SurSim** consists of similarity features applied on surface form of each pair T/H.
- **SynSem** consists of other features: entailment probability, dependency-parse based feature, named-entity mismatch and polarity mismatch features.

Entailment classifiers are trained using above features subsets and combination of them on the development sets. Table 4 shows accuracies of various settings on the test sets of two subtasks.

Feature analysis indicated that similarity features significantly contribute to the performance of RTE systems. As shown in Table 4, without using similarity features, the accuracies of SVM_bi and SVM_mono decrease much. Similarity features applied on base form of each pair T/H and its English translation (in the group LemmaSim) are important in Exam subtask while the contribution of features in SynSem group is not so significant in both two subtasks.

4.7 Ablation Tests

In RTE task, it is interesting to know how additional resources or components contribute to the performance of our Japanese RTE system. This section presents ablation tests for two subtasks. We only analyze the effects of RTE resources and components to SVM_mono method to avoid unpredictable errors propagated

Table 5. Ablation Tests

Ablated Resource	BC	Exam
JWordNet	0%	0%
Goi Taikei	0.2%	0.2%
Polarity Words	-0.2%	0.7%
JWordNet + Goi Taikei	0.2%	0.2%
JWordNet + Polarity Words	-0.2%	0.5%
Goi Taikei + Polarity Words	-0.2%	-0.4%
JWordNet + Goi Taikei + Polarity Words	0%	0%

from the Machine Translation component. Table 5 provides performance differences between of the SVM_{mono} using complete additional resources and the system without using some resources. The percentages shown in Table 5 indicate the contribution of resources to the performance of the system. As indicated in the table, the impact of additional resources on the performance of our system is not so significant. A possible explanation for this may be that resources were used only in computing a small subset of features in our system. Specifically, Japanese WordNet was used to compute word-overlapping features, Nihongo goi taikei was used to compute entailment probability feature, and Polarity Word List was used to compute polarity mismatch in a pair.

5 Conclusion

We have presented an empirical study of recognizing textual entailment for Japanese. Our system is based on machine learning, in which multiple entailment features extracted from both original Japanese pairs and their English translation are combined to learn an entailment classifier. Extensive feature analyses and ablation tests have been conducted to quantitatively measure the impact of various entailment features and additional resources on the performance of our RTE system. Experimental results achieved on the two benchmark data sets indicated that our proposed method significantly outperforms the baseline method based on lexical matching, and Machine Translation component can be used to improve the performance of the RTE system.

Our study still has several major limitations. First, the system is not very precise at detecting *hard* false-entailment pairs in which H is highly covered by T. Second, due to the lack of entailment and paraphrase rules, our system fails to determine the entailment relationship in pairs that use complex inference rules. We plan to address these problems by developing an alignment component for RTE task and acquiring entailment/paraphrase rules from large text corpora.

References

1. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The fifth pascal recognizing textual entailment challenge. In: Proceedings of TAC Workshop (2009)

2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27:1–27:27 (2011), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. Dagan, I., Glickman, O., Magnini, B.: The Pascal Recognising Textual Entailment Challenge. In: Quiñero-Candela, J., Dagan, I., Magnini, B., d’Alché-Buc, F. (eds.) *MLCW 2005. LNCS (LNAI)*, vol. 3944, pp. 177–190. Springer, Heidelberg (2006)
4. Dagan, I., Roth, D., Massimo, F.: A tutorial on textual entailment (2007), <http://l2r.cs.uiuc.edu/~danr/Talks/DRZ-TE-Tutorial-ACL07.ppt>
5. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press (1998)
6. Glickman, O., Dagan, I., Koppel, M.: Web based probabilistic textual entailment. In: *Proceedings of the 1st RTE Workshop*, Southampton, UK (2005)
7. Harabagiu, S., Hickl, A.: Methods for using textual entailment in open-domain question answering. In: *Proceedings of ACL*, pp. 905–912 (2006)
8. Hirschberg, D.S.: Algorithms for the longest common subsequence problem. *J. ACM* 24, 664–675 (1977)
9. Ikehara, S., Miyazaki, M., Sirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., Hayashi, Y.: *Nihon-go goi taikai*, Iwanami, Japan (1997) (in Japanese)
10. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pp. 63–69 (2002)
11. MacCartney, B.: *Natural Language Inference*. Ph.D. thesis, Stanford University (2009)
12. Malakasiotis, P., Androutsopoulos, I.: Learning textual entailment using svms and string similarity measures. In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 42–47 (2007)
13. Mehdad, Y., Negri, M., Federico, M.: Towards cross-lingual textual entailment. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 321–324 (June 2010)
14. Mehdad, Y., Negri, M., Federico, M.: Using bilingual parallel corpora for cross-lingual textual entailment. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1336–1345 (June 2011)
15. 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 (2002)
16. Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., Lavelli, A.: Investigating a generic paraphrase-based approach for relation extraction. In: *Proceedings of EACL*, pp. 401–408 (2006)
17. Shima, H., Kanayama, H., Lee, C.W., Lin, C.J., Mitamura, T., Miyao, Y., Shi, S., Takeda, K.: Overview of ntcir9 rite: Recognizing inference in text. In: *NTCIR9 Proceedings* (2011)
18. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 133–140. Association for Computational Linguistics, Ann Arbor (2005), <http://www.aclweb.org/anthology/P05-1017>
19. Vapnik, V.N.: *Statistical learning theory*. John Wiley (1998)
20. Wan, S., Dras, M., Dale, R., Paris, C.: Using dependency-based features to take the “para-farce” out of paraphrase. In: *Proceedings of ALTW* (2006)

Automated Detection of Local Coherence in Short Argumentative Essays Based on Centering Theory

Vasile Rus and Nobal Niraula

Department of Computer Science, The University of Memphis
Memphis, TN, 38152, USA
{vrus, nobal}@memphis.edu

Abstract. We describe in this paper an automated method for assessing local coherence in short argumentative essays. We use ideas from Centering Theory to measure local coherence of essays' paragraphs and compare it to human judgments on one analytical feature of essay quality called Continuity. Paragraphs which correspond to a discourse segment in our work and which are dominated by one prominent concept were deemed locally coherent according to Centering Theory. A dominance measure was proposed based on which local coherence was judged. Results on a corpus of 184 argumentative essays showed promising results. Our findings also suggest that focusing on nominal subject for detecting candidate concepts for a discourse segment's central concept is sufficient, which confirms previous findings. Compared to previous approaches to assessing local discourse coherence in essays, our method is fully automated.

Keywords: coherence, centering, short essay scoring.

1 Introduction

We describe in this paper an automated method for assessing local coherence in short argumentative essays. The method was inspired from an expert analysis of short essays' cohesion and coherence and relies on ideas from Centering Theory [6].

Coherence is an inherent property of text discourse, which is defined as a set of coherent sentences [8]. A related concept, cohesion refers to the explicit cues in the text that link the ideas together [4, 8, 10]. While cohesion defines the texture that keeps a text together (in the sense defined by Halliday and Hassan [7]), coherence defines the overall structure and meaning of the text, i.e. the discourse. In other words, cohesion is the fabric while coherence is the outfit. Obviously, same fabric could lead to very different outfits, some more "coherent" than others. It is important to add that according to a school of thought from Cognitive Science, the coherence, i.e. the outfit in our metaphor, is reader-dependent [4, 10]. That is, different readers could see different outfits depending on their background relative to the text they are reading. It is beyond the purpose of this paper to discuss this point in more detail. On the other hand, the Natural Language Processing (NLP) community adopts a more reader-independent view of coherence (or by default an average reader is assumed; [8]), as explained later. In this paper, we adopt the NLP-community's view of

coherence but not necessarily disagreeing with the other view of reader-dependence coherence.

Following Centering Theory, we distinguish between local and global coherence. Local coherence refers to the coherence of a discourse segment, e.g. a paragraph, while global coherence refers to the overall coherence of an entire text, i.e. an essay in our case.

We focus on short argumentative essays similar to the ones used in the Standard Achievement Test (SAT). In such essays, students are required to take a position (main thesis) relative to the topic/theme described by the essay prompt and argue for it. Arguments must be supported by evidence to be convincing.

Our work can be viewed as being part of the broader context of automatic assessment of written essays. Automatic essay scoring (AES; also known as automatic essay grading or automatic essay evaluation) is the task of automatically assessing student-written essays or, in a broader sense, any written response [2,15,18]. AES holds the hope of providing systematic, timely, and cost-effective solutions to the task of grading essays, which can be very expensive for standardized tests that are taken by hundreds of thousands of students and whose essays must be graded.

It is beyond the scope of this paper to provide an overview of previous work related to automatic scoring of essays. Rather, we focus on discourse level features that could be used in AES tools. Indeed, the discourse structure of essays is an important aspect used often in assessing the essays' quality [1]. To this end, we analyzed the relation between coherence and essay quality as measured by holistic scores.

We started with a corpus of 184 essays which were analyzed along two analytical features, Continuity and Reader Orientation, as well as overall quality. We make the claim the Continuity measures local coherence as defined in Grosz and Sidner's theory of discourse [5] and Grosz, Joshi, and Weinstein's Centering Theory [6]. This paper is a first step towards providing the evidence supporting this claim. Indeed, we show how an automated method based on ideas inspired from Centering Theory can be used to compute essays' local coherence. A centered paragraph would be a paragraph dominated by one central concept which should appear in prominent syntactic roles in the sentences of the paragraph according to Centering Theory. Discourse segments dominated by one center concept are deemed locally coherent.

In our work, a discourse segment corresponds to one paragraph. In other words, we make the assumption that each paragraph contains only one discourse segment. While this is not always the case because a paragraph may contain several discourse segments, each with its own center, our analysis of argumentative essays indicated that students tend to have only one discourse segment per paragraph. That is, there is a tendency to develop only one supporting argument in short essays' body paragraphs. Another argument spawns from the difficulty of detecting discourse segments automatically [19], existing algorithms assuming coherence when detecting the segments which is not a valid assumption in student-written essays.

To decide whether the paragraph is locally coherence we automatically compute a dominance percentage for each concept appearing in subject positions in essays' paragraphs. An analysis of variance (ANOVA) with the dominance percentage of the most dominant concept in a paragraph as the factor was then performed to reveal any significant differences between low and high cohesion.

The rest of the paper is organized as in the followings. The next section provides an overview of related work on coherence and automated methods to capture coherence with a focus on methods developed in the context of automated essay scoring (AES). Then, we provide the conceptual framework behind our basic idea of using Centering Theory to capture the local coherence of short essays. The Experiments and Results section describes our experimental setup and the results obtained. We conclude with Conclusions and Future Work.

2 Literature Review

According to McNamara and colleagues [10], coherence is the understanding the reader derives from the text while cohesion refers to the explicit cues in text that allow the reader to connect the ideas in it. Coherence is therefore a reader-dependent feature of text influenced by factors such as prior knowledge and reading skills [10].

Researchers in the field of Natural Language Processing (NLP; [9]) define coherence in a more reader-independent way as the goal is to develop automated tools to process discourse, e.g. discourse parsers, without a specific user model in mind or with an implied assumption of an average reader. Typically, the development of automated discourse processing tools is based on journalistic texts such as news articles, which are targeted primarily to average readers. We adopt a similar reader-independent view in this article. For instance, Jurafsky and Martin [9] refer to coherence as the meaning relations between textual units while cohesion is the way in which textual units are linked together. The meaning of the entire discourse can be understood by following the coherence relations among its textual units. Based on this view, NLP researchers defined a set of coherence relations, such as explanation, which are used in the development of automated discourse parsers to discover the structure of discourse [11, 12]. The output of discourse parsers is usually a discourse tree implying a hierarchically structure but more complex structures are possible such as graphs in which a coherent or discourse relation can be observed between any two textual units [14]. It should be noted that linear discourse structures are also used [2].

An important issue in discourse parsing is choosing the set of coherence relations to use. Hierarchical models of discourse distinguish among global coherence, which is revealed through the coherence relations among larger discourse segments shown at higher levels in discourse parse trees, and local coherence, which is observed among utterances within a discourse segment. A discourse segment is more or less coherent depending to the cognitive load it puts on the reader. Discourse segments with a clear focus, i.e. center of attention, are more coherent according to Centering Theory [6].

As mentioned before, our work is conducted in the context of automatic essay scoring. The discourse structure of essays is an important aspect used often in assessing the essays' quality [1]. The complex interactions among coherence, cohesion, and text organization are essential to accurately infer the discourse plan and structure.

It is beyond the scope of this work to analyze in-depth the advantages and disadvantages of AES systems. We rather focus on discourse analysis components used in AES systems. More details regarding automated analysis of essay's discourse

structure are available for E-rater, the AES system developed by Educational Testing Services (ETS; [1, 15]). Shermis and Burstein [15] described in detail E-raters' method for analyzing essays' discourse structure. They assumed a linear discourse structure for essays which are regarded as a sequence of discourse spans each serving one essay-specific communicative goal such as thesis statement, main idea, or supporting idea. It should be noted that E-rater was developed for assisting with scoring the Analytical Writing Assessments of the Graduate Management Admission Test (GMAT). GMAT essay questions are of two types: analysis of an issue and analysis of an argument. E-rater's discourse processing module was developed with the aim of handling both types of essays. Shermis and Burstein [15] did not report about the usefulness of their discourse processing method on scoring essays or providing feedback to individual students, in case the system is used in instructional settings.

The most related work to ours is by Miltsakaki and Kukich [11] who assessed local discourse coherence of essays using a measure of topic continuity, called rough-shift transitions. As opposed to their work, our method is fully automated, e.g. they manually solved the coreferring expressions while we automatically detect them. Furthermore, we propose a different measure of topic continuity called dominance. In addition, we worked with a larger set of essays (184 versus 100).

Our goal here is different from designing a fully-automated method for parsing essays' discourse. Rather, we analyze the relationship between coherence and essay quality as measured by holistic scores and relate the observed patterns of this relationship to elements from Centering Theory.

3 Motivation

Our work started by analyzing the relation between cohesion, coherence and essay quality as measured by holistic scores. We focus only on coherence and essay quality in this paper. We worked with a corpus of 184 essays which were analyzed manually by two experts along two analytical features, Continuity and Reader Orientation, as well as overall quality. The details of how the corpus was collected and annotated are provided in [3]. Continuity assesses an essay's exhibited strength of connections of ideas and themes within and between the essays' paragraphs. Continuity was meant to measure cohesion according to [3] although they found out that computational indices of cohesion do not correlate well with overall essay quality. On the other hand, Continuity does correlate with overall essay quality ($r=0.646$). We show here that Continuity measures local coherence.

Reader Orientation represents the essay's overall coherence and ease of understanding. The holistic scale and all of the analytic features had a minimum score of 1 and a maximum score of 6. Table 1 provides an overview of the essay dataset with some statistics.

A closer analysis of the relation between Continuity and Reader Orientation on one hand and overall essay quality, i.e. SAT score, on the other hand, revealed several patterns.

Table 1. The set of essays and their distribution by essay prompt together with some statistics

<i>Prompt Label</i>	<i>Prompt</i>	<i>Total</i>	<i>Number of Words (SD)</i>	<i>Number of Types (SD)</i>	<i>Number of Paragraphs (SD)</i>
Creativity	<i>Some people say that in our modern world, dominated by science, technology, and industrialization, there is no longer a place for dreaming and imagination. What is your opinion?</i>	59	693.36 (139.47)	239.07 (48.70)	5.37 (1.19)
Religion and Television	<i>Marx once said that religion was the opium of the masses. If he was alive at the end of the 20th century, he would replace religion with television.</i>	60	708.37 (140.63)	254.5 (45.93)	5.28 (1.64)
Unequal Men	<i>In his novel 'Animal Farm', George Orwell wrote "All men are equal: but some are more equal than others". How true is this today?</i>	65	758.57 (112.96)	258.06 (44.29)	5.35 (1.37)

The two measures of Continuity (C; average=3.57, stdev=1.24) and Reader Orientation (RO; average=3.96, stdev=0.90) correlate strongly with the holistic SAT scores (average=3.53, stdev=1.07), the latter correlating stronger ($r=.786$ for RO; $r=0.646$ for C). RO tends to be higher than SAT scores for low and medium quality essays (as judged by holistic SAT scores). Continuity scores tend to be closer to SAT holistic scores with a higher degree of variation for medium quality essays. For extremely high-quality essays (i.e., those essays scored 5.5 or 6) the three measures (C, RO, and SAT) tend to converge.

Probably the most interesting emerging pattern is the fact that while RO scores tend to follow the SAT scores closely, the C scores fluctuate around the SAT scores. Indeed, our data shows that essays can have low or high Continuity scores while still being overall high or low quality essays. That is, at almost all levels of SAT scores, except the very extremes (SAT scores of 1, 1.5 and 5.5, 6) Continuity scores group in two very different categories, low and high scores, forming a binary pattern. This pattern can be seen in Figure 1 where we plotted the scores for the three measures with the essays on X-axis being ordered based on the holistic SAT scores.

In order to characterize in more depth this difference in which Continuity (C) and Reader Orientation (RO) scores follow SAT scores, we analyzed essays that had extreme values for C and RO. In particular, we looked at four possible combinations of C and RO scores: (low-C, low-RO), (low-C, high-RO), (high-C, low-RO), and (high-C, high-RO).

The selected essays were analyzed manually in terms of local coherence at paragraph level. We show in Figure 2 two paragraphs from two different essays that have low and high Continuity scores, respectively, while both having high Reader Orientation scores. The two essays are of high quality as their SAT scores are 5.5 and 4.5, respectively. These examples, which differ significantly in their Continuity scores, display different interrelations among the main theme and the supportive arguments. The high Continuity paragraph at the top of Figure 2 has a "thin" main theme, *dreaming and imagination*, which mentioned only in the opening and closing sentences of the paragraph. The paragraph focuses then on the supporting argument that is being developed and forms the central concept of the paragraph.

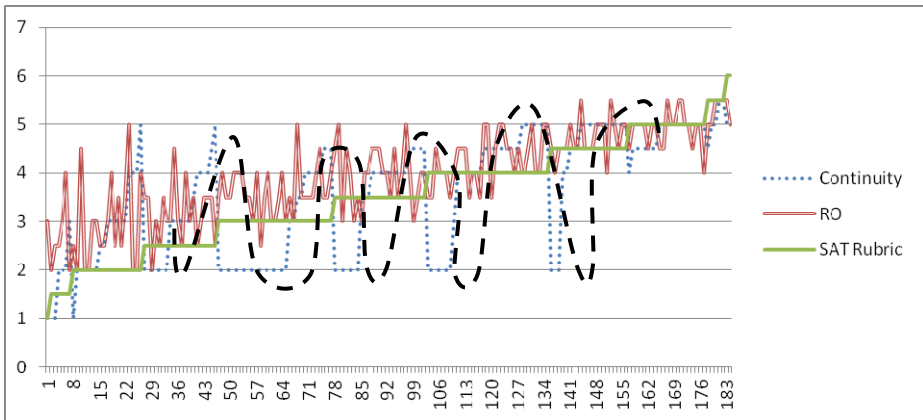


Fig. 1. Continuity, Reader Orientation, and SAT scores for our MSU set. Essays on the X-axis are ordered based on their holistic SAT score. The binary pattern for Continuity is shown with the bold black line. The bold, flat line indicates a floor-effect for Continuity-scores for medium quality essays.

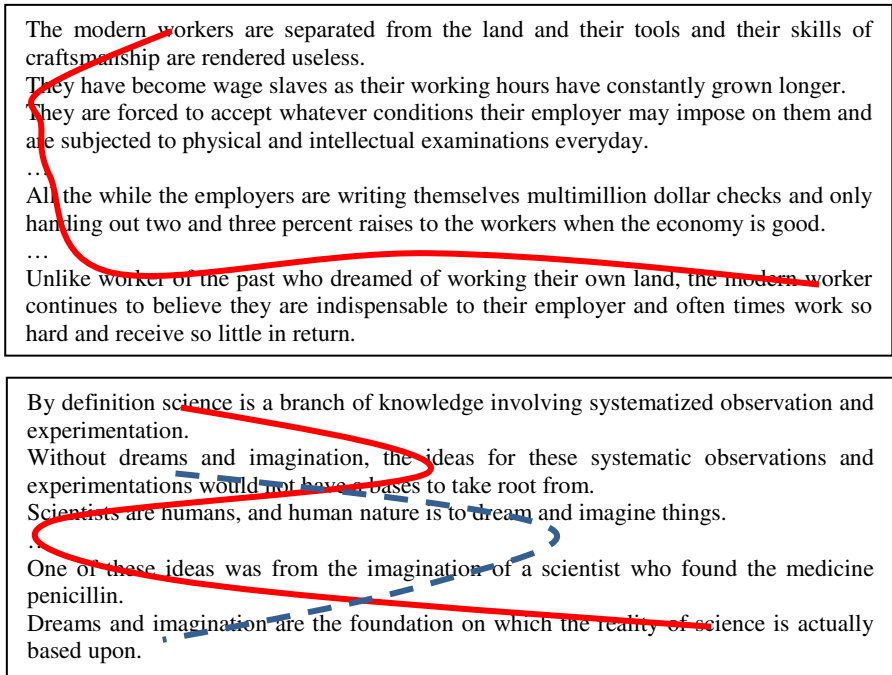


Fig. 2. Examples of two essay paragraphs one with good local coherence (top; Continuity score of 5.5) and one with poor local coherence (Continuity score of 2)

On the other hand, in the low Continuity paragraph, shown at the bottom of Figure 2, the main theme is referred to throughout the entire paragraph, being present even in the middle of the essay's body paragraphs. The student writer comes back to the main theme, *dreaming and imagination*, every other sentence, in this example paragraph. The paragraphs seems to lack focus, that is, is not centered on a single concepts and therefore having low local coherence according to Centering Theory.

Our conclusion from the qualitative analysis of paragraphs in essays with extreme values for Continuity and Reader Orientation is that the essay body paragraphs' discourse flow must focus on the supporting arguments with as few references to the main theme as possible. To test this hypothesis we will propose a method based on Centering Theory. The idea is to automatically detect the centers of paragraphs in essays. Paragraphs that are not focused (i.e. have more than one center or focus) will lack local coherence according to Centering Theory and expected to have corresponding low Continuity scores assigned by human judges.

4 Centering-Based Approach to Detecting Local Coherence

As already mentioned, our basic approach to capturing local coherence in essays is to use ideas from Grosz and Sidner's theory of discourse [5] and Grosz, Joshi, and Weinstein's Centering Theory [6]. Grosz and Sidner view discourse as the result of three interrelated components: linguistic structure, intentional structure, and attentional state. The linguistic structure refers to the discourse segments into which utterances naturally group. The intentional structure captures speaker's intentions expressed as discourse purposes and their relationships. The intentions provide the basic rationale for the discourse. The attentional state models the discourse participants' focus of attention at any given moment. Centering Theory links the attentional state and perceived coherence of discourse segments, i.e. local coherence. Discourse is more coherent if it limits the number of inferences necessary to understand its utterances. More locally, discourse segments are perceived more coherent if the focus of attention within the segment is carefully maintained by the use of appropriate linguistic devices, e.g. placing the entity that is the discourse focus in prominent syntactic positions, e.g. subject, in utterances. In our work, we make the assumption that a discourse segment corresponds to one paragraph. While this is not always the case because a paragraph may contain several discourse segments in some cases, each with its own center, our analysis of essays indicated that students tend to have only one discourse segment per paragraph.

To illustrate the relation between centering and local coherence we use the two paragraphs in Figure 3 (from [6]). The top paragraph is coherent because it has one center, i.e. *John*, appearing in subject positions of every sentence in the paragraph. The bottom paragraph is less coherent as it lacks centering on one concept – both John and the store appear in subject positions of paragraph's sentences.

In our case, paragraphs with high Continuity are characterized by high local coherence while low Continuity paragraphs have poor local coherence (i.e. their Continuity is broken by too many references to the main theme of the essay at body-paragraph level). From a Centering Theory point of view, the former paragraphs lack a clear focus at discourse segment level as the center of attention switches back and

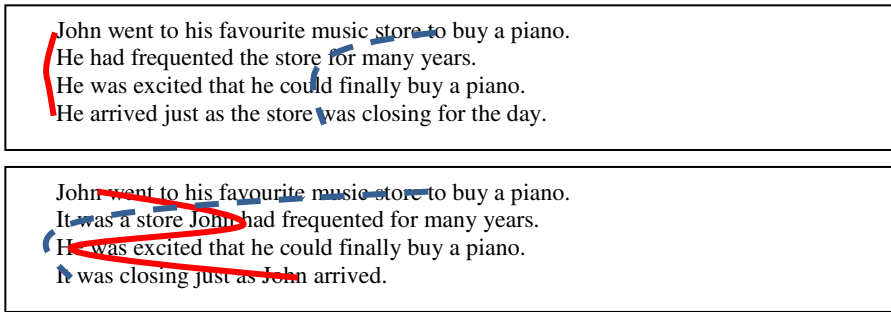


Fig. 3. Examples of two discourse segments presenting the same information in different ways (from [6]). The top paragraph is coherent while the bottom one is less coherent according to Centering Theory as it leads to higher inference loads.

forth between the discourse segment purpose (DSP, i.e. the supporting argument being developed in the segment) and the overall discourse purpose (DP, i.e. the main theme of the essay). Examples of paragraphs with high and low Continuity scores are shown in Figure 2. The top paragraph in the figure has a Continuity score of 5.5 (on a 1-6 scale) while the bottom one has a Continuity score of 2. A centering analysis of the two paragraphs (a center is considered a word appearing in the main subject position of each sentence) reveals that the high Continuity paragraph at the top of the figure has a clear center, i.e. *workers*. The low Continuity paragraph switches between two centers *science-ideas-scientists* to *dreams and imagination*.

Expert raters seem to be generous when grading essays with respect to local coherence as long as the intentions of the speaker are clear (good global coherence) and the essay are contents rich (enough information expressed with a sizeable vocabulary). Such expert raters seem to be able to untangle the intertwined ideas inside discourse segments with poor local coherence without much effort and therefore do not see the need to penalize the student-writers significantly.

To distinguish the two types of discourse structures, we advance the idea of automatically identifying the degree of centering in essay's paragraphs. Centers are concepts appearing in prominent syntactic roles, e.g. subjects, of sentences. We considered several instances of this basic idea. In one instance, we consider concepts occurring only in subject positions of the main clause. In another instance, we consider concepts occurring in subject roles in any clause. Furthermore, another parameter that leads to two other instances is whether to consider the opening and closing sentences of paragraphs which in essay's body paragraphs usually refer to the main theme of the essay and not necessarily to the supporting argument which is the center of the paragraph.

Besides the mentioned benefit to AES system developers, the proposed method to assess local coherence of essays could inform the development of objective criteria to be incorporated in scoring rubrics which in turn can be used to train human raters.

Furthermore, the proposed ideas could be used in instructional settings to teach students to write quality essays that are locally and globally coherent.

5 Experimental Setup and Results

This section presents the experiments we conducted to study the relationship between local coherence, as defined by Centering Theory, and human measurements of essay quality such as Continuity. The basic idea is to see if paragraphs that are centered, i.e. have one central concept, do correspond to high scores of Continuity and vice versa. We present results for several instances of this basic idea as explain next.

First, we will vary the way we consider candidate centers. Given a paragraph, centers can be considered either words in the subject position of the main verb of the main clause of a sentence or words that are subjects in any clause of a sentence. Second, instead of selecting centers as individual words we can select centers as sets of related words. For instance, if the words *couple* and *parents* appear in subject positions in two consecutive sentences they more or less refer to the same center (for space reasons we do not show the actual paragraph from which these example words were chosen). In such cases, we propose to use word-to-word similarity measures to decide which candidate centers should be collated together.

To illustrate the details of our basic approach, we will use the example paragraph at the top of Figure 3. Given such a paragraph, we detect first the words that are nominal subjects, i.e. the subjects of the main verbs of each sentence, using a dependency parser and then generate a center-matrix – see Table 2 – in which there is one column for each nominal subject and one row for each sentence in the paragraph.

Table 2. Example of center-matrix for the top paragraph in Figure 3

	<i>Workers</i>	<i>They</i>	<i>Worker</i>	<i>Employers</i>
Sentence 1	1	0	0	0
Sentence 2	0	1	0	0
Sentence 3	0	1	0	0
Sentence 4	0	0	0	1
Sentence 5	0	0	1	0

An ideal center-matrix corresponding to an ideally centered paragraph would have a column filled with 1s or something close to that. In our example, the presence of pronouns or related words referring to the same center (coreferents) complicates the derivation of the matrix calling for a more general approach. Such an approach would link a center to its pronominal and also nominal referents. The more general approach would collate together different columns in the matrix that refer to the same center. The center-matrix obtained initially from the example paragraph should be transformed into a simplified matrix where columns referring to the dominant center, *workers/they/worker* in our case, are collated together in a single column (see Table 3). We initially used the BART coreference resolution system [20] to collate columns together. The results were not close to our expectations which determined us to use semantic similarity between words. That is, for each two columns in the center-matrix we compute a similarity score using Latent Semantic Analysis (LSA; [9]). If the similarity is above a certain threshold (.30 – determined empirically on a subset of the essay corpus), the corresponding columns are collated.

Table 3. Example of collated center-matrix for the top paragraph in Figure 3. The columns corresponding to *workers*, *worker*, and *they* in Table 2 were collated together.

	<i>Workers</i>	<i>Employers</i>
Sentence 1	1	0
Sentence 2	1	0
Sentence 3	1	0
Sentence 4	0	1
Sentence 5	1	0

Given the collated matrix, for each new center, i.e. column, we compute a dominance score which is the percentage of sentences in the paragraph in which it occurs as subject. To compute the percentages, we normalize by the largest paragraph in our collection to avoid bias towards short paragraphs. The bias consists of short paragraphs, say of two sentences, resulting in high dominance percentages even though the center occurs only in one sentence (out of two).

We present results with several variations of this basic idea. We use nominal subjects versus all subjects and no-collation versus collation of columns.

Results. To evaluate our method to automatically detect local coherence in short essays, we compared the output of the proposed method with human judgments of Continuity. We selected a subset of our original essay corpus to test our ideas. This was necessary as essay that are extremely poor may contain essays with a single very long paragraph or many very short paragraphs (≤ 2 sentences). Because our focus is on body paragraphs, which means paragraphs in the main body of the essay besides the introductory and conclusion paragraphs, and paragraphs which have more than the introductory and concluding sentences, i.e. having more than two sentences, we dropped in our analysis those essays that do not meet these criteria. This way, we were left with 171 essays out of the 184 original essays. Each of the remaining essays was split into individual paragraphs and for each paragraph several center-matrices were generated, one for each of the variants of the basic idea. From the matrices, we derived dominance percentages for each of the candidate centers, i.e. columns.

Once the dominance percentage for each paragraph has been obtained, we conducted an one-way analysis of variance (ANOVA) with the local coherence as the factor and human judgments of Continuity as the grouping variable. Continuity scores were used to map essay into three groups: group A included essays with Continuity scores 1-2 which correspond to low local coherence, group B included essays with Continuity scores 3-4 (intermediate local coherence), and group C included essays with Continuity scores 5-6 (high local coherence). The one-way analysis of variance would tell us whether dominance percentages, which capture centering of paragraphs, are statistically different among the three levels of local coherence (low, intermediate, and high). This in turn would inform us whether dominance percentages can be used as a predictor of local coherence. Table 4 provides a summary of the results of the analysis for the four instances of our basic method. Across all methods, differences between groups were noticed primarily for essays levels with average overall essay scores (SAT scores of 3, 3.5 and 4 which included 87 essays in total). From the table,

Table 4. Overall results of the automated method for local coherence and its variants

<i>Method</i>	<i>F(85, 2)</i>	<i>Significance (p-value)</i>
NominalSubjects	2.096	0.129
NominalSubjects+Collation	3.126	0.014
AllSubjects	1.877	0.227
AllSubjects+Collation	2.347	0.149

we can see that the statistically significant (at $p < .05$ level) method is the one using only nominal subjects with collation based on semantic similarity between subject words. This result in favor of nominal subject provides support for the current view that an utterance in Centering Theory corresponds to a sentence as opposed to certain types of clauses (see [12]). The ANOVA only tells us that at least two of the groups are significantly different but not which ones. A post-hoc analysis showed that the only significantly different groups were the low and high local cohesion groups (i.e. groups A and C in our notation above). That is, the ANOVA and post-hoc analysis revealed that centering could differentiate between average essays with low and high local coherence for the body paragraphs. We have not used this finding further in a prediction model for local coherence as the subset of 87 essays for which the differences were observed to be significant was not large enough for a training-test split. We plan to address this issue in the future by collecting more essays. Furthermore, we plan to improve the way we collate centers together by addressing issues such as pronoun resolution which word-to-word similarity based collation of centers does not address. An alternative would be to use a high-quality coreference resolution engine in which case there is no need to rely on word-to-word similarity.

6 Conclusions

We have explored in this paper the relation between local coherence and human judgments of local coherence called Continuity. We designed an automated method to characterize essays with low, medium, and high Continuity. An analysis of variance analysis revealed that indeed our proposed measure of local coherence is statistically different across the three levels of local coherence with the the differences being significant only for average essays (holistic SAT scores of 3, 3.5, and 4).

Our proposed methods for measuring automatically the local coherence of essays' discourse offers a new instrument which could be integrated in automated essay scoring (AES) systems to better score and provide feedback during writing assessment and instruction. We plan to automate the proposed thread-based analysis, validate it on large corpus of essays, and integrate it in our writing strategy training tutoring system.

Acknowledgments. This research was supported in part by Institute for Education Sciences under awards R305A100875, R305A100875, and R305A080589. Any opinions, findings, and conclusions or recommendations expressed in this material are solely the authors' and do not necessarily reflect the views of the sponsoring agencies.

References

1. Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., Braden-Harder, L., Harris, M.D.: Automated Scoring Using A Hybrid Feature Identification Technique. In: Proceedings of ACL, pp. 206–210 (1998)
2. Burstein, J., Marcu, D., Knight, K.: Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 32–39 (January/February 2003)
3. Crossley, S.A., McNamara, D.S.: Cohesion, Coherence, and Expert Evaluations of Writing Proficiency. In: Proceedings of the 32nd Annual Conference of the Cognitive Science Society (2010)
4. Graesser, A.C., McNamara, D.S., Louwerse, M.M.: What do readers need to learn in order to process coherence relations in narrative and expository text. In: Sweet, A.P., Snow, C.E. (eds.) *Rethinking Reading Comprehension*, pp. 82–98. Guilford Publications, New York (2003)
5. Grosz, B.J., Sidner, C.L.: Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3), 175–204 (1986)
6. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* 21(2), 202–225 (1995)
7. Halliday, M.A.K., Hasan, R.: *Cohesion in English*. Longman, London (1976)
8. Jurafsky, D., Martin, J.H.: *Speech and Language Processing*. Prentice Hall (2009) ISBN: 0131873210
9. Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.): *Latent Semantic Analysis: A Road to Meaning*. Lawrence Erlbaum (1998) (2004)
10. McNamara, D.S., Kintsch, E., Butler-Songer, N., Kintsch, W.: Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction* 14(1), 1–43 (1996)
11. Miltsakaki, E., Kukich, K.: Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering* 10(1) (2004)
12. Miltsakaki, E.: Dissociating discourse salience from information structure: Evidence from a centering study in Modern Greek and Japanese. In: *Computational Linguistics in the Netherlands, CLIN 1999* (1999)
13. Mann, W.C., Thompson, S.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text* 8(3), 243–281 (1988)
14. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008 (2008)
15. Shermis, M.D., Burstein, J.: *Automated Essay Scoring: A Cross Disciplinary Perspective*. Lawrence Erlbaum Associates, Mahwah (2003)
16. Wolf, F., Gibson, E.: Representing discourse coherence: a corpus-based study. *Computational Linguistics* 31, 249–287 (2005)
17. Burstein, J., Marcu, D., Knight, K.: Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, 32–39 (January/February 2003)
18. Page, E.B.: The imminence of grading essays by computer. *Phi Delta Kappan* 48, 238–243 (1966)
19. Passonneau, R., Litman, D.: Discourse segmentation by human and automated means. *Computational Linguistics* 23(1), 103–139 (1997)
20. 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: Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008 (2008)

A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra-sentence Discourse Segments in Spanish

Iria da Cunha¹, Eric SanJuan², Juan-Manuel Torres-Moreno^{2,3},
M. Teresa Cabré¹, and Gerardo Sierra⁴

¹ Institut Universitari de Lingüística Aplicada - Universitat Pompeu Fabra
C/ Roc Boronat, 138, 08018 Barcelona, Spain

² Laboratoire Informatique d'Avignon - Université d'Avignon et des Pays de Vaucluse
339 chemin des Meinajaries, BP91228 84911 Avignon Cedex 9, France

³ École Polytechnique de Montréal - Département de génie informatique
CP 6079 Succ. Centre Ville H3C 3A7 Montréal (Québec), Canada

⁴ Instituto de Ingeniería - Universidad Nacional Autónoma de México
Torre de Ingeniería, Basamento, Ciudad Universitaria, Mexico D.F. 04510 Mexico
{iria.dacunha, teresa.cabre}@upf.edu,
{eric.sanjuan, juan-manuel.torres}@univ-avignon.fr,
GSierraM@iingen.unam.mx

Abstract. Nowadays automatic discourse analysis is a very prominent research topic, since it is useful to develop several applications, as automatic summarization, automatic translation, information extraction, etc. Rhetorical Structure Theory (RST) is the most employed theory. Nevertheless, there are not many studies about this subject in Spanish. In this paper we present the first system assigning nuclearity and rhetorical relations to intra-sentence discourse segments in Spanish texts. To carry out the research, we analyze the learning corpus of the RST Spanish Treebank, a corpus of manually-annotated specialized texts, in order to build a list of lexical and syntactic patterns marking rhetorical relations. To implement the system, this patterns' list and a discourse segmenter called DiSeg are used. To evaluate the system, it is applied over the test corpus of the RST Spanish Treebank. Automatic and manual rhetorical analyses of each sentence are compared, by means of recall and precision, obtaining positive results.

Keywords: Nuclearity, Rhetorical Relations, Intra-sentence Discourse Segments, Rhetorical Structure Theory, Corpus, Symbolic Approach, Spanish.

1 Introduction¹

Nowadays, several examples of indispensable and useful Natural Language Processing (NLP) tools exist: grammar checkers, stemmers, syntactical parsers,

¹ This work has been partially financed by the Spanish projects RICOTERM (FFI2010-21365-C03-01) and APLE (FFI2009-12188-C05-01), and the Mexican CONACYT project 82050.

semantic parsers, among several others. These diverse linguistic levels of data processing have allowed the development of intelligent and useful applications. Nevertheless, as Hovy (2010) mentions, one of the most difficult phenomena to process is discourse structure. Most of the discourse NLP tools are based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). This is a language independent theory based on the idea that a text can be segmented into Elementary Discourse Units (EDUs) linked by means of nucleus-satellite or multinuclear rhetorical relations. In the first case, the satellite gives additional information about the other one, the nucleus, on which it depends (ex. Result, Condition or Concession). In the second case, several elements, all nuclei, are connected at the same level, that is, there are no elements dependent on others and they all have the same importance with regard to the intentions of the text author (ex. Contrast, List or Sequence). RST has been used to develop several applications, like automatic summarization, information extraction (IE), text generation, question-answering, automatic translation, sentence compression, coherence evaluation, etc. (Taboada and Mann, 2006). Nevertheless, most of these works have been developed for English, Portuguese or Japanese. This is due to the fact that at present RST parsers are available only for these three languages. We consider that it is necessary to build a RST Spanish parser, useful for the development of several applications related to computational linguistics.

The rhetorical analysis of a text by means of RST includes three phases: a) segmentation, b) detection of relations and c) building of a hierarchical rhetorical tree of the text. Figure 1 exemplifies these three phases. The relations annotation process is the following: once a text is segmented, rhetorical relations between EDUs are detected. The order of relations detection is the following one: 1) EDUs inside the same sentence are linked in a binary way, 2) sentences inside the same paragraph are linked and 3) paragraphs are linked. This paper focuses on phase b) and c) of the analysis, but specifically on the first level of relations detection, that is, at an intra-sentence level. For phase (a), a discourse RST segmenter for Spanish called DiSeg (da Cunha et al., 2010, 2012) is used. In this paper, the development of the first automatic

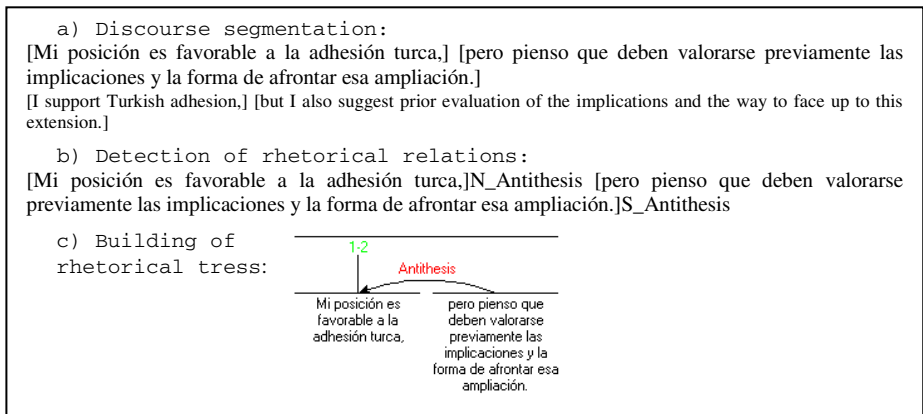


Fig. 1. Example of the three discourse analysis phases

system assigning nuclearity and rhetorical RST relations to intra-sentence discourse segments in Spanish texts is presented. We consider that this system constitutes an important step in order to develop a complete discourse parser for Spanish.

In Section 2, related work is presented. In Section 3, the resources and tools that have been used in this work are explained. In Section 4, corpus analysis is detailed. In Section 5, system implementation is shown. In Section 6, experiments and evaluation are presented. In Section 7, some conclusions and future work are established.

2 Related Work

With regard to discourse RST segmenters, they exist for English (Soricut and Marcu, 2003; Tofiloski et al., 2009), Brazilian Portuguese (Maziero et al., 2007), French (Afantenos et al., 2010) and Spanish (da Cunha et al., 2010, 2012). They require some syntactic analysis of the sentences. The segmenters developed by Soricut and Marcu (2003), Mazeiro et al. (2007), Tofiloski et al. (2009) and da Cunha et al. (2010, 2012) rely on a set of linguistic rules. The one by Afantenos et al. (2010) relies on machine learning techniques: it learns rules automatically from thoroughly annotated texts.

Regarding RST corpora, nowadays they exist only for 4 languages: English (Carlson et al., 2002; Taboada and Renkema, 2008), German (Stede, 2004), Portuguese (Pardo and Nunes, 2008; Pardo and Seno, 2005) and Spanish (da Cunha et al., 2011). These RST corpora suppose an important step on the RST research and they have been very useful to develop several applications, like information extraction, text generation, automatic summarization, etc. They have differences related to the number of included texts and words, the annotation systematicity, the texts' domain heterogeneity, the amount of double-annotated texts (to measure the agreement between annotators), etc. Most of these corpora have been annotated using the annotation interface RSTtool (O'Donnell, 2000).

Finally, discourse RST parsers for some languages are available: for English (Marcu, 2000; Soricut and Marcu, 2003; Subba and Di Eugenio, 2009), Japanese (Sumita et al., 1992) and Brazilian Portuguese (Pardo and Nunes, 2008). These discourse parsers use symbolic or statistical approaches. Some of them, besides some limited assumptions, achieve near human performance (see, e.g., Soricut and Marcu [2003], for sentence level analysis). Nevertheless, at the moment, there is not a discourse parser for Spanish.

3 Ressources and Tools

In this section, the resources and tools that have been used in this work are presented.

3.1 DiSeg

DiSeg (da Cunha et al., 2010, 2012) is the only discourse segmenter existing for Spanish. It produces state of the art results while it does not require syntactic analysis but only shallow parsing with a reduced set of linguistic rules that insert segment boundaries into the sentences. Therefore it can be easily included in applications

requiring fast text analysis on the fly. Thus, this segmenter has been chosen for this work because of this reason.

DiSeg segmentation criteria are very similar to the ones used by da Cunha and Iruskieta (2010):

- a) All the sentences of the text are segmented as EDUs.
- b) Intra-sentence EDUs are segmented, using the following criteria:
 - b1) An intra-sentence EDU has to include a finite verb, an infinitive or a gerund.
 - b2) Subject/object subordinate clauses or substantive sentences are not segmented.
 - b3) Subordinate relative clauses are not segmented.
 - b4) Elements in parentheses are only segmented if they follow the criterion b1.

DiSeg performance was evaluated over a corpus of manually annotated texts, obtaining an F-score between 80% and 96% in experiments with a corpus containing medical texts, and an F-Score of 91% with a corpus of texts about terminology.

3.2 RST Spanish Treebank

The RST Spanish Treebank (da Cunha et al., 2011) is the only corpus including texts in Spanish annotated with rhetorical relations of RST. It is free for research purposes and it can be consulted or downloaded by means of an on-line interface. This is the main reason to use this corpus in this research. The corpus contains texts from nine specialized domains (Astrophysics, Earthquake Engineering, Economy, Law, Linguistics, Mathematics, Medicine, Psychology and Sexuality). The used segmentation criteria are similar to those employed by DiSeg. It includes 52,746 words, 267 texts, 2,256 sentences and 3,349 discourse segments. The 79% of the corpus was tagged by 1 person (called “training corpus”), from a team of 10 RST expert annotators. There is a 31% of the corpus double-annotated (called “test corpus”), with high agreement regarding EDUs, SPANs (that is, a group of related EDUs), nuclearity and relations. The annotation interface used was the RSTtool. The list of rhetorical relations employed to manually annotate the texts is included in Table 1. In this table, the quantity of relations of the training corpus of the RST Spanish Treebank is shown. In this work, this is the corpus analyzed in order to detect linguistic patterns for relations detection, explained in Section 4.

Table 1. Quantity of relations in the training corpus

Relation name	Type	Quantity of relations	
		Nº	%
Elaboration	N-S	508	24.71
Preparation	N-S	273	13.28
Background	N-S	152	7.39
Result	N-S	118	5.74
Means	N-S	108	5.2
Purpose	N-S	108	5.2

Table 1. (*Continued*)

List	N-N	106	5.16
Joint	N-N	85	4.13
Circumstance	N-S	86	4.18
Interpretation	N-S	69	3.36
Antithesis	N-S	53	2.58
Cause	N-S	50	2.43
Sequence	N-N	44	2.14
Condition	N-S	44	2.14
Evidence	N-S	43	2.09
Contrast	N-N	41	1.99
Concession	N-S	40	1.95
Justification	N-S	38	1.85
Solution	N-S	20	0.97
Motivation	N-S	16	0.78
Reformulation	N-S	13	0.63
Conjunction	N-N	10	0.49
Evaluation	N-S	8	0.39
Summary	N-S	8	0.39
Disjunction	N-N	5	0.24
Enablement	N-S	5	0.24
Otherwise	N-S	3	0.15
Unless	N-S	2	0.10
TOTAL		2056	100

As it can be observed, it contains more than 20 examples of most of the relations. The exceptions are the nucleus-satellite relations of Enablement, Evaluation, Summary, Otherwise and Unless, and the multinuclear relations of Conjunction and Disjunction, because it is not so usual to find these rhetorical relations in the language, in comparison with others.

4 Corpus Analysis

The first step to carry out a list of linguistic patterns useful to detect discourse relations automatically was to build a table including all the identifiers of the texts of the training corpus and all the analyzed RST relations (Table 1). The second step was to analyze manually all the relations of these texts, observing all the possible lexical or syntactic markers indicating the presence of a RST relation (not only traditional discourse markers were analyzed). Then, these markers were included in the table. Table 2 shows a sample of this table (the complete table is too long, so it is not possible to include it here). In the first column, the relation name is included; in the second and third columns, the text identifier (ID) is shown (for example, ec00009 indicates that it is the text 9 of the economic subcorpus), besides the linguistic

markers found in each text. These markers contain the Spanish form and their translation to English, with their number of occurrences in the corresponding text.

Once all the linguistic markers were analyzed, they were divided into 3 categories:

1. Traditional discourse markers, as for example *primero* (“first”), *segundo* (“second”), *si* (“if”), *ya que* (“since”), etc. The classification of Spanish discourse markers by Portolés (1998) was used.
2. Markers including lexical units, specifically, substantives and verbs, as for example *metodología* (“methodology”), *procedimiento* (“procedure”), *usar* (“to use”), etc.
3. Markers including verbal structures, as for example *para* + *INFINITIVO* (“to + INFINITIVE”), *siempre que* + *SUBJUNTIVO* (“provided that + SUBJUNCTIVE”), etc.

Table 2. Example of the table including texts identifiers, RST relations and linguistic makers

Rel./ ID	ec00009	li00035
Purpose	<i>con el fin de</i> (1) “with the aim of”	<i>para</i> + <i>INF</i> (3) “to + INF”-
Cause	<i>al</i> + <i>INF</i> (1) “about to + INF”	<i>ya que</i> (2) “since <i>debido a</i> (1) “due to”
Sequence	<i>primero</i> (1) “first	-
Means	<i>metodología</i> (1) “methodology” <i>procedimiento</i> (1) “procedure”	<i>método</i> (1) “method <i>VB usar</i> (2) “VB to use”
Result	-	<i>resultado</i> (2) “result”
Condition	<i>siempre que</i> + <i>SUBJ</i> (2) “provided that + SUBJ”	<i>si</i> (2) “if”

Linguistic markers of Elaboration relations were not analyzed, since this is the most general and frequent relation in the language. As it is explained in Section 5, our algorithm includes a rule assigning the Elaboration relation to all the EDUs that have not been categorized under any RST relation, so it was not really necessary to analyze the linguistic markers of this relation. Preparation relation is not analyzed either, since, in the corpus, Preparation satellites are always document titles, so they do not contain any marker (the only marker is that they are sentences without verb).

After the analysis, 778 general markers were detected. The training corpus includes 2056 RST relations; therefore this means that the 37.9% of the relations of this corpus is marked. This percentage is higher than the one included in other works (20-25% for Spanish and English; see Taboada [2006], and Iruskieta and da Cunha [2010]). This is due to the fact that, usually, works about discourse markers and relations only consider traditional markers. In our work, any relevant lexical or syntactic unit can be considered as a marker. Table 3 includes statistics about some marked relations in the training corpus.

Table 3. Some statistics about marked relations in the training corpus

Relation name	Type	N° of linguistic markers	
		N°	%
Background		75	49.34
Purpose		95	87.96
Result		55	46.61
Means		44	40.74
Antithesis		43	81.13
Cause		32	64
Sequence		31	70.45
...	
Reformulation		6	46.15
Summary		2	25
Disjunction		4	80
Unless		0	0
TOTAL		2056	100

It can be observed that some relations have a very high percentage of marks (for example Purpose, Antithesis or Sequence), even if they have not a high number of occurrences in the corpus (as for example Disjunction). This means that these relations appear in texts evidenced by means of a linguistic marker quite frequently. These relations are the easiest ones to detect automatically. Nevertheless, there are some relations with a medium percentage or markers, as for example Result, Means or Reformulation, that are more difficult to detect. The most extreme case is the relation of Unless, which has 2 occurrences in the corpus and no markers. These cases are the most difficult ones to detect automatically. Hovy (2010) states that, given the lack of examples in the corpus, there are 2 possible strategies: a) to leave the corpus as it is, with few or no examples of some cases (but the problem will be the lack of training examples for machine learning systems), or b) to add low-frequency examples artificially to “enrich” the corpus (but the problem will be the distortion of the native frequency distribution and perhaps the confusion of machine learning systems). In the current state of our project, we have chosen the first option.

Another difficult that has been detected is the possible ambiguity of some markers. As van Dijk (1984) explains, one of the problems of the semantics of natural connectors is that the same connector can express different connection types and one connection type can be expressed by several connectors. Exactly, Versley (2011) carries out a NLP research about the ambiguity of temporal discourse markers in English. In the analyzed corpus, some cases of ambiguous markers were found. For example, with regard to the third type of markers (Markers including verbal structures), it is found that the verbal form “GERUND” can evidence 3 different relations: Circumstance (10 occurrences), Means (6 occurrences) and Result (2 occurrences). In Section 5 it is explained how this problem is treated in this work.

5 System Implementation

In this section, the algorithm to detect RST relations and nuclearity at an intra-sentence level is explained. The algorithm architecture has 3 main stages:

- Given a text, *sentence segmentation* (using a typical sentence segmenter).
- Given a sentence, *discourse segmentation* in EDUs (using the discourse segmenter DiSeg).
- Given a sentence segmented in EDUs, *application the following list of rules* (in this order):
 - PHASE 1: Traditional discourse markers (155 rules).
 - PHASE 2: Markers including lexical units (77 rules).
 - PHASE 3: Markers including verbal structures (41 rules).
 - PHASE 4: Elaboration rule (1 rule).

Moreover, these 4 types of rules have to be applied taking into account the following order:

1. Nucleus-Satellite (N-S) relations rules.
2. Satellite-Nucleus (S-N) relations rules.
3. Multinuclear relations rules (N-N).

Let's see some examples or rules:

PHASE 1:

- NUCLEUS-SATELLITE (N-S) RULES:
 - *sin embargo* ("nevertheless") => Satellite of Antithesis of the previous EDU
- SATELLITE-NUCLEUS (S-N) RULES:
 - *ya que* ("since") => Satellite of Cause of the next EDU
- MULTINUCLEAR (N-N) RULES:
 - *en primer lugar* ("in the first place") => Nucleus of Sequence of the next EDU

PHASE 2:

- NUCLEUS-SATELLITE (N-S) RULES:
 - *como alternative* ("as alternative") => Satellite of Otherwise of the previous EDU
- SATELLITE-NUCLEUS (S-N) RULES:
 - *esto es evidencia de que* ("that is an evidence of") => Satellite of Evidence of the previous EDU

PHASE 3:

- NUCLEUS-SATELLITE (N-S) RULES:
 - *siempre que + SUBJUNTIVO* ("always that + SUBJUNTIVE") => Satellite of Condition of the previous EDU
- SATELLITE-NUCLEUS (S-N) RULES:
 - *aun + GERUND* ("although + GERUND") => Satellite of Concession of the previous EDU

The algorithm (implemented in Perl) reads each sentence from left to right, and the first rule that it finds is applied. Rules order is important, because the most confident

rules are the ones included in Phase 1, that is, the rules based on traditional markers. After the application of the rules included in Phases 1-3, the Elaboration rule is applied in Phase 4. This rule assigns the category “Satellite of Elaboration” to all the EDUs that have not been categorized in Phases 1-3. It is linked with the previous EDU, which will be its Nucleus:

- if no relation was detected between 2 EDUS => Satellite of Elaboration of the previous EDU

The implemented rules are based on most of the linguistic markers detected. Some markers were not used because they were too general. For, example, it was detected that the syntactic structure “SUBJECT + VERB TO BE + OBJECT” marks the Background relation frequently. Nevertheless, this structure is so common in the language, that it was decided not to use it as marker.

To deal with the problem of markers ambiguity, 3 strategies could be used: a) to choose the relation with more number of markers of this type, b) to give to the algorithm all the possible relations or c) to develop more fine-grained strategies combining several markers to try to choose only one relation. In the current state of our project, in most of the cases, the first option is chosen. Nevertheless, in the future, strategy 3 will be explored deeply, in the same line that Versley (2011). At the moment, we have developed a few rules of this kind. For example, the marker *mientras* (“while”) can express at the same time Contrast, Circumstance and Condition or. See the following examples:

- a) [**Mientras** a mí me encanta bailar]EDU1 [a ella le gusta cantar.]EDU2
 [While I love dancing.]EDU1 [she loves singing.]EDU2
- b) [**Mientras** me dijo lo que quería escuchar]EDU1 [estuve tranquilo.]EDU2
 [While he told me what I wanted to listen]EDU1 [I was tranquil.]EDU2
- c) [**Mientras** me digas lo que quiero escuchar]EDU1 [todo irá bien]EDU2
 [While you tell me what I want to listen]EDU1 [everything will be ok.]EDU2

On the one hand, example (a) contains 2 Nucleus of Contrast. On the other hand, the EDU1 of example (b) is a satellite of Circumstance of EDU2, while the EDU1 of example (c) is a satellite of Condition of EDU2. The 3 examples contain the same marker, but its meaning is different. In our rules, this marker is used to evidence that all these relations could be possible, but verbal information is given as well to differentiate automatically between both. For example, in Spanish, if the meaning of the relation is Condition, the main verb in EDU1 should be a form in SUBJUNCTIVE (as “digas”). This information is given to our rules, in order to manage some cases of makers’ ambiguity.

6 Experiments and Evaluation

After designing and implementing the algorithm, it was applied over a subcorpus of the test corpus of the RST Spanish Treebank. The mathematics corpus was used. It includes 48 research articles about mathematics, published in scientific journals. Therefore, the texts are very specialized and they contain formulae, numbers, specialized phraseological units, name entities, etc. This means that the treatment of

this kind of texts is difficult. This mathematics corpus test contains 164 sentences. After our analysis, it is observed that the manually-annotated sentences can be divided in two groups: there are 110 sentences that constitute a complete EDU and there are 54 sentences that are divided into some EDUs. In this second group, there are 39 sentences divided into 2 EDUs, 4 sentences divided into 3 EDUs, and 4 sentences divided into 4 EDUs. Regarding the automatically-annotated corpus (using DiSeg and our relations detection system), it is noted that there are 110 sentences that constitute a complete EDU and there are 54 sentences that are divided into some EDUs. In this second group, there are 41 sentences divided into 2 EDUs, 9 sentences divided into 2 EDUs, and 4 sentences divided into 4 EDUs. These statistics mean that the performance of the discourse segmenter DiSeg is quite high. Nevertheless, DiSeg performs an over-segmentation (4.3%) with regard to the manual segmentation. This difference will be one of the limitations of the system presented here.

To evaluate the algorithm, it was applied to the original test corpus, and its results were compared with the manually-annotated corpus. The RST trees comparison methodology by Marcu (2000) was used. This methodology evaluates 4 elements (EDUs, SPANs, Nuclearity and Relations), by means of precision and recall measures. It compares a manual gold standard annotation with an automatic annotation. An on-line automatic tool for RST trees comparison, RSTeval (Maziero and Pardo, 2009) is used, where Marcu's methodology has been implemented (for 4 languages: English, Portuguese, Spanish and Basque). For each of the 164 sentence-trees pair from the test corpus, precision and recall were measured separately. Afterwards, all those individual results were put together to obtain general results. Table 4 shows global results for the 4 categories. Segmentation (EDUs), SPANs and Nuclearity categories obtain recall and precision of 81.75%. Relations category obtains a bit less of precision and recall: 81.73%, but the difference is not high. We think that these results can be considered acceptable. Nevertheless, we cannot compare these results with the results of other systems for nuclearity and relations detection, since, as it has been mentioned, they do not exist for Spanish.

Table 4. Results of the evaluation

Category	Precision	Recall
EDUs	81.75	81.75
SPANs	81.75	81.75
Nuclearity	81.75	81.75
Relations	81.73	81.73

Here, an example of sentence perfectly analyzed by the system is shown:

[Se muestra cómo puede usarse la manipulación directa de las representaciones dinámicas generadas en este ambiente,]EDU1_Nucleus
 [para observar patrones de comportamiento]EDU2_Satellite_Purpose_of_the_previous_EDU
 [y formular conjeturas sobre las transformaciones bajo estudio.]
 EDU3_Nucleus_List_of_the_previous_EDU
 [It is shown how direct manipulation of the dynamic representations generated in this environment can be used]EDU1_Nucleus

[to observe behavior patterns]EDU2_Satellite_Purpose_of_the_previous_EDU
 [and to formulate conjectures on the transformations that we are studying.]
 EDU3_Nucleus_List_of_the_previous_EDU

The obtained results could be considered an extrinsic quantitative evaluation of the discourse segmenter DiSeg. Results show that, although segmentation results are quite good, there are some errors. From 164 sentences of the corpus, there were 26 sentences that DiSeg segmented wrong (that is, 15.85%). These segmentation errors are the cause of some errors of our relations detection system.

After this quantitative analysis, a qualitative analysis was carried out, in order to observe the main limitations of the proposed system, regarding nuclearity and relations. The main differences between manual and automatic annotations were analyzed. With regard to nuclearity, the main error is related to rules' order, which we will have to optimize in the future. For example:

[Sin embargo, ante la amplitud de su obra nos hemos visto obligadas a escoger algunos temas]EDU1_Satellite_Antithesis
 [y hemos dejado de lado otros igualmente importantes.]EDU2_Nucleus
 [However, because of his huge work we had to choose some subjects]
 EDU1_satellite_Antithesis
 [and forget other ones very important as well.]EDU2_Nucleus

In this example, EDU1 includes the marker *sin embargo* (“nevertheless”), so the system interprets that it is a Satellite of Antithesis of EDU2. But the reality is that, although this marker indicates Antithesis, here it is relating the entire sentence (EDU1 + EDU2) with the previous sentence (that does not appear in the example). This is one of the limitations of the intra-sentence analysis, which will be solved in the future. In this case, EDU1 and EDU2 would be related by a relation of List (marked by *y* “and”). Regarding rhetorical annotations, the main reason of error is the mentioned ambiguity of some markers included in the rules. For example, in the next sentence, the relation assigned by the system was Circumstance (marked in the rules by the GERUND verbal form at the beginning of a sentence), instead of Means, the relation assigned by humans:

[Generalizando la idea de palanca a la de un sólido "plano"]EDU1_Satellite_Circumstance
 [obtenemos la definición general de momento estático en un plano, la cual se traduce, fácilmente, en un vector área en el espacio tridimensional.]EDU2_Nucleus
 [Generalizing the idea about lever to that of a "plane"]EDU1_Satellite_Circumstance
 [we obtain the general definition of static moment in a plane, which means, easily, an area vector in the three-dimensional space.]EDU2_Nucleus

7 Conclusions and Future Work

In this work, the first automatic system assigning nuclearity and rhetorical RST relations to intra-sentence discourse segments in texts in Spanish is presented. Precision and recall results are acceptable. We consider that this system constitutes an important step in order to develop a complete discourse parser for Spanish. Moreover, we think that this work means a notable step as well for the general RST research in Spanish, and that the system that is presented here will be useful to carry out diverse researches about RST in this language, from a descriptive point of view (ex. analysis

of texts from different domains or genres) and an applied point of view (development of NLP applications, like automatic summarization, automatic translation, sentence compression, IE, etc.).

As future work, we plan to optimize the rules of the system, since some errors in the rules' performance have been detected. We want to develop some strategies to treat the ambiguity of some markers. With regard to the evaluation, we would like to create a baseline to compare the results of the system. The final goal of our project is to develop the next stages of automatic discourse analysis (relation of sentences and paragraphs), in order to build the first complete discourse parser for Spanish.

References

1. Afantenos, S., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. In: Proceedings of the Conference LREC 2010, pp. 3578–3584 (2010)
2. Carlson, L., Marcu, D., Okurowski, M.E.: RST Discourse Treebank. Linguistic Data Consortium, Pennsylvania (2002)
3. da Cunha, I., Torres-Moreno, J.-M., Sierra, G.: On the development of the RST Spanish Treebank. In: Proceedings of the Fifth Law Workshop (ACL 2011), pp. 1–10 (2011)
4. da Cunha, I., SanJuan, E., Torres-Moreno, J.-M., Lloberes, M., Castellón, I.: Discourse Segmentation for Spanish based on Shallow Parsing. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010. LNCS, vol. 6437, pp. 13–23. Springer, Heidelberg (2010)
5. da Cunha, I., SanJuan, E., Torres-Moreno, J.-M., Lloberes, M., Castellón, I.: DiSeg 1.0: The First System for Spanish Discourse Segmentation. Expert Systems with Applications 39(2), 1671–1678 (2012)
6. da Cunha, I., Irukieta, M.: Comparing rhetorical structures of different languages: The influence of translation strategies. Discourse Studies 12(5), 563–598 (2010)
7. Hovy, E.: Annotation. A Tutorial. Presented at the 48th Annual Meeting of the Association for Computational Linguistics (2010)
8. Irukieta, M., da Cunha, I.: Marcadores y relaciones discursivas en el ámbito médico: un estudio en español y euskera. In: Bueno, J.L., et al. (eds.) Analizar datos > Describir variación: XXVIII Congreso Internacional AESLA, pp. 146–159. Universidade de Vigo, Servizo de Publicacións, Vigo (2010)
9. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243–281 (1988)
10. Marcu, D.: The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. Computational Linguistics 26(3), 395–448 (2000)
11. Maziero, E., Pardo, T.A.S., Nunes, M.G.V.: Identificação automática de segmentos discursivos: o uso do parser PALAVRAS. Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional. Universidade de São Paulo, São Carlos (2007)
12. Maziero, E., Pardo, T.A.S.: Metodologia de avaliação automática de estruturas retóricas. In: Proceedings of the III RST Meeting (7th Brazilian Symposium in Information and Human Language Technology), Brasil (2009)
13. O'Donnell, M.: RSTTOOL 2.4 – A markup tool for rhetorical structure theory. In: Proceed. of the International Natural Language Generation Conference, pp. 253–256 (2000)

14. Pardo, T.A.S., Nunes, M.G.V.: On the Development and Evaluation of a Brazilian Portuguese Discourse Parser. *Journal of Theoretical and Applied Computing* 15(2), 43–64 (2008)
15. Pardo, T.A.S., Seno, E.R.M.: Rhetalho: um corpus de referência anotado retoricamente. In: *Anais do V Encontro de Corpora*. São Carlos-SP, Brasil (2005)
16. Portolés, J.: *Marcadores del discurso*. Ariel, Barcelona (1998)
17. Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. In: *Proceedings of the 2003 Conference of NAACL-HLT*, pp. 149–156 (2003)
18. Stede, M.: The Potsdam commentary corpus. In: *Proceedings of the Workshop on Discourse Annotation, 42nd Meeting of the ACL* (2004)
19. Subba, R., Di Eugenio, B.: An effective discourse parser that uses rich linguistic information. In: *Proceedings of the 2009 Conference of HLT-ACL*, pp. 566–574 (2009)
20. Sumita, K., Ono, K., Chino, T., Ukita, T., Amano, S.: A discourse structure analyzer for Japanese text. In: *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 1133–1140 (1992)
21. Taboada, T.: Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics* 38, 567–592 (2006)
22. Taboada, M., Mann, W.C.: Applications of Rhetorical Structure Theory. *Discourse Studies* 8(4), 567–588 (2006)
23. Taboada, M., Renkema, J.: *Discourse Relations Reference Corpus [Corpus]*. Simon Fraser University and Tilburg University (2008), http://www.sfu.ca/rst/06tools/discourse_relations_corpus.html
24. Tofiloski, M., Brooke, J., Taboada, M.: A Syntactic and Lexical-Based Discourse Segmenter. In: *Proceedings of the 47th Annual Meeting of ACL* (2009)
25. van Dijk, T.A.: *Texto y contexto (Semántica y pragmática del discurso)*. Cátedra, Madrid (1984)
26. Versley, Y.: Multilabel Tagging of Discourse Relations in Ambiguous Temporal Connectives. In: *Proceedings de la 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011)*, pp. 154–161 (2011)

Feature Specific Sentiment Analysis for Product Reviews

Subhabrata Mukherjee and Pushpak Bhattacharyya

Dept. of Computer Science and Engineering, IIT Bombay
{subhabratam,pb}@cse.iitb.ac.in

Abstract. In this paper, we present a novel approach to identify *feature specific* expressions of opinion in product reviews with *different features* and *mixed emotions*. The objective is realized by identifying a set of potential features in the review and extracting opinion expressions about those features by exploiting their associations. Capitalizing on the view that more closely associated words come together to express an opinion about a certain feature, dependency parsing is used to identify relations between the opinion expressions. The system learns the *set of significant relations* to be used by dependency parsing and a *threshold parameter* which allows us to merge closely associated opinion expressions. The data requirement is minimal as this is a *one time learning of the domain independent parameters*. The associations are represented in the form of a graph which is partitioned to finally retrieve the opinion expression describing the *user specified feature*. We show that the system achieves a *high accuracy across all domains* and performs at par with state-of-the-art systems despite its data limitations.

1 Introduction

In recent years, the explosion of social networking sites, blogs and review sites provide a lot of information. Millions of people express uninhibited opinions about various product features and their nuances. This forms an active feedback which is of importance not only to the companies developing the products, but also to their rivals and several other potential customers.

Sentiment Analysis is the task of tapping this goldmine of information. It retrieves opinions about certain products or features and classifies them as *recommended* or *not recommended*, that is *positive* or *negative*.

The sentiment regarding a particular product in a review is seldom explicitly positive or negative; rather people tend to have a mixed opinion about various features, some positive and some negative. Thus the feature specific opinion matters more than the overall opinion.

Consider a review “*I like Micromax’s multimedia features but the battery life sucks.*” This sentence has a mixed emotion. The emotion regarding *multimedia* is positive whereas that regarding *battery life* is negative. Hence, it is of utmost importance to extract only those opinions relevant to a particular feature (like *battery life* or *multimedia*) and classify them, instead of taking the complete sentence and the overall sentiment.

In this work, we propose a method that represents the features and corresponding opinions in the form of a graph where we use dependency parsing to capture the relations between the features and their associated opinions. The idea is to capture the association between any specific feature and the expressions of opinion that come together to describe that feature. This is done by capturing the *short range* and *long range dependencies* between the words using dependency parsing. Clustering is done on the graph to retrieve only those opinion expressions that are *most closely related* to the *target feature* (user specified feature) and the rest are pruned. We apply merging in the final phase of our algorithm to merge the opinions about any 2 features that cannot be described independent of each other. We apply our method to domain specific reviews to test the efficacy of the system. We achieved a high accuracy across all domains over the baseline. We compare our approach with state-of-the-art systems [4] where we achieve a comparable accuracy despite data limitations. The system performance improved greatly not only over the naïve baseline but also over the chosen improved baseline [5].

The roadmap to the remaining part of the paper is as follows:

Section 1 presents the motivation and objective of the current work. Section 2 gives a related work section. Section 3 defines the problem statement. Section 4 gives the algorithm to extract features and their associated opinion expressions. It presents a graph based representation of the features and their relations, which is partitioned to obtain feature specific opinions. A rule-based and supervised classification system is presented in Section 5 to find the final sentiment polarity. We present the learning of the domain independent parameters in Section 6, followed by extensive experiments across various product domains in review blogs to validate our claim in Section 7. Section 8 gives the conclusions and directions for future work followed by references.

2 Related Work

Chen *et. al* [1] use dependency parsing and shallow semantic analysis for Chinese opinion related expression extraction. They categorize relations as, topic and sentiment located in the same sub-sentence and quite close to each other (like the rule “an adjective plus a noun” is mostly a potential opinion-element relation), topic and sentiment located in adjacent sub-sentences and the two sub-sentences are parallel in structure (that is to say, the two adjacent sub-sentences are connected by some coherent word, like although/but, and etc), topic and sentiment located in different sub-sentences, either being adjacent or not, but the different sub sentences are independent of each other, no parallel structures any more.

Wu *et. al* [2] use phrase dependency parsing for opinion mining. In dependency grammar, structure is determined by the relation between a head and its dependents. The dependent is a modifier or complement and the head plays a more important role in determining the behaviors of the pair. The authors want to compromise between the information loss of the word level dependency in dependency parsing as it does not explicitly provide local structures and syntactic categories of phrases and the information gain in extracting long distance relations. Hence they extend the dependency tree node with phrases.”

Hu *et. al* [3] used frequent item sets to extract the most relevant features from a domain and pruned it to obtain a subset of features. They extract the nearby adjectives

to a feature as an *opinion word* regarding that feature. Using a seed set of labeled Adjectives, which they manually develop for each domain, they further expand it using WordNet and use them to classify the extracted opinion words as positive or negative.

Lakkaraju *et. al* [4] propose a joint sentiment topic model to probabilistically model the set of features and sentiment topics using HMM-LDA. It is an unsupervised system which models the distribution of features and opinions in a review and is thus a generative model.

Most of the works mentioned above require labeled datasets for training their models for each of the domains. If there is a new domain about which no prior information is available or if there are mixed reviews from multiple domains inter-mixed (as in *Twitter*), where the domain for any specific product cannot be identified, then it would be difficult to train the models. The works do not exploit the fact that majority of the reviews have a lot of domain independent components. If those domain independent parameters are used to capture the associations between features and their associated opinion expressions, the models would capture majority of the feature specific sentiments with minimal data requirement.

3 Problem Statement

Given a product review containing multiple features and varied opinions, the objective is to extract expressions of opinion describing a *target feature* and classify it as positive or negative. The objectives can be summarized is:

1. Extract all the features from the given review

In the absence of any prior information about the domain of the review (in the form of untagged or tagged data belonging to that domain), this will give a list of *potential features* in that review which needs to be pruned to obtain the exact features.

Consider the review, “*I wonder how can any people like Max, given its pathetic battery life, even though its multimedia features are not that bad.*”

Here, *multimedia features* and *battery life* are the exact features pertaining to the *mobile domain*. But without any prior domain information, we can use an approximate method to obtain a list of potential features that may include other noisy features as well, example *people*. So this list needs to be pruned to remove the noise and obtain the exact set of features.

2. Extract opinion words referring to the target feature

The opinion words are not only Adjectives like *hate*, *love* but also consist of other POS categories like Nouns (*terrorism*), Verbs (*terrify*) and Adverbs (*gratefully*). A naïve method, like extracting the opinion words closest to the *target feature*, does not work so well when the sentence has multiple features and distributed emotions (as we will see later).

In the example above, *pathetic* and *not bad* are the opinion expressions referring to *battery life* and *multimedia features* respectively.

3. Classify the extracted opinion words as positive or negative

This step will mark *pathetic* as a negative opinion and *not bad* as a positive opinion.

4 Feature Specific Sentiment Analysis

In this section, we will first outline a method to extract features and their associated relations.

4.1 Feature Extraction

We will elaborate 2 methods for extracting features corresponding to the availability of domain knowledge.

4.1.1 Feature Extraction in Absence of Domain Knowledge

In the absence of any prior information about the product domain, we can make a list of *potential features* in the review by constraining the features only to be Nouns (Example: *multimedia, firmware, display, color* etc.). All the words in the sentence are POS-tagged and all the Nouns are retrieved. Initially, all the Nouns are treated as features and added to the *feature list F*.

Consider the review,

“*I have an ipod and it is a great buy but I'm probably the only person that dislikes the iTunes software.*”

$F = \{ipod, buy, person, software\}$

This forms our initial feature set. But the intended features are *ipod* and *software* as they are the features specific to the *mobile domain*. We will later present an algorithm to prune this initial feature set, such that any 2 features *strongly related* will be merged. Thus, *buy* will be merged with *ipod* when the target feature is *ipod*, and $\{person, software\}$ will be pruned. If the target feature is *software*, *person* will be merged with *software*, and $\{ipod, buy\}$ will be pruned.

4.1.2 Feature Extraction in Presence of Domain Knowledge

If domain information is available (in the form of crawled reviews from the domain in focus, when the product domain has been identified) we can extract all the features in the domain using *Latent Dirichlet Allocation* (Blei *et. al* [6]) or *HMM-LDA* (Griffiths *et. al* [7]). In presence of domain knowledge, we would readily know that *software* and *ipod* are mobile-domain specific features whereas *buy* and *person* are not. Using this information we can directly prune the feature list *F*.

4.2 Relation Extraction

Relation extraction is necessary to identify the associations between the opinion expressions in a review. We will shortly formulate our hypothesis that necessitates this phase. We identify two kinds of relations between the words in a sentence that associate them to form a coherent review:

1. Direct Neighbor Relation

Let *Stopwords* be the list of pre-compiled stop words occurring frequently in any text. This comprises mainly of *be* verbs, personal pronouns, prepositions, conjunctions etc. All Nouns, Adjectives, Adverbs, Verbs (except *be* verbs) are excluded from the list.

Consider a sentence S and 2 consecutive words $w_i, w_{i+1} \in S$. If $w_i, w_{i+1} \notin \text{Stopwords}$, then they are directly related. This helps us to capture *short range dependencies*.

2. Dependency Relation

Let *Dependency_Relation* be the list of significant relations. We call any *dependency relation* significant, if

- It involves any *subject, object* or *agent* like *nsubj, dobj, agent* etc
- It involves any *modifier* like *advmod, amod* etc
- It involves *negation* like *neg*
- It involves any *preposition* like *prep_of*
- It involves any *adjectival or clausal* component like *acomp, xcomp*

The above set of relations is not minimal, in the sense that not all of them are equally significant in capturing the semantic coherence in reviews. We will later show how to prune the above set of relations, to obtain a minimal set of significant relations, by a small seed set of data using ablation test.

Any 2 words w_i and w_j in S are directly related, if

$$\exists D_i \text{ s.t. } D_i(w_i, w_j) \in \text{Dependency_Relation} .$$

This helps us to capture *long range dependencies*.

The direct neighbor and dependency relations are combined to form the master *relation set R*.

We now formulate the following hypothesis:

More closely related words come together to express an opinion about a feature.

If there are ' n ' features of a product in a sentence, then those words that are most closely related (in terms of relations defined above) to a feature ' i ' will come together to express some opinion about it, rather than about some other feature ' j ', to which they are not so closely associated.

For Example: "*I want to use Samsung which is a great product but am not so sure about using Nokia*".

Here $\{\textit{great, product}\}$ are related by an adjectival modifier relation, and $\{\textit{product, Samsung}\}$ are related by a relative clause modifier relation. Thus $\{\textit{great, Samsung}\}$ are transitively related. **Here $\{\textit{great, product}\}$ are more closely related to Samsung than they are to Nokia.** Thus $\{\textit{great, product}\}$ come together to express an opinion about the entity "Samsung" than about the entity "Nokia". The adjectival relation is important as it associates the opinion *great* with *product* and the relative clause modifier relation is significant as it associates *product* with *Samsung*.

These relations are provided by the Dependency Parser. We used the Stanford Dependency Parser (<http://nlp.stanford.edu:8080/parser/index.jsp>).

4.3 Graph Representation

Given a sentence S , let W be the set of all words in the sentence S .

A Graph $G(W, E)$ is constructed such that any $w_i, w_j \in W$ are directly connected by $e_k \in E$, if $\exists R_l$ s.t. $R_l(w_i, w_j) \in R$.

In other words, in the graph G all the words in the given sentence are considered as vertices. Any 2 vertices are connected, if there is any relation between them governed by the relation set R .

4.4 Dependency Extraction

We have the set of all features F and a graph G . Let $f_i \in F$ be the target feature. For example in Section 4.1, *ipod* or *software* can be the *target* feature i.e. the feature with respect to which we want to evaluate the sentiment of the sentence.

Let there be ' n ' features where n is the dimension of F . The algorithm for extracting the set of words $w_i \in S$, that express any opinion about the target feature f_i proceeds as follows:

- i. Initialize n clusters $C_i \forall i = 1..n$
- ii. Make each $f_i \in F$ the clusterhead of C_i . The target feature f_t is the clusterhead of C_t . Initially, each cluster consists only of the clusterhead.
- iii. Assign each word $w_j \in S$ to cluster C_k s.t.

$$k = \arg \min_{i \in n} \text{dist}(w_j, f_i),$$
 Where $\text{dist}(w_j, f_i)$ gives the number of edges, in the shortest path, connecting w_j and f_i in G .
- iv. Merge any cluster C_j with C_i if $\text{dist}(f_j, f_i) < \theta$, Where θ is some threshold distance.
- v. Finally the set of words $w_i \in C_i$ gives the opinion expression regarding the target feature f_t .

Algorithm 1. Dependency Parsing Based Clustering for Sentiment Analysis

In words, we initialize ' n ' clusters C_i , corresponding to each feature $f_i \in F$ s.t. f_i is the clusterhead of C_i . We assign each word $w_i \in S$ to the cluster whose clusterhead is closest to it. The distance is measured in terms of the number of edges in the shortest path, connecting any word and a clusterhead. Any 2 clusters are merged if the distance between their clusterheads is less than some threshold. Finally, the set of words in the cluster C_i , corresponding to the target feature f_i gives the opinion about f_i .

Reviews often have opinions about any specific feature that is closely tied with their opinions about some other feature. Consider the review “*I like Nokia a bit more than Samsung*”. Here, the opinion regarding Nokia is positive but that regarding Samsung is *not negative*. Thus, if we evaluate the polarity of this sentence with respect to Samsung, the opinion about Nokia has to be factored in i.e. they are not independent. This is the reason for merging the opinion expressions of 2 features if they are closely associated.

4.5 Feature Specific Opinion Extraction with Example

Consider the following review given in Section 4.1.1, “*I have an ipod and it is a great buy but I’m probably the only person that dislikes the itunes software*”.

As shown there, $F=\{ipod, buy, person\}$ forms our initial feature set (represented by rectangles in figure 3). The target feature is $f_t = ipod$. The graph consists of all the words in the sentence as vertices. All the words are connected by relations defined by the master relation set R (shown by thin edges in figure 3). The target cluster C_t has the clusterhead f_t . $\{I, have, it\}$ are closest to $ipod$ and are assigned to the corresponding cluster whereas $\{great, probably, but, im\}$ are closest to buy and assigned to its corresponding cluster (the assignment is shown by bold arrows in figure 3). Now, $ipod$ and buy are related through it . The intercluster-distance between them is 2 which is less than $\theta=3$ and thus the 2 clusters are merged. So, buy with all its members is assigned to the target cluster C_t . $\{an, is, a\}$ are ignored as StopWords.

Finally C_t comprises of $\{I, have, ipod, it, great, buy, probably, but, im\}$ which represents the opinion expression about the target feature $f_t= ipod$.

5 Classification of Extracted Features

Now, have the set of opinion words $w_i \in C_t$, that describes the target feature f_t .

Rule Based Classification

We use a sentiment lexicon to find the polarity of each word $w_i \in C_t$. If the number of words tagged positive is greater than that tagged negative, we conclude the sentiment regarding the target feature f_t , to be positive or else negative.

Supervised Classification

Each sentence in the review is represented as a vector consisting of the target feature f_t and its associated opinion words $w_i \in C_t$. These set of vectors are fed into any supervised classification system like the SVM.

6 Learning Parameters

We have two principal parameters to learn, the *significant relation set* and the *merging threshold*.

a. Significant Relation Set

Dependency Parsing gives more than 40 relations, not all of which are equally significant. In order to obtain the subset of relations, which are most significant, we have to probe the entire relation space of $O(2^{40})$ if we use an exhaustive search, which is infeasible. So, we use an alternative approach to find the most significant relations to suit our purpose. We partition the relation space in 3 parts:

- Relations that *should be included* in R
 These consist of the relations *nsubj*, *nsubjpass*, *dobj*, *amod*, *advmod*, *nn*, *neg*.
- Relations that *should not be included* in R

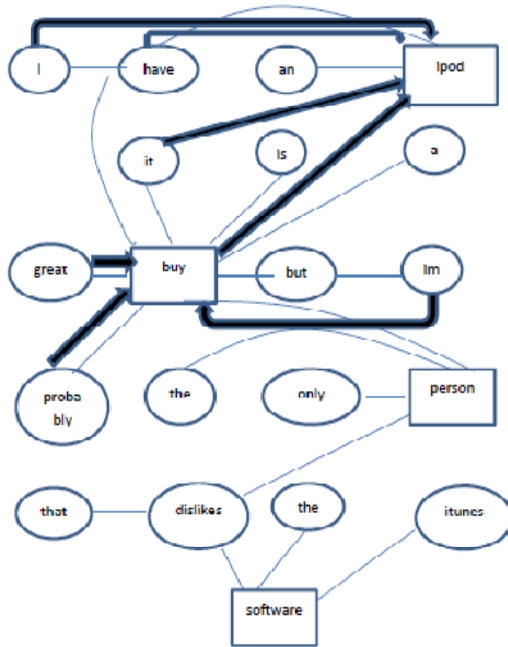


Fig. 1. Dependency parsing based Clustering of Features

These consist of relations irrelevant to our purpose like *numeric modifiers*, *abbreviation relations* etc.

- Relations that *may be included* in R

This partition consists of around 21 relations which may or may not be significant. We now perform leave-one-relation out test or *ablation test* using relations in partition 3. In this, we leave out one relation at a time and compute the overall accuracy of sentiment classification with the remaining relations. Our objective is to find the relations in the 3rd partition that causes *significant* accuracy change. We select an arbitrary domain to perform this test and cross-validate in another domain. We used the labeled data from Hu and Liu *et. al* [5] for learning the parameters.

Table 1. Ablation Test for Significant Relations

Relations	Accuracy (%)
All	63.5
Dep	67.3
Rcmmod	65.4
xcomp, conj_and ccomp, iobj	61.5
advcl, appos, csubj, abbrev, infmod, npavmod, rel, acomp, agent, csubjpass, partmod, pobj, purpcl, xsubj	63.5

In Table 1, we find that leaving out *Dep* and *Rcmmod* causes significant accuracy improvement, over including all the relations. But, we still cannot be sure which among *Dep* and *Rcmmod* plays the spoilsport. So we perform another experiment in a different domain involving only these 2 relations.

Table 2. Ablation Test for Dep and Rcmmod

Relation Set	Accuracy
With Dep+Rcmmod	66
Without Dep	69
Without Rcmmod	67
Without Dep+Rcmmod	68

In Table 2, we find that *Dep* causes the real problem. This is also intuitive when we see the definition of the *Dep* relation in Stanford Dependencies Manual which says “*dependency is labeled as dep when the system is unable to determine a more precise dependency relation between two words*”. Thus it captures many stray relations and introduces noise in the graph. Finally, all the relations in Table 1 (excluding *Dep*) are considered as significant relations.

b. Merging Threshold

Any 2 feature clusters are merged if the inter-cluster distance is less than some threshold distance θ . The distance is measured as the number of edges in the shortest path connecting the 2 cluster-heads. If θ is very small, then any 2 clusters having some long-range dependency will not be merged. Whereas if θ is very large, then all the features will be merged and feature specific dependencies will be lost. We used a small seed set from an arbitrary domain to find the optimal value of θ and cross-validated it across other domains.

Table 3 indicates that $\theta = 3$ will give the optimal result. $\theta = 2$ means all the clusters are disjoint and there is no merging, whereas $\theta = 3$ implies any 2 clusters are merged if there is only one intermediate word linking them.

Table 3. Inter-cluster distance threshold accuracy

θ	Accuracy (%)
2	67.85
3	69.28
4	68.21
5	67.40

7 Experimental Evaluation

We used 2 datasets. Dataset₁ consisted of 500 reviews extracted from the dataset used by Lakkaraju *et. al* [4]. The extracted data came from 3 domains *laptops, camera and printers*.

The second dataset was extracted from the data used by Hu and Liu *et. al* [5]. It consisted of about 2500 reviews from varied domains like *antivirus, camera, dvd, ipod, music player, router, mobile* etc. Each sentence is tagged with a feature and sentiment orientation of the sentence with respect to the feature.

In the original dataset (Hu and Liu, [5]), majority of the sentences consisted of a single feature, and had either entirely positive or entirely negative orientation. From there a new dataset was constructed, by combining each positive sentiment sentence with a negative sentiment sentence using connectives (*but, however, although* etc.), *in the same domain, describing the same entity*. For Example, “The display of the camera is bad” and “It is expensive” were connected by *but*. This forms our Dataset₂.

Table 4. Domain specific accuracy for our rule based system in dataset₂

Domain	Baseline 1 (%)	Baseline 2 (%)	Proposed System (%)
Antivirus	50.00	56.82	63.63
Camera 1	50.00	61.67	78.33
Camera 2	50.00	61.76	70.58
Camera 3	51.67	53.33	60.00
Camera 4	52.38	57.14	78.57
Diaper	50.00	63.63	57.57
DVD	52.21	63.23	66.18
IPOD	50.00	57.69	67.30
Mobile 1	51.16	61.63	66.28
Mobile 2	50.81	65.32	70.96
Music Player 1	50.30	57.62	64.37
Music Player 2	50.00	60.60	67.02
Router 1	50.00	58.33	61.67
Router 2	50.00	59.72	70.83

Now, each sentence in this new dataset has a mixed emotion about various features.

We determined Baseline_1 by counting the number of positive and negative opinion words in the sentence. The final polarity is determined by majority voting. This is a very naïve baseline. So we defined an improved Baseline_2 (Hu and Liu *et. al.*, [5]). If there ‘ n ’ features f_i and ‘ m ’ opinion words O_i , each O_i expresses an opinion about the *closest* feature f_i .

We used the sentiment lexicon used by Hu and Liu *et. al* [5] for rule based classification. Since we have a 2-class classification (positive or negative) problem, any tie is resolved by flipping a coin.

Table 4 gives the domain specific accuracy comparison of our system with Baseline_1 and Baseline_2 . We find that the proposed system performs way better than both the baselines in *every domain*. Table 5 gives the average accuracy of the system and the baselines across all the domains.

Table 5. Overall accuracy for our rule-based system in Dataset₂

System	Accuracy (%)
Baseline_1	50.35
Baseline_2	58.93
Proposed System	70.00

We also performed comparisons with another state-of-the-art system namely, CFACTS developed by Lakkaraju *et. al* [4]. Unlike the CFACTS system, our system has a much less data requirement as it does not train on domain-specific data. Hence the domain-specific feature extraction accuracy of CFACTS is better. Thus we compared only the final sentiment evaluation accuracy of the 2 systems. This is a valid comparison as CFACTS claimed to have 100% topic purity in feature extraction which means its feature extraction accuracy cannot degrade its sentiment evaluation accuracy.

Table 6. Sentiment Classification accuracy comparison for rule-based classification in Dataset₁

System	Sentiment Evaluation Accuracy (%)
Baseline_1	68.75
Baseline_2	61.10
CFACTS-R	80.54
CFACTS	81.28
FACTS-R	72.25
FACTS	75.72
JST	76.18
Proposed System	80.98

The performance comparison between the feature specific module of CFACTS and our system is made under the *assumption that the features should be explicitly present in the review*. This is necessary in our system as the user is providing the feature with respect to which the review has to be analyzed. Consider the review sentence, “*The mobile is too heavy*”. Here the implicit feature is *weight* and the implicit sentiment is

negative. Since the system, we developed, does not use any domain specific data for sentiment classification, such reviews cannot be aptly handled by the system.

From *Table 6*, we find that the proposed system performs at par with all the given systems, *with no data requirement*. This, however, comes at a cost that it cannot capture domain-specific implicit feature or hidden sentiment.

In order to have a flavor of the system performance, when tagged data is available, we performed experimental evaluations in 2 arbitrary domains namely, *camera and mobile* using Dataset₁.

Table 7. Supervised classification accuracy in 2 domains in Dataset₂

Domain	Baseline ₁ (%)	Proposed System (%)
Mobile	51.42 (50.72/99.29)	83.82 (83.82/83.82)
Camera	50	86.99 (84.73/90.24)

The supervised system uses Support Vector Machines for classification of feature vectors. *Table 7* shows the huge leap in accuracy from the naïve baseline. The difference in accuracy between the rule-based system and the supervised classification system stems from the fact, that the system can now capture both domain specific sentiment and implicit features. But this comes at a cost of enhanced tagged data requirement for every domain and the system needs to be trained separately for every domain.

8 Conclusions and Future Work

In this paper, we developed a system that extracts potential features from a review and clusters opinion expressions describing each of the features. It finally retrieves the opinion expression describing the user specified feature. The main achievements of the paper can be summarized as:

1. The work exploits associations between the opinion expressions about various features that form a coherent review using dependency parsing.
2. We perform an in-depth analysis of the *dependency relations* deemed significant while mining the relations between the words forming opinion expressions.
3. The system takes into consideration the phenomena where opinion expressions about various features are co-related and thus merges them.
4. The parameters, namely the *significant relation set* and the *merging threshold*, are domain independent. Thus the system has a minimal data requirement as it performs a one-time learning of these parameters.
5. Extensive evaluations were made across various domains over two datasets where the system outperformed the chosen baselines in *all domains*.
6. The system showed improved accuracy not only over the naïve baseline but also over the chosen sophisticated baseline [5].

7. It performed at par with the state-of-the-art systems [4] despite its data limitations, as it does not use any domain specific data for training.
8. We showed that using supervised classification (when tagged data is available for training) the system outperforms the naïve baseline by a huge margin.

The drawback of the system is that it cannot evaluate domain dependent implicit sentiment as it does not train on any domain specific data. Thus the system does not distinguish between “The story is unpredictable” (positive sentiment) and “The steering wheel is unpredictable” (negative sentiment). This is due to the usage of a generic sentiment lexicon, in the final stage, in rule based classification. Supervised classification can distinguish between the two sentiments but it needs tagged data and separate training for every domain.

References

1. Mosha, C.: Combining Dependency Parsing with Shallow Semantic Analysis for Chinese Opinion-Element Relation Identification, pp. 299–305. IEEE (2010)
2. Wu, Y., Zhang, Q., Huang, X., Wu, L.: Phrase Dependency Parsing for Opinion Mining. In: EMNLP 2009, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3 (2009)
3. Zhang, Q., Wu, Y., Li, T., Ogihara, M., Johnson, J., Huang, X.: Mining Product Reviews Based on Shallow Dependency Parsing. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009 (2009)
4. Lakkaraju, H., Bhattacharyya, C., Bhattacharya, I., Merugu, S.: Exploiting Coherence for the simultaneous discovery of latent facets and associated sentiments. In: SIAM International Conference on Data Mining (SDM) (April 2011)
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD 2004: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
6. Griffiths, T.L., Steyvers, M., Blei, D.M., Tenenbaum, J.B.: Integrating Topics and Syntax. In: Advances in Neural Information Processing Systems, vol. 17 (2005)
7. Blei, D.M., Jordan, M., Ng, A.: Latent Dirichlet Allocation. *Journal of Machine Learning and Research*, 993–1022 (2003)

Biographies or Blenders: Which Resource Is Best for Cross-Domain Sentiment Analysis?

Natalia Ponomareva and Mike Thelwall

University of Wolverhampton, UK
Statistical Cybermetrics Research group
{nata.ponomareva,m.thelwall}@wlv.ac.uk

Abstract. Domain adaptation is usually discussed from the point of view of new algorithms that minimise performance loss when applying a classifier trained on one domain to another. However, finding pertinent data similar to the test domain is equally important for achieving high accuracy in a cross-domain task. This study proposes an algorithm for automatic estimation of performance loss in the context of cross-domain sentiment classification. We present and validate several measures of domain similarity specially designed for the sentiment classification task. We also introduce a new characteristic, called domain complexity, as another independent factor influencing performance loss, and propose various functions for its approximation. Finally, a linear regression for modeling accuracy loss is built and tested in different evaluation settings. As a result, we are able to predict the accuracy loss with an average error of 1.5% and a maximum error of 3.4%.

1 Introduction

Lack of annotated corpora that would suit the needs of NLP researchers is a common problem for many NLP tasks where machine learning is involved. More specifically, whilst there is a plethora of available annotated resources on the Internet, in many cases these resources do not match the data to be classified. Most of studies on domain adaptation research how to decrease the loss of performance when adapting a classifier trained on a domain with available annotated data to the target domain [6], [5], [7]. However, the choice of pertinent data seems to be equally important for obtaining satisfactory results. Indeed, machine-learning techniques are based on the assumption that training and test data are driven from the same probability distribution, and, therefore, they perform much better when training and test data sets are alike. Thus, the task of finding the best training data transforms into the task of finding data whose feature distribution is similar to the test one.

The present paper considers this issue in the context of sentiment classification (SC). A drastic drop in performance when training and test data are different is common for cross-domain SC algorithms [12]. Usually this problem is tackled by proposing another domain-adaptation method, like ensembles of classifiers [2] or graph-based approaches [13]. However, these algorithms normally do not work

well for very different source and target domains. Some studies also prove the benefit of using only the closest data set for training instead of exploiting all available data. For example, combinations of classifiers from different domains in some cases perform much worse than a single classifier trained on the closest domain [4]. These facts confirm the necessity to find a technique for choosing the most appropriate training data out of various available annotated data sets. The main goal of the present research is to analyse the principal factors causing performance loss and construct a model to predict the accuracy drop for a given pair of training and test data sets.

Due to the specificity of SC, the most discriminative features used in machine learning are not necessary the most frequent words but words bearing sentiment. Numerous studies in Sentiment Analysis pointed out that adjectives, adverbs and verbs are usually good indicators of sentiment [10]. Thus, reduction of the feature set to unigrams and bigrams containing these parts-of-speech (POS) may give a better approximation to the real feature distribution. After building a feature representation of the domain we need to establish a similarity metric to measure closeness between source and target domain distributions. We analyse and compare different similarity functions, e.g. geometrically motivated measures and metrics borrowed from Information Theory and Corpus Linguistics (as the notion of domain similarity is identical to the notion of corpora comparability). Similarity in the sense of the proposed measures is controlled by the most frequent features, as they make the major impact on the value of the functions. At the same time, the tail of the distribution is also important because it indicates the complexity of the problem: longer tails correspond to richer domains which tend to be more complex for machine-learning tasks. We demonstrate this property in Section 3 by comparing in-domain accuracy for different corpora. Our experiments show that more diverse domains like books and DVDs give lower accuracy than kitchen appliances or electronics. In the light of this, we introduce another measure, called domain complexity, that is determined by the tail of the distribution and reflects the difficulty of classification task for a given data set. In Section 4.2 several functions for approximating domain complexity are suggested, and their high correlation with in-domain accuracy is evidenced.

In the final step of our work we model cross-domain performance loss on the basis of two characteristics: domain similarity and complexity variance between source and target domains. We assume linear dependency between model output and its parameters and use multiple linear regression framework to compute model coefficients.

The paper is structured as follows. In Section 2 some related studies are overviewed. In Section 3 we describe our data and give preliminary results on domain adaptation. Section 4 introduces and validates measures of domain similarity and complexity, which are used in accuracy loss modeling described and evaluated in Sections 5-6. Finally, we sum up our contributions and give directions for the future research in Section 7.

2 Related Work

The majority of works on cross-domain SC aim to elaborate a good transfer algorithm which minimises the performance loss observed on a target domain. The main approaches there include ensembles of classifiers [2], Structural Correspondence Learning [4], graph algorithms [13] and Spectral Feature Alignment [9]. However, there is very little research on evaluation of corpus quality for being a good training data for a given test data. In the paper of [4] the necessity of a domain similarity measure was mentioned for the first time. The authors proposed to apply \mathcal{A} -measure [3] and computed its proxy in a supervised manner. At the same time, a great interest to this problem has been expressed recently with regard to other NLP tasks, e.g. dependency parsing and POS tagging. During the last two years several works claiming the importance of domain similarity have been published [1], [11]. The number of recent parallel studies prove the timeliness of the present research.

The objective of the work described in [1] is similar to ours; the authors aimed to estimate the performance loss of a POS tagger across domains. They experimented with several POS tagger algorithms and different domains from BNC, provided by BNC annotators. Their conclusions are quite optimistic as they were able to predict the accuracy loss with 95% of confidence. However, we see several problems with the results reported in the paper. First of all, the best similarity measure was chosen on the basis of its correlation with the accuracy instead of the accuracy drop. In our opinion, this is not absolutely correct, because it implies the same accuracy for any pair of identical domains, which is not true as in-domain accuracy is determined by the corpus properties. Second, the authors scrutinised quite similar data, as the average cross-domain accuracy lies between 93-95% which is very close to the state-of-the-art accuracy for POS tagging. Therefore, their results can not be presumed to be representative and be generalised for the case of rather different domains.

The study of [11] approached the cross-domain problem from a different angle – instead of choosing the best corpus out of the set of existing ones, the authors disregarded domain boundaries and proposed to acquire a corpus of separate documents similar to the test data. One of the interesting ideas suggested by this research is the use of topic models for document representation. The authors compared topic models representation with word representation and concluded the slight advantage of the former for the dependency parser problem, although the difference between them is significant only for one test data set out of the three considered in the paper.

3 Preliminary Results

Our data consist of Amazon product reviews on 7 different topics: books (BO), electronics (EL), kitchen&housewares (KI), DVDs (DV), music (MU), health&personal care (HE) and toys&games(TO) [4]. Reviews were rated using binary scale, 1-2 stars reviews are considered as negative and 4-5 stars reviews

Table 1. Reviews corpora statistics

corpus	num words	mean words	vocab size	vocab size ($freq \geq 3$)	% of rare words
BO	364k	181.8	23k	8,256	64.77
DV	397k	198.7	24k	8,632	64.16
MU	300k	150.1	19k	6,163	67.16
EL	236k	117.9	12k	4,465	61.71
KI	198k	98.9	11k	4,053	61.49
TO	206k	102.9	11k	4,018	63.37
HE	188k	93.9	11k	4,022	61.83

- as positive. The data within each domain are balanced, they contain 1000 positive and 1000 negative reviews.

In Table 1 we can see that BO, DV and MU have the longest reviews, which also implies that the size of their dictionaries is larger than that of other domains. These observations confirm our intuition that vocabularies of BO and DV are more diverse and sophisticated.

First, we accomplish in-domain experiments which represent the top boundary that any cross-domain algorithm is aiming to reach (Fig. 1). We test different feature sets and conclude that unigrams together with bigrams of stems weighted with binary values yield the best performance for all domains under consideration. Fig. 1 reveals that in-domain accuracies are higher for corpora with a smaller vocabulary and a lower percentage of rare words (words that appear less than 3 times in total). One of the logical explanations of this fact is that vocabulary diversity and long tail of feature distribution make automatic learning, in general, and SC, in particular, more difficult.

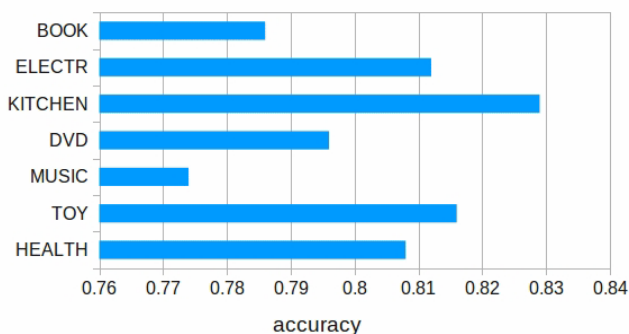
**Fig. 1.** In-domain accuracies

Fig. 2 gives cross-domain accuracies for all domain pairs. Products on the y-axis are used as training and products on the x-axis - as test data. It can be seen

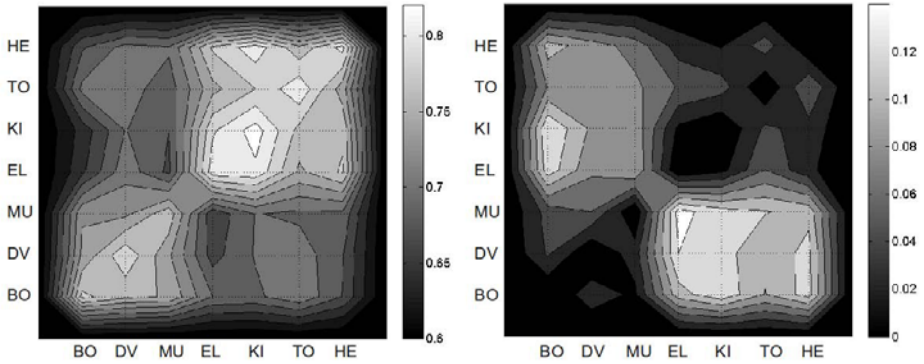


Fig. 2. Cross-domain results: a) accuracy, b) accuracy drop

that MU, DV and BO are relatively close and the same is true for HE, EL and KI. The adaptation of the classifier inside these domain clusters costs less than 5% in terms of accuracy drop for almost all domain pairs. At the same time, the drop of performance jumps sharply up to 14-15% for completely different domain pairs like (DV, HE) or (BO, EL). Interestingly, the performance loss is non-symmetric and it is normally higher when the source domain is more diverse. Non-symmetry of the drop is especially evident for a pair of very different domains.

4 Domain Similarity and Complexity

4.1 Domain Similarity

A domain is represented by a corpus of documents, therefore, domain similarity is identical to corpus similarity. The easiest way to find an appropriate measure of corpus similarity that would suit our needs is to adopt existing measures widely used in Corpus Linguistics (CL) for measuring the comparability of corpora pairs. However, there are certain differences between our objectives and CL ones:

1. We are not interested in all terms of a corpus but rather on those which bear sentiment and can be considered as discriminative features for a machine learning algorithm. The study on SA suggested that adjectives, verbs and adverbs suit our purposes the best, therefore, we keep only unigrams and bigrams that contain those POS as features to compute corpus similarity.

2. CL compares frequencies of terms and, thus, the terms with the highest frequency bring the most substantial impact into similarity. However, it is not obvious that the most frequent terms are the most valuable for SC. Therefore, together with term frequencies we adopt and compare TF-IDF and IDF term weighting schemes.

Kilgariff [8] introduced and tested 3 measures of corpus similarity - χ^2 , Spearman rank correlation coefficient and cross-entropy, and in their experiments χ^2 showed the best correlation with the gold standard. We adopt χ^2 together with

Table 2. Correlation for different domain similarity measures

measure	R (freq)	R (filtr., freq)	R (filtr., TFIDF)	R (filtr., IDF)
<i>cosine</i>	-0.790	-0.840	-0.836	-0.863
<i>Jaccard</i>	-0.869	-0.879	-0.879	-0.879
χ^2	0.855	0.869	0.876	0.879
D_{KL}	0.734	0.827	0.676	0.796
D_{JS}	0.829	0.833	0.804	0.876

some measures borrowed from Information theory – Kullback-Leibler divergence (D_{KL}) and its symmetric analogue Jensen-Shannon divergence (D_{JS}), and other well-known similarity functions including the Jaccard coefficient (*Jaccard*) and cosine similarity (*cosine*). Note that some proposed functions measure actual similarity like cosine or Jaccard, while the others (χ^2 , D_{KL} and D_{JS}) measure distance. This difference does not really matter much as similarity can always be obtained by inverting the distance.

Table 2 gives correlations between the accuracy drop and similarity functions applied to different feature representations of corpora. There “freq”, “TFIDF” and “IDF” state for a term-weighting scheme and “filtr” means that the features were filtered by appearance of adjectives, adverbs and verbs. The different sign of the correlations is due to the fact that domain distance is directly-proportional and domain similarity is inversely-proportional to the accuracy loss.

We can observe that correlation is higher for filtered features. A small increase in correlation can be identified for IDF weights with respect to frequencies and TFIDF. As far as similarity functions are concerned, strangely, the simplest function which does not depend on feature weights, Jaccard, shows the best overall correlation. χ^2 with IDF weights gives the same results and we select it for the accuracy loss modeling as it was previously chosen to be the most adequate measure of domain similarity.

Clusters of similar domains are clearly seen in Fig. 3 where pairwise similarities of our corpora according to $\chi^2_{inv} = \frac{1}{\chi^2}$ measure are depicted. Analysing this figure we can say that the boundary between similar and distinct domains approximately corresponds to $\chi^2_{inv} = 1.7$.

4.2 Domain Complexity

Similarity between domains is mostly controlled by frequent words, but the shape of the corpus distribution is also influenced by rare words representing its tail. In Section 3 we showed that richer domains with more rare words are more complex for SC (Fig. 1). We can also observe that the accuracy loss is higher in cross-domain settings when the source domain is more complex than the target one (Fig. 2). These facts suggest that domain complexity in the sense of vocabulary diversity is another important characteristic regulating the performance loss.

We propose several measures to approximate domain complexity: *percentage of rare words*, *word richness*, calculated as a proportion of vocabulary size in a corpus size, and *relative entropy*, which is a percentage of corpus entropy out

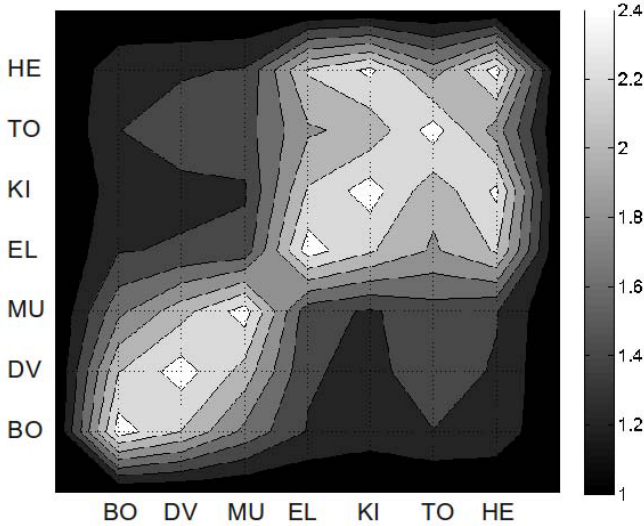


Fig. 3. Pairwise similarity of corpora according to χ_{inv}^2

of the maximum entropy when all vocabulary words are distributed uniformly. Table 3 clearly divides all corpora into 2 groups: more complex domains like BO, DV and MU and simpler domains like EL, KI, TO and HE. More complex domains have higher number of rare words, a higher level of word richness and lower relative entropy. The last phenomena can be explained that distributions of more simple domains are closer to uniform than distributions of more complex domains.

Table 4, where correlations between complexity measures and in-domain accuracy are presented, shows that the number of rare words approximates domain complexity the best. Interestingly, relative entropy, which is determined by the shape of the distribution and not only its tail, gives the lowest correlation with the in-domain accuracy. That means that domain complexity according to our definition is mostly associated with noisy low frequency terms.

Complexity is a property of one domain, therefore, for cross-domain settings we need to consider complexity variance between source and target domains. Let us denote Δc a measure of complexity variance which is positive for more complex and negative for more simple target domains:

$$\Delta c = c_t - c_s, \quad (1)$$

where, c_s and c_t are relative entropy of source and target domains respectively.

5 Modeling the Accuracy Loss

In the previous section we presented two domain characteristics: domain similarity and complexity variance, that were proved to have an impact on

Table 3. Corpora complexity

corpus	accuracy	% of rare words	word richness	rel.entropy, %
BO	0.786	64.77	0.064	9.23
DV	0.796	64.16	0.061	8.02
MU	0.774	67.16	0.063	8.98
EL	0.812	61.71	0.049	12.66
KI	0.829	61.49	0.053	14.44
TO	0.816	63.37	0.053	15.27
HE	0.808	61.83	0.056	15.82

Table 4. Pearson correlation with indomain accuracy for complexity measures

% of rare words	word richness	rel.entropy
0.904	0.846	-0.793

the accuracy loss. These measures are independent, their values do not imply each other. In particular, if domains are different, they can still be of the same complexity level, or either of them can be more complex than another one. For example, on one hand, TO seems to be much more complex than EL, KI and HE (Table 3), but, on the other hand, it is quite similar to these domains (Fig 3).

To model the performance drop we assume its linear dependency on domain similarity and complexity variance and propose the following linear regression model F :

$$F(s_{ij}, \Delta c_{ij}) = \beta_0 + \beta_1 s_{ij} + \beta_2 \Delta c_{ij}, \quad (2)$$

where s_{ij} stands for domain similarity (or distance) between domains i and j and Δc_{ij} - for the difference between domain complexities. The unknown coefficients β_i are solutions of the following system of linear equations:

$$\beta_0 + \beta_1 s_{ij} + \beta_2 \Delta c_{ij} = \Delta a_{ij}, \quad (3)$$

where Δa_{ij} is the accuracy drop when adapting the classifier from domain i to domain j .

Apart from establishing the unknown coefficients, it is necessary to accomplish a thorough analysis of model goodness. In particular, it is important to analyse whether the correlation between our predictors and the response variable is statistically significant and whether each of the predictors give a significant impact to the model. Another part of the evaluation consists of estimating the average accuracy and prediction of model behaviour on unseen examples.

6 Evaluation

Our data comprise 7 domains, which gives us a sample of 42 pairs to estimate unknown parameters and validate the model. We intended to artificially increase the number of examples by random division of each domain in 2 parts, but

smaller data sets did not give reliable estimations for domain similarity and complexity. Therefore, we decided to carry out the experiments with the initial data. In previous section we established that χ^2 and the percentage of rare words are the best measures for domain similarity and complexity respectively. These functions were used in our accuracy loss modeling.

The evaluation of the constructed regression model includes following steps:

- *Global test (or F-test)* to verify statistical significance of regression model with respect to all its predictors.
- *Test on individual variables (or t-test)* to reveal regressors that do not bring a significant impact into the model.
- *Leave-one-out-cross validation* for the data set of 42 examples.
- *Domain-out validation* to estimate the accuracy drop obtained on domains which do not participate in modeling. This can give us a more reliable approximation to a real performance loss observed on unseen data.

6.1 Model Validation

The null hypothesis for global test states that there is no correlation between regressors and the response variable. Our purpose is to demonstrate that this hypothesis must be rejected with a high level of confidence. In other words, we have to show that coefficient of determination R^2 is high enough to consider its value significantly different from zero.

Table 5. F-test for the linear regression model

R^2	R	F-value	p-value
0.873	0.935	134.60	$\ll 0.0001$

Table 5 proves statistical significance of our model with the confidence level more than 99.9%, the correlation between the response and predictors is over 0.85. However, the global test evaluates the model in general, affirming that one of the coefficients is different from zero, but it does not say anything about each of the model parameters in particular.

Table 6. t-test for regression coefficients

	β_0	β_1	β_2
value	-8.67	27.71	-0.55
standard error	1.08	1.77	0.11
t-value	-8.00	15.67	-4.86
p-value	$\ll 0.0001$	$\ll 0.0001$	$\ll 0.0001$

Table 6 presents the results of the test on individual coefficients. All of them are justified to be statistically significant with the confidence level higher than

Table 7. Leave-one-out cross-validation results

accuracy drop	standard error	standard deviation	max error, 95%
all data	1.566	1.091	3.404
< 5%	1.465	1.133	3.373
>5%, < 10%	1.646	1.173	3.622
> 10%	1.556	1.166	3.519

Table 8. Domain-out validation results

Domain	correlation coeff.	standard error	standard deviation	max error, 95%
BO	0.882	2.0856	1.400	4.600
DV	0.943	1.253	1.308	3.602
MU	0.898	1.789	1.869	5.145
EL	0.946	1.861	1.156	3.937
KI	0.970	1.216	0.750	2.563
TO	0.932	1.148	1.381	3.628
HE	0.974	1.111	1.160	3.195

99.9%. The model conforms with an intuitive assumption that high similarity leads to a small accuracy drop and vice versa. It also proves that relatively simpler source domains tend to give a smaller accuracy drop on relatively more complex target domains.

6.2 Accuracy Estimation

The next step after validating the model is an estimation of the error it gives for unseen data. We evaluate the accuracy for 2 different settings: leave-one-out cross-validation and domain-out validation, which means that we take each domain out of the consideration when learning the model and use it only for testing. The latter procedure seems to give more reliable estimate for the accuracy on unseen data.

Results of leave-one-out cross-validation are presented in Table 7. The average error is around 1.5% and it does not exceed a threshold of 3.4% with the confidence level of 95%. Table 7 gives detailed results for different accuracy drops. As it can be seen, errors are very similar regardless of how close the domains are, but lower values are being noticed for more similar domains (with accuracy drops less than 5%). This is a strength of the model as our main purpose is to identify the closest domains.

The results of domain-out validation reveal 3 domains (BO, MU and EL) with a very high standard error close to 2% (Table 8). It suggests that these domains have a higher level of noise and the size of the data is not large enough for reliable estimation of domain similarity and complexity. Concerning the rest of the domains, the constructed model gives much more accurate prediction of the performance loss with the standard error varying from 1.1% to 1.25%.

This is much lower than the average error of leave-one-out cross-validation. The maximum error for the confidence level equal to 95% is high for all domains because of the small data sample containing only 12 examples. To obtain a more precise confidence interval for the error rate, more domain pairs are needed.

7 Conclusions and Future Work

The paper gives an alternative insights into the domain adaptation problem. Instead of proposing a new efficient technique of domain transfer we present the method of modeling the performance loss on different data sets. This technique can help to choose the most appropriate data similar to the existing test one. First, we introduce measures of domain similarity and complexity and demonstrate their influence into the performance loss of a cross-domain classifier. Then, using these measures, we construct a linear regression model, which we verify and test in different evaluation settings. Our model demonstrates a satisfactory behaviour, predicting performance loss with an error of 1.5%. The standard error and deviation of the predictions are slightly lower for small accuracy drops that proves the ability of the model to identify similar domains.

In future, we plan to carry out experiments on larger data sets, verifying our results on new topics and genres. In particular, challenging data gathered from Twitter, MySpace and Youtube will be examined and the possibility to use them in the cross-domain task will be studied. We believe that these experiments will prove the importance of the domain complexity measure because corpus complexity differs even more when considering distinct genres than just different topics.

Another direction of the future work will focus on improvement of proposed measures of domain similarity and complexity. The work of [11] shows that for some data sets topic models are more efficient for approximation of domain similarity than word representations. However, the authors dealt with dependency parsing and their conclusions may not be valid for SC. Therefore, we intend to compare topic models with our unigrams+bigrams corpus representations and analyse their influence into the overall accuracy of our prediction model.

Acknowledgements. This work was supported by a European Union grant by the 7th Framework Programme, Theme 3: Science of complex systems for socially intelligent ICT. It is part of the CyberEmotions project (contract 231323).

References

1. Asch, V.V., Daelemans, W.: Using domain similarity for performance estimation. In: Proceedings of the 2010 Workshop on Domain Adaptation for NLP, ACL 2010, pp. 31–36 (2010)
2. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: A case study. In: Proceedings of RANLP 2005 (2005)

3. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: *Advances in Neural Information Processing Systems*, NIPS (2006)
4. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: *Proceedings of ACL 2007*, pp. 440–447 (2007)
5. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: *Proceedings of EMNLP 2006*, pp. 120–128 (2006)
6. Daume III, H., Marcu, D.: Domain adaptation for statistical classifiers. *Artificial Intelligence Research* 26, 101–126 (2006)
7. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: *Proceedings of ICML 2011* (2011)
8. Kilgarriff, A.: Comparing corpora. *International Journal of Corpus Linguistics* 6(1), 97–133 (2001)
9. Pan, S.J., Niz, X., Sunz, J.T., Yangy, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: *Proceedings of WWW 2010* (2010)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
11. Plank, B., van Noord, G.: Effective measures of domain similarity for parsing. In: *Proceedings of ACL 2011*, pp. 1566–1576 (2011)
12. Read, J.: Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: *Proceedings of the ACL Student Research Workshop*, pp. 43–48 (2005)
13. Wu, Q., Tan, S., Cheng, X.: Graph ranking for sentiment transfer. In: *Proceedings of ACL-IJCNLP 2009*, pp. 317–320 (2009)

A Generate-and-Test Method of Detecting Negative-Sentiment Sentences

Yoonjung Choi¹, Hyo-Jung Oh², and Sung-Hyon Myaeng¹

¹ Department of Computer Science, KAIST, Daejeon, South Korea

² ETRI, Daejeon, South Korea

ohj@etri.re.kr, {choiyj35,myaeng}@kaist.ac.kr

Abstract. Sentiment analysis requires human efforts to construct clue lexicons and/or annotations for machine learning, which are considered domain-dependent. This paper presents a sentiment analysis method where clues are learned automatically with a minimum training data at a sentence level. The main strategy is to learn and weight sentiment-revealing clues by first generating a maximal set of candidates from the annotated sentences for maximum recall and learning a classifier using linguistically-motivated composite features at a later stage for higher precision. The proposed method is geared toward detecting negative sentiment sentences as they are not appropriate for suggesting contextual ads. We show how clue-based sentiment analysis can be done without having to assume availability of a separately constructed clue lexicon. Our experimental work with both Korean and English news corpora shows that the proposed method outperforms word-feature based SVM classifiers. The result is especially encouraging because this relatively simple method can be used for documents in new domains and time periods for which sentiment clues may vary.

Keywords: Sentiment Analysis, Opinion Analysis, Contextual Advertising.

1 Introduction

Sentiment Analysis (SA) that usually determines positive or negative polarity values of text in online news, blogs, reviews, online communities etc. have drawn attention in many areas such as product or service reviews, election analyses, and issue detection [3,15,16]. Polarity decisions, often modeled as a classification problem, are affected by sentiment-revealing clues in sentences. As positive or negative opinions and feelings are sometimes expressed differently in different domains [2,13,14], however, sentiment clues are known to be domain-dependent. Therefore, sentiment classifiers need to be trained for new domains, or a clue lexicon needs to be constructed for or adapted to a target domain. When features are generated automatically for a machine learning approach, annotated examples are required.

Needless to say, the same situation arises for different languages. While several sentiment-related resources have been developed manually or semi-automatically for English, such as SentiWordNet [17] and General Inquirer [16], it is difficult to find

such resources in other languages except for the recent development in NTCIR [18]. Even if some resources exist for non-English, such as Chinese and Japanese, their sizes are usually limited because of the human efforts required for building them.

An alternative is to rely on machine learning approaches where sentiment clues or features are learned from polarity-annotated data, typically individual sentences, generated manually. It has been found that generating features is inferior to hybrid approaches where a clue lexicon is utilized for feature weighting [15]. In order to address the domain-dependency and labor intensiveness issues, some attempts have been made to learn new clues automatically for a new domain using existing ones in another domain [10]. The lack of resources (such as WordNet and thesauri) in different languages and domains obviously hamper not only research but also development of operational systems.

In order to achieve an effect of using a clue lexicon, we devised a method of generating a maximal set of clue candidates with a minimal training set of polarity-annotated sentences. Using this set of features for SA allows for a maximum recall but with low precision. In order to increase precision, we construct a classifier that employs several different types of features that are linguistically motivated, genre-specific, and language-specific. No additional manual work is required to generate these features.

We applied this “generate-and-test” method to online news articles for the purpose of selecting negative sentences. This functionality is critical for a contextual ad matching system that attaches advertisements to relevant news articles. If an article talks about a company or product with a negative sentiment, for example, the advertisement relevant to it should be suppressed. While it is not a big problem that the irrelevant advertisement is attached to the positive article, attaching an ad to an article that provides negative sentiment to the related entity would be detrimental. As such, SA in our work is equivalent to binary classification of a sentence between negative and non-negative (i.e. both positive and neutral) categories.

2 Related Work

There are two types of prior work in sentence-level SA: lexicon-based approaches and feature-based approaches using a machine learning method.

Lexicon-based methods attempt to collect clues using synonym and antonym relations in a dictionary like WordNet [7] based on a set of seed opinion words [5]. Kim and Hovy [9] propose a bootstrapping approach that uses a small set of seed opinion words to find their synonyms and antonyms in WordNet and predict the semantic orientation of adjectives. However, these methods rely on manually generated resources which vary in different domains.

Pang et al. [15] successfully applied a machine learning approach to classifying sentiment for movie reviews. Wilson et al. [12] also formulated sentiment detection as a supervised learning task. Instead of using word-based classification, however, they focus on the construction of linguistic features and train classifiers using Boostexter. Durant and Smith [4] applied multiple text categorization models, Naïve Bayes and

SVMs, to classification of political blog posts. Machine learning approaches have to rely on human efforts in annotation, which often determine their performances.

Recently, Melville et al. [11] present a machine learning approach that overcomes the problem of constructing training data or lexicons by effectively combining background lexical knowledge such as a lexicon with supervised learning. They construct a model based on a lexicon and another trained on labeled data. Then a composite multinomial Naïve Bayes classifier is created to capture both models. Fan and Chang [6] utilize hybrid sentiment classification to contextual advertising. They apply a dictionary from [5] to assign each word a weight (strength) in either a positive, negative or objective direction.

The aforementioned methods are based on flat bag-of-features representation, and do not consider syntactic structures which seem essential to infer the polarity of a whole sentence. Subjective sentences often contain words which reverse the sentiment polarities of other words. Advanced methods have been proposed to utilize composition of sentences [1,8], but they use rules to handle polarity reversal.

3 The Generate-and-Test Approach

The overall strategy is to follow the “generate and test paradigm”: we first generate a maximal set of negative clue candidates to identify potentially negative sentences, and then test whether each sentence is truly negative by means of a classifier at the second step. In other words, the set of clue candidates is generated in such a way that all possible negative sentences are identified for maximum recall. Some non-negative sentences are then filtered out as a way of enhancing precision by training a binary classifier with negative and non-negative categories.

As in Fig. 1, the annotated data are used at the training stage to generate a maximal set of candidate clues and learn a classifier. For clue candidate generation, we take a simplistic approach: all the words in the negative sentences in the training set are assumed to be candidates, without any pre-constructed lexicon. For the classifier, we devised new feature types because mere inclusion of candidate clues or even a fine-tuned clue set, which is difficult to construct to begin with, does not guarantee that the sentence carries negative sentiment. The main novelty of our approach lies in the idea of using very general, almost domain-independent clues extracted from a small training data set for the “generation” step (i.e. as a recall device) and train a classifier with linguistically-motivated, genre-specific features of the same training data set for the “test” step (i.e. as a precision device) where the maximally generated sentiment sentences are classified into negative and non-negative.

When a new input sentence is entered at the SA stage, it is checked to see if it contains at least one negative clue. The goal is not to lose any negative sentences. If it passes the test, it is sent to the classifier for final appraisal, where many sentences are filtered out at this step for higher precision. Further details are found in Sections 3.1 and 3.2.

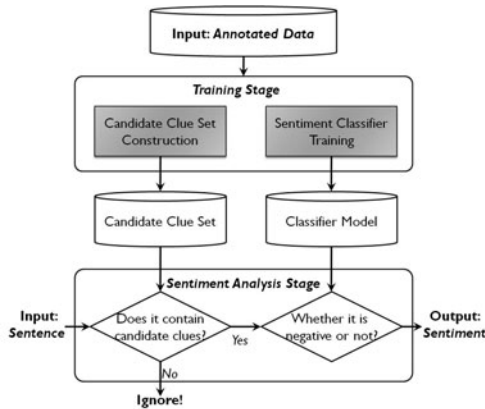


Fig. 1. Overview of the two-stage method using the “Generate and Test” paradigm

3.1 Candidate Clue Set Construction: “Generation” Step

As our goal is to contain all negative sentences at the SA stage, we attempt to select words that have a potential to become true negative sentiment clues, which are words or phrases that cause positive or negative sentiment [2], in the given domain. As such, the candidate clues are generated from the training data where negative and positive sentences are tagged. For an individual word appearing in a polarity-tagged sentence, we apply the following relatively simple heuristics:

- The word w should occur in the training data in a sufficient number of times to be a significant clue:

$$n(w) \leq \text{threshold} \quad (1)$$

The threshold is determined empirically; it is set to be as large as possible provided that the resulting clue set identifies most of the negative sentences in the training set. In our experiment, it was set to be 0.1% of the total number of sentences.

- The word w should occur more often in sentiment-containing sentences (negative in this case) than in non-sentiment sentences. This condition attempts to ensure that w has negative sentiment.

$$n_{neg}(w) / n(w) \geq 0.5 \quad (2)$$

where $n(w)$ is the number of occurrences of w in the training data.

- The word w should not appear in both positive and negative sentences:

$$n_{neg}(w) / (n_{pos}(w) + n_{neg}(w)) = 1 \quad (3)$$

where $n_{pos}(w)$ and $n_{neg}(w)$ are the number of occurrences of w in positive and negative sentences, respectively.

The goal of applying the heuristics is to collect a set of clues that detect most potentially negative sentences. Since the process is close to that of feature selection,

where the goal is to optimize the classification result, we compared the method with other well-known feature selection methods as in reported in the experiment section. Note that the use of simple heuristics coincides with our motivation to minimize resources for SA.

3.2 Sentiment Classifier: “Test” Step

Given a clue-containing sentence, the classifier determines if it belongs to the negative or non-negative category. We chose to use an SVM classifier since it shows the best performance in binary classification. Since the first-cut sentences were selected based on the crude set of clues, the classifier at this step must use more sophisticated and carefully designed features for a finer level testing. We consider 5 types of features constructed out of the clues considered at the “generation” step: 2 basic features, 2 syntactic features, and 1 related to a language-specific property.

Basic Features. One of the two basic features is the number of negative sentiment clues in a given sentence. The more clues it contains, the higher probability it has for the negative sentiment class. The other is the clue weights that discriminate among the candidate clues. We employed three types of clue weight calculation methods:

- *Simple frequency ratio*: It is the ratio of the count of negative-sentiment sentences containing the word w to that of all the sentences containing it:

$$weight(w) = \frac{n_{neg}(w)}{n(w)} \quad (4)$$

- *Z-score*: It measures how many standard deviations an observation is above or below the mean. The clue weight can be calculated based on Z-score. Table 1 shows the contingency table, and we assume that the frequency of word w is under a binomial distribution. The probability of the word w can be calculated as:

$$p(w) = \frac{a+b}{n} \quad (5)$$

The expected number of occurrences of w in the negative sentence set and the variance are as follows:

$$E[n(w)] = (a + c) \times p(w) \quad (6)$$

$$Var[n(w)] = (a + c) \times p(w) \times (1 - p(w)) \quad (7)$$

Then Z-score for a clue weight is computed as:

$$weight_z(w) = \frac{a - (a + c) \times p(w)}{\sqrt{(a + c) \times p(w) \times (1 - p(w))}} \quad (8)$$

Table 1. The contingency table for the word w

	Neg	~Neg
word w	A	B
not w	C	D
	$a+c$	$b+d$

- *Chi-square statistic*: It indicates how much different the distribution of observed frequencies and the expected distribution are. Instead of means and variance, it uses frequencies. The chi-square statistic for a clue weight can be estimate as (using the same contingency table):

$$weight_C(w) = \frac{n \times (a \times d - b \times c)^2}{(a+b)(c+d)(a+c)(b+d)} \tag{9}$$

Syntactic Features. The bag of word approach is known to have an inherent limitation for SA [1]. The first syntactic feature is chosen based on our observation that not all occurrences of clue words affect the sentiment of a sentence equally. In “*Although they have failed to win approval from their fellow-members, they are still debating a number of ideas*”, for example, the negative clue “failed” does not play a key role in determining the sentiment of the sentence because it is included in the subordinate clause. Assuming that the clues associated with the main verb and subject are most influential to determination of the sentiment in a sentence, we consider if a clue serves as part of the subject or main predicate. The corresponding syntactic feature is either absent in a sentence or has the weight that is the average of the weights of the clues that appear in the subject or predicate part.

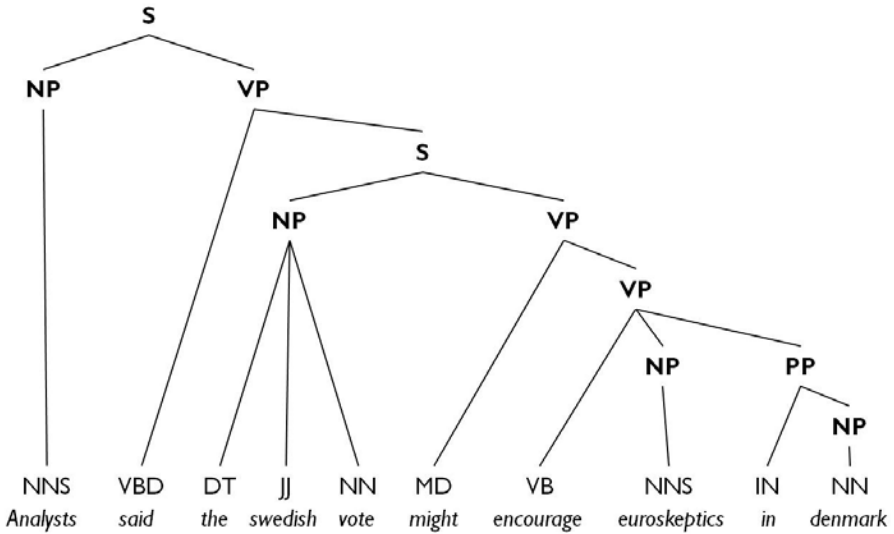


Fig. 2. Parsing result for a sentence containing a subordinate clause with the main predicate

Another syntactic feature has to do with the nature of news articles, the majority of which report on something that happened or have been mentioned by somebody or some organizations. As such, there are many sentences using a quote like “... *said* ...” and “... *reported* ...” as in “*Analysts said the Swedish vote might encourage euroskeptics in Denmark*”, whose parsing result is shown in Fig. 2. The parser tree shows that the main verb is “*said*”, and the main subject is “*analysts*”. Since the main theme appears in the clause “*the Swedish vote might encourage euroskeptics in*

Denmark”, the second syntactic feature has to reflect this phenomenon often found in news articles. Thus the feature is based on whether a clue appears in the nested clause when the primary predicate is one of those used as quotes. This feature overrides the first syntactic feature in that even though a clue appears in a sub-ordinate clause, it is considered a good clue if the main clause contains the quoting predicate.

Language-specific Features. It is obvious that language-specific properties must be considered especially when syntactic features are considered. In Korean, for which we developed a negative sentiment detector, many verbs take the form of noun+“do” or adjective+“be” (or a suffix) as in “*비난하다*” (bi-nan-ha-da) meaning “criticize” in the form of “*비난* (bi-nan, criticism)+ “*하다* (ha-da, do)” and “*지나치다* (ji-na-chi-da) meaning “exceed” in the form of “*지나치* (ji-na-chi, excessive)+“*다* (da, do)”. That is, a noun or an adjective is transformed into a verb with a suffix. The surface level part-of-speech information has to be analyzed further with a morphological analyzer to identify the POS for the component morphemes since an adjective is usually a stronger clue for sentiment analyses than a noun. As a result, POS-tagged words are used as features.

4 Experiments

The main goal of the experiments is to evaluate our SA method without having to use a pre-constructed clue lexicon. Our sub-goals are to understand the amount of data required to build a candidate clue set and to evaluate the classifier using newly devised features introduced to enhance overall effectiveness.

4.1 Experimental Data

Since there is no Korean data set for SA, we constructed a corpus consisting of news articles from commercial web portals over two years (Table 2). The periods of the training and test data sets were made different (one year each) deliberately to avoid content-overlapping and over-fitting for the classifier.

Table 2. Summary of the news corpus

	Training Data	Test Data
Period	Jun. 2008 ~ May. 2009	Jun. 2009 ~ Jun. 2010
# of articles	2,769	2,400
# of sentences	45,343	30,406

To annotate individual sentences for polarity values, we employed undergraduate students majoring in linguistics. Each sentence is tagged by two annotators. The agreement ratio was about 53%. When there was a conflict for a sentence, it was treated as neutral because neither was strong enough; in most conflict cases, one annotator gave a positive or negative value while the other thought it was neutral. The case where one gives positive and the other gives negative is very few (0.0026%).

Table 3 presents the annotation result. While the number of positive sentences is less than that of negative or neutral sentences, mainly due to the nature of news article, the imbalance is not a problem as our focus is on detecting negative sentences.

For our English data, we used the NTCIR-8 MOAT Corpus [18] which contains 5,967 sentences (the number of negative sentences is 737). Half of them from each topic were gathered and used for training and others for testing.

Table 3. Annotation result

	Training	Test Data	Total
positive	2,072 (4.6%)	1,247 (4.1%)	3,319 (4.4%)
negative	6,807 (15.0%)	7,028 (23.1%)	13,835 (18.3%)
neutral	36,464 (80.4%)	22,131 (72.8%)	58,595 (77.3%)
	45,343	30,406	75,749

4.2 Experimental Result – Korean Data

Candidate Clue Set Construction. To extract negative clue candidates, we should determine the threshold in such a way that the recall reaches close to 100%. For experiments, we firstly construct the maximal set of candidate clues from the training data. Table 4 shows how recall values change as the number of clues increases when applying these constructed clues to the test data. The precision values are shown just as a reference. It indicates that only 1,000 clues are sufficient to cover about 97% of negative sentiment sentences. Even if the number increased twice as large (2,185), the gain was negligible.

Table 4. Recall performance changes incurred by different numbers of clues

# of clues	Negative		
	Precision	Recall	F-measure
61	0.3899	0.6148	0.4773
460	0.2617	0.9256	0.4080
1,000	0.2435	0.9747	0.3896
2,185	0.2352	0.9947	0.3804

The recall performance depends on not only the number of clues but also the number of documents from which clues are extracted. Fig. 3 shows that the recall values increase rapidly as the number of clues increases up to about 500 and then they reach to a plateau close to 100%. An interesting observation is that the number of documents does not affect the performance increase when it is more than 300. This indicates that most sentiment clues can be obtained with as small as 300 documents in news articles.

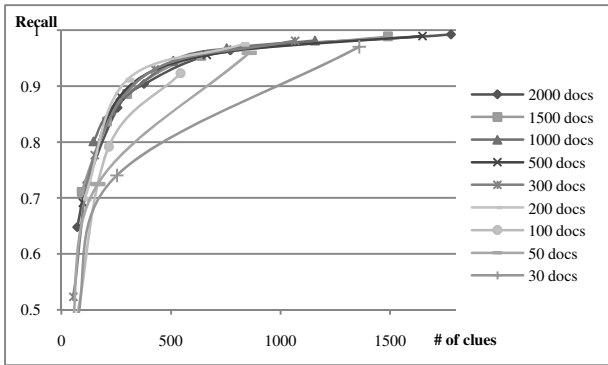


Fig. 3. Recall performance changes for different numbers of clues and docs containing them

While a relatively small number of documents are sufficient for generating clue candidates to cover most negative sentences, the corpus of that size may not be sufficient to calculate appropriate weights for the clues. In order to find the proper number of documents for clue generation in newspapers, we extracted top 500 clues from different sets of documents, varying the numbers from 50 to 2,769 (maximum possible in the corpus). Table 5 shows the results obtained by running an SVM classifier with the basic and syntactic features. It indicates that it is not very helpful to increase the number of documents beyond 1,500, much smaller than the corpus size.

Table 5. Performance changes for different numbers of documents in training

# of clues	Negative		
	Precision	Recall	F-measure
50	0.5474	0.3961	0.4597
100	0.5623	0.4395	0.4934
500	0.6104	0.5530	0.5803
1,000	0.6202	0.5908	0.6051
1,500	0.6217	0.6639	0.6421
2,769	0.6253	0.6985	0.6599

Table 6. Overall performance of the proposed method in comparison with word-based SVM, LM-based SVM, and LM-based Logic Regression (LR)

		LR +LM	SVM + LM	SVM (word)	Our method
Neg.	P	0.555	0.553	0.718	0.652
	R	0.693	0.692	0.540	0.721
	F	0.616	0.615	0.617	0.685 (+11%)
Non-Neg.	P	0.591	0.588	0.632	0.688
	R	0.445	0.44	0.788	0.615
	F	0.508	0.503	0.701	0.649 (-7%)

Sentiment Classifier. We compared our proposed method of automatically generating clues automatically and subsequently using a few syntactic features against three other basic SA methods: word-based SVM, Language Model (LM)-based SVM, and LM-based Logistic Regression (LR). We used unigram features for LM. The main difference among the cases is that the other three methods do not attempt to extract clues separately but just use words as features.

Though the results show both negative and non-negative cases, we mainly focus on improving the performance of detecting negative sentiment. While the word-based SVM gives a much higher F-value for the non-negative case, the three being compared against our method show similar results for the negative case. Our method outperforms all the others by about 11% in F and at the same time shows a reasonable performance for the non-negative case.

Since it was not clear whether the increase was caused by the use of candidate sets or new features beyond the basic ones, we compared the proposed method against the one without generating the candidate clues. Note that the same feature generation methods were employed for both treatments. The performance increase 12.8% allows us to conclude that the difference comes from the overall strategy of automatically generating clues and expanding the features.

Table 7. The effect of using clues (the same features were used for both cases)

		Using the same features without clues	Our method
Neg.	P	0.4863	0.6518
	R	0.8078	0.7208
	F	0.6071	0.6846 (+12.8%)

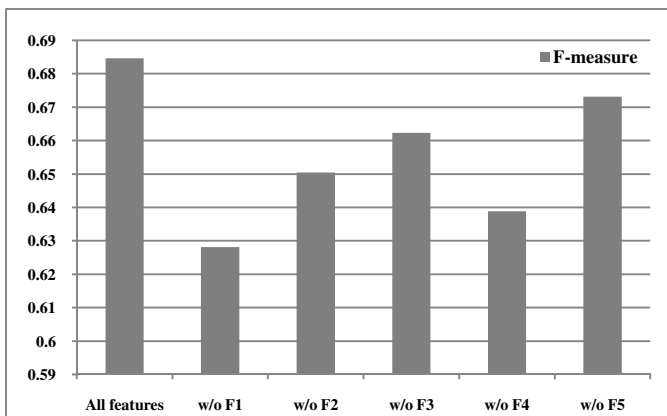


Fig. 4. The effect of excluding one feature at a time

In order to show relative importance of basic features (F1, simple frequency ratio, and F2, Z-score), syntactic features (F3, where or not the clue is associated with the main predicates, and F4 whether the clue is in the quoting predicates), and the

language-specific one (F5), we measured changes of classification performance caused by excluding one feature at time. As in Fig. 4, F1 and F4 features are the most important since the performance in negative sentiment was decreased most significantly. The effect of others cannot be disregarded because their performance was decreased at least with a small amount. The decrement of ignoring F4 is larger than that of F3. This means that the linguistic feature associated with the genre of the corpus, news articles in this case, is more important than the usual phenomenon. The feature associated with the Korean language (F5) had almost no effect.

We compared the three types of weighting methods described in Section 3.2. In the experiment, both basic and syntactic features were used. As in Table 8, the performance differences are not significant but the Z-score method shows the best. We conjecture that the Z-score method works better than the others because it uses the extent to which a word frequency in negative sentences deviates from its mean without considering the value of d in the table, frequency of a word not occurring in non-negative sentences. Since the numbers of the contingency table are skewed due to the nature of the corpus, using only the frequency information for a word in negative sentences is more reliable

Table 8. Comparison among three types of weighting methods

		Simple Frequency Ratio	Z-score	Chi-squared
Neg.	P	0.6253	0.6443	0.6519
	R	0.6985	0.7047	0.6501
	F	0.6599	0.6732	0.6510

4.3 Experimental Result – English Data

As an attempt to show that the proposed method can be applied to SA in English, we ran experiments with the NTCIR8 sentiment corpus. We generate candidate clues from the training data in the same way and show how the recall values change as the number of clues increases. Table 9 shows that the maximum recall is 83% when we use all the available data in the corpus. This turns out to be too small to even cover all the necessary clues for the testing data. This means that the SA must be performed with a limited candidate clue set, which should be inferior to a manually constructed lexicon.

Table 9. Sentiment classification performance changes with different numbers of clues

# of clues	Negative		
	Precision	Recall	F-measure
244	0.1274	0.6556	0.2134
546	0.1304	0.7781	0.2234
1,055	0.1329	0.8367	0.2293

Table 10. Overall performance comparisons with the NTCIR-8 corpus

		SentiWordNet	Our method
Neg.	P	0.5314	0.6702 (+26%)
	R	0.5957	0.4549 (-24%)
	F	0.5617	0.5419 (-3.5%)
Non-Neg.	P	0.5410	0.5886 (+9%)
	R	0.4757	0.7770 (+63%)
	F	0.5062	0.6698 (+32%)

Even with the limited candidate clue set, we developed the same SA module. The only difference was that we did not use the language-specific feature, which turned out to be marginally useful in Korean. The proposed method was compared against the case that uses a lexicon instead of our clues, SentiWordNet, which is a well-known and widely used resource for English. Table 10 shows that our method gave a 3.5% decrease in negative while the non-negative category obtains 32% improvements. The low recall in negative (0.4549) seems to be caused by the small number of clues, 1,055; the number of entries in SentiWordNet is about twenty thousand, from which 2,057 clues were actually used in our SA experiment. This result is particularly encouraging since the number of documents used to generate the clues is only 69, which yields only a 3.5% decrease in F-measure. The amount of effort required to build such a lexicon would be considerably more expensive than generating the small number of annotated articles.

5 Conclusion and Future Work

We proposed a novel approach to SA, which does not assume availability of a pre-constructed clue lexicon. The experiments showed that the frequency-based and genre-based features were most valuable. The overall performance increase shows that our proposed approach is superior to that of machine learning approaches using language modeling. As a future work, we plan to devise more extensive ways of generating new features in different categories.

References

1. Choi, Y., Cardie, C.: Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis. In: Proc. of EMNLP 2008 (2008)
2. Choi, Y., Kim, Y., Myaeng, S.-H.: Domain-specific Sentiment Analysis using Contextual Feature Generation. In: Proc. of CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement, TSA (2009)
3. Choi, Y., Jung, Y., Myaeng, S.-H.: Identifying Controversial Issues and their Sub-topics in News Articles. In: Chen, H., Chau, M., Li, S.-H., Urs, S., Srinivasa, S., Wang, G.A. (eds.) PAISI 2010. LNCS, vol. 6122, pp. 140–153. Springer, Heidelberg (2010)

4. Durant, K.T., Smith, M.D.: Predicting the Political Sentiment of Web Log Posts using Supervised Machine Learning Techniques Coupled with Feature Selection. In: Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B. (eds.) *WebKDD 2006. LNCS (LNAI)*, vol. 4811, pp. 187–206. Springer, Heidelberg (2007)
5. Esuli, A., Sebastian, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: *Proc. of the LREC 2006 (2006)*
6. Fan, T.-K., Chang, C.-H.: Sentiment-oriented Contextual Advertising. *Knowledge Information System* 23 (2010)
7. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press (1998)
8. Jia, L., Yu, C., Meng, W.: The Effect of Negation on Sentiment Analysis and Retrieval Effectiveness. In: *Proc. of CIKM 2009 (2009)*
9. Kim, S.-M., Hovy, E.: Determining the Sentiment of Opinions. In: *Proc. of COLING 2004 (2004)*
10. Kim, Y., Choi, Y., Myaeng, S.-H.: Generating Domain-specific Clues using News Corpus for Sentiment Classification. In: *Proc. of Weblogs and Social Media 2010 (2010)*
11. Melville, P., Gryc, W., Lawrence, R.: Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In: *Proc. of SIGKDD 2009 (2009)*
12. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: *Proc. of HLT/EMNLP 2005 (2005)*
13. Tan, S., Cheng, X., Wang, Y., Xu, H.: Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) *ECIR 2009. LNCS*, vol. 5478, pp. 337–349. Springer, Heidelberg (2009)
14. Pan, S., Ni, X., Sun, J.-T., Yang, Q., Chen, Z.: Cross-Domain Sentiment Classification via Spectral Feature Alignment. In: *Proc. of WWW 2010 (2010)*
15. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: *Proc. of EMNLP 2002 (2002)*
16. Stone, P., Bales, R., Namenwirth, J., Ogilvie, D.: *The General Inquirer: A Computer System for Content Analysis and Retrieval Based on the Sentence as a Unit of Information*. The MIT Press (1966)
17. Esuli, A., Sebastian, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: *Proc. of LREC 2006 (2006)*
18. Seki, Y., Ku, L.-W., Sun, L., Chen, H.-H., Kando, N.: Overview of Multilingual Opinion Analysis at NTCIR-8. In: *Proc. of NTCIR-8 (2010)*

Roles of Event Actors and Sentiment Holders in Identifying Event-Sentiment Association

Anup Kumar Kolya¹, Dipankar Das¹, Asif Ekbal², and Sivaji Bandyopadhyay¹

¹ Department of Computer Science and Engineering, Jadavpur University,
Kolkata 700 032, India

² Department of Computer Science and Engineering, IIT Patna, Patna-800013, India
{anup.kolya, dipankar.dipnil2005}@gmail.com,
sivaji_cse_ju@yahoo.com, asif@iitp.ac.in

Abstract. In this paper, we study the roles of *event actors* and *sentiment holders* from the perspective of event sentiment relations within the TimeML framework. The proposed algorithm is bootstrapping in nature that identifies the association between the event and sentiment expressions. There are two basic steps of the algorithm and they deal with lexical keyword spotting and co-reference resolution. We consider the associations between the event and sentiment expressions that are in the same or different text segments. Guided by the classical definitions of events in the TempEval-2 shared task, a manual evaluation is attempted to distinguish the sentiment events from the factual events and the agreement was satisfactory. In order to computationally estimate the different sentiments associated with different events, the knowledge of *event actors* and *sentiment holders* is introduced. To identify the roles between the *event actors* and *sentiment holders*, appropriate method is proposed. From the experiments, it is observed that the lexical equivalence between event and sentiment expressions easily identifies the similar entities that are both responsible for the *event actors* and *sentiment holders*. If the event and sentiment expressions occupy different text segments, the identification of their corresponding *event actors* and *sentiment holders* needs the knowledge of parsed-dependency relations, named entities along with the anaphors. The manual evaluation produces satisfactory results on the test documents of the TempEval-2 shared task in case of identifying the many to many associations between the *event actors* and *sentiment holders* for a specific event.

Keywords: Event Actor, Sentiment Holder, TempEval-2, Semantic Role Labeling (SRL), Name Entity Recognizer (NER).

1 Introduction

Sentiment of people with respect to an event is important as it has great influence on our society. Nowadays, in the Natural Language Processing (NLP) communities, several research activities on sentiment analysis and event tracking are in full swing but separately on their own tracks. Our main motivation to investigate the insides of event-sentiment relation lies with the facts that events and sentiments are closely

coupled with each other from social, psychological and commercial perspectives. The identification of the temporal relations between two events by taking the sentiment features into account is crucial to analyze and track human sentiments [1]. This is also important in a wide range of other NLP applications that include temporal question answering, document summarization, current information retrieval systems etc. Sometimes, similar or different types of sentiments are expressed on single or multiple events. The challenge lies in the determination of sentiments expressed in the text with respect to its reader or writer. Therefore, the extraction of sentiment holders is important for discriminating between the sentiments that are viewed from the different perspectives [2]. On the other hand, the current research trends are trying to develop and test the computational algorithms that analyze how the world views between different countries compare. This comparison is conducted using both linguistic and non-linguistic channels, such as, newspapers, blogs, and twitter and so on. In order to computationally estimate the different worldviews, we use landmark events (e.g., 2004 Indian Ocean tsunami, 2009 global stock market crash), cities (e.g., Paris, Tokyo), famous people (e.g., Michael Jackson, Osama Bin Laden), countries (e.g., Afghanistan, Pakistan), and organizations (e.g., Al-Qaeda, Royal Swedish Academy of Sciences). Thus, the determination of event actors from the text also invokes a challenge. By grouping the event actors and sentiment holders of different stance on diverse social and political issues, a better understanding of the relationships among countries or among organizations [3] can be obtained. The identification of event sentiment relations from text has started recently [4]. To the best of our knowledge, the identification of the association between event actors and sentiment holders based on their corresponding event and sentiment expressions has not yet been established in the literature.

Let us consider the following two example sentences from the TempEval-2 corpus. In Example 1, one of the annotated events is a factual event (*created*) and another is a sentiment event (*killed*) and their corresponding event actors are also different (*Israelis* and *Hezbollah and other guerrillas*). But, In Example 2, both of the annotated events are sentiment events (*killed* and *wounded*) and their corresponding event actors are also the same (*Hezbollah fighters*).

Example 1: *Since 1985, when the Israelis <created/> a security buffer zone in Southern Lebanon, nearly 200 Israeli soldiers have been <killed/> by Hezbollah and other guerrillas.*

Example 2: *Hezbollah fighters <killed/> three Israeli soldiers and seriously <wounded/> six others.*

Research works related to identifying event expressions, sentiment expressions and their relations can found in [1] [4]. The identification of event actors and sentiment holders has been discussed in [5] [6].

In this paper, we study the roles of *event actors* and *sentiment holders* from the perspective of event sentiment relations within the TimeML framework. The proposed algorithm is bootstrapping in nature that identifies the association between the event and sentiment expressions. There are two basic steps of the algorithm and they deal with lexical keyword spotting and co-reference resolution. The benchmark datasets of TempEval-2 are used for experiments. In case of *event actor* identification,

the baseline model is developed based on the *subject* information of the dependency-parsed event sentences. In order to improve the performance further, an open source Semantic Role Labeler (SRL) and an unsupervised syntax based model are used. The syntactic model is based on the relationship of the event verbs with their argument structures extracted from the *head* information of the chunks in the parsed sentences. Similarly, the baseline model for identifying sentiment holders is developed based on the *subject* information of the dependency-parsed sentences. The unsupervised syntax based model along with the named entity (NE) clues is prepared based on the relationship of the verbs with their extracted argument structures from the dependency parsed sentences. The chains of *event actors* and *sentiment holders* are also deduced based on their anaphoric presence in text. An open source implementation of anaphora resolution¹ is used for identifying the anaphoric presence of the *event actors* and *sentiment holders*. With the given input of TempEval-2 corpus, it is able to generate a list of anaphora-antecedent pairs as well as an in-place annotation or substitution of the anaphors with their antecedents. The association between an event actor and a sentiment holder is identified from the sentences that are clustered for each of the event chains. The event chains are deduced from the documents of the TempEval-2 corpus based on the temporal relations [1]. The lexical equivalence of event and sentiment expressions along with the basic clues improves the performance of the system significantly. In case of distributed presence of event expressions and sentiment expressions in different text fragments, the many to many association between event actors and sentiment holders needs the incorporation of the basic clues like named entities, parsed dependency relation along with the presence of anaphors.

The rest of the paper is organized as follows. Section 2 describes the bootstrapping approach for tagging the sentiment events using three techniques, *viz.* lexical equivalence, anaphora and/or co-reference resolution and cause-effect analysis. The identification of event actors and sentiment holders are discussed in Section 3 and Section 4 respectively. The association of event actors with sentiment holders is described in Section 5. The evaluation schemes and results of the association are mentioned in Section 6. Finally, Section 7 concludes the paper.

2 Identification of Sentiment Events

Sentiment is an important cue that effectively describes the events associated with it. Thus, the identification of the association between event actors and sentiment holders also involves the identification of sentiment events. In the present task, the coarse-grained binary classification of the sentiments (*positive* and *negative*) as well as the six fine-grained sentiment categorization into Ekman's (1993) [7] six emotions (*happy*, *sadness*, *anger*, *disgust*, *fear* and *surprise*) have been considered for identifying sentiment events. We use the TempEval-2 event tagged corpus that contains 42 training documents and 11 test documents respectively [8]. The corpus also consists of the annotation regarding the temporal information (AFTER, BEFORE, OVERLAP) between the event pairs.

¹ <http://www-appn.comp.nus.edu.sg/~rpnlpir/cgi-bin/JavaRAP/JavaRAPdemo.html>

But, no sentiment related annotation was present in the corpus. Thus, a bootstrapping approach has been developed to identify the sentiment events and their association. Along with the cause-effect analysis, the bootstrapping approach combines two techniques, *viz.* lexical equivalence followed by co-reference. To accomplish the evaluation, we manually verified 207 sentences of 11 documents of the TempEval-2 test set. The linguistic evaluation to identify the relations between factual and sentiment events was guided by the classical definitions of events in the TemEval-2 shared task.

2.1 Lexical Equivalence for Identifying Sentiment Events

The tagging of the *evaluative expressions* or more specifically the sentiment expressions on the TempEval-2 corpus is carried out using the available sentiment lexicons. We passed the sentences through three sentiment lexicons, *Subjectivity Wordlists* [9], *SentiWordNet* [10], *WordNet Affect* [11] and English *VerbNet* [12]. The *Subjectivity Wordlist* assigns words with the strong or weak subjectivity and prior polarities of types *positive*, *negative* and *neutral*. The *SentiWordNet*, used in opinion mining and sentiment analysis, assigns three sentiment scores such as *positive*, *negative* and *objective* to each synset of *WordNet* [13]. The *WordNet Affect*, a small well-used lexical resource and valuable for its affective annotation contains the words that convey emotion. The main criterion for selecting the member verbs as sentiment expressions is the presence of “*emotional_state*” type predicate in their frame semantics of the *VerbNet*. The word level sentiment expressions are tagged using the available sentiment related lexical resources [4].

If any sentiment word directly appears to specify an event word, their association is termed as lexical equivalence. The preliminary experiment shows that 57.23% of the sentiment bearing words matched as the event words in the TempEval-2 corpus. For example, the emotional verbs “*love*” and “*enjoy*” are treated as both event word as well as sentiment word.

2.2 Co-reference Based Approach for Sentiment Event Association

If the event and sentiment expressions occupy separate text spans in a sentence, a co-reference (or, anaphora) based approach is adopted for identifying their associations. The parsed dependency relations along with some basic rhetoric components, such as *nucleus*, *satellite* and *locus* help in identifying the existing co-reference relations between the event and sentiment expressions [4]. The text span that reflects the primary goal of the writer is termed as *nucleus* (marked as “{ }”) whereas the span that provides supplementary material is termed as *satellite* (marked as “[]”). The distinguished identification of *nucleus* and *satellite* as well as their separation from each other is carried out based on the *direct* and *transitive* dependency relations, *causal verbs*, *relaters* or *discourse markers*. Simultaneous presence of *locus* and *event* in the same dependency relation is considered as *direct* dependency relation and the connection of *locus* and *event* through one or more intermediate dependency relations

is identified as *transitive* dependency relation. In the following, two rules are introduced for event sentiment co-reference.

Rule1: If *locus* (sentiment expression, **great**) and event expression (**thought**) both are identified together in either *nucleus* or *satellite*, their association is termed as co-referenced.

{*When Wong Kwan <spent/> seventy million dollars for this house*}, [*he <thought/> it was a **great** deal*].

Rule2: If the event expressions (**said** and **remains**) appear in *nucleus* and *satellite* separately but they share at least one *direct* dependency relation (*cop(nervous-7, remains-5)*, *ccomp(said-2, nervous-7)*) with sentiment expressions (**nervous**), their association is considered as co-referenced. Similarly, “seen” and “wild” in the following sentence build an event sentiment pair.

{*Traders <said/> the market <remains/> extremely **nervous***} because [*the **wild** swings <seen/> on the New York Stock Exchange last week*].

2.3 Cause-Effect Based Sentiment-Event Association

Event and sentiment expressions are generally distributed in different text spans of a text. Such distributed discourse elements are related by a relatively small set (20–25) of *rhetorical relations* [14] and/or discourse coherent relations [15]. The next question that comes to the mind is which of these relations are most important from the point of view of event and sentiment relation, or, in other words, which of the above relations give more meaningful information in the process of making a judgment about any particular entity or event. Though, the exact definition of a discourse segment is a matter of debate, the interaction relation between discourse segments may be a *cause-effect* relation where conjunctions like *because*, *so* can be used to relate the segments and can be distinguished by capturing the rhetorical structure of the text.

This relation merely identifies the reason behind the occurrence of any particular event and no extra information about any sentiment behind the cause or effect. Simple scoring based on the number of positive or negative terms in the sentence will give the final sentiment. Thus, by spotting the discourse elements (e.g. *so*, *because* etc.), the sentences are identified and the events contained in these sentences are tagged as factual events. The following example shows a cause-effect relation where the two discourse segments (*I ... semester*) and (*I... exams*) are connected by the conjunction *so*. The event expressions, *study* and *pass* carry no sentiment in the cause-effect relation.

(*I did not **study** anything throughout the semester*), *so* (*I was unable to **pass** in the exams*).

2.4 Bootstrapping Approach to Sentiment Event Tagging

We have event tagged TempEval-2 training corpus which contains 42 documents and 725 sentences. But, there is no sentiment annotation. We develop a bootstrapping based method for sentiment event tagging rather than applying any machine learning

framework [1]. Each of the concept templates contains the *lexical* pattern p using a context window that contains the words around the left and the right of the *Sentiment Word* (SW) and *Event Word* $\langle EW \rangle$, e.g.,

$$p = [l_{-i} \dots l_{-3} l_{-2} l_{-1} \langle SW \rangle \dots \langle EW \rangle l_{+1} l_{+2} l_{+3} \dots l_{+i}]$$

Where, $l_{\pm i}$ are the *context* of p . We identify all the tokens surrounding a specific term in a window proposed by Bostad [16]. We need to identify contexts that are good predictors for sentiments and events in a corpus. Equivalent sentiment and event seed words play two roles in this process; they are used to learn context rules, and they are used in the application of the rules, because the rules contain information about the equality of the words in context. We learnt context based patterns from the 50 sentences of the training corpus. The manual evaluation achieves 72.34% F-score in sentiment event tagging. We follow the same technique to tag the entire training and test data. In each iteration of the algorithm, 50 sentences are tagged using the set of all patterns learned from all the previous iterations. This approach associates the sentiment properties to the event expressions. We considered the classical definitions of events in separating the sentiment events from their factual counterparts during evaluation. We have followed the guideline that was provided in the TempEval-2 shared task. It has been observed that the events which are classified as “OCCURRENCE” (*The Defense Ministry said 16 planes have **landed** so far with protective equipment against biological and chemical warfare.*) and “REPORTING” (*No injuries were **reported** over the weekend.*) in the guideline are also identified as the factual events. On the other hand, the event classes denoting “I_STATE” are identified as sentiment events. (*An occupation Israel would **love** [to end], but ...*).

3 Identification of Event Actors

It has been observed from the detailed text analysis that almost all events are associated with some actors (“*anything having existence (living or nonliving)*”), either active or passive. More generally, event actions are associated with persons or organizations and sometimes with locations. In this section, we show how we identified the actors for the events.

3.1 Subject Based Baseline Model

The input event sentences are passed through the Stanford Parser [17] to extract the dependency relationships from the parsed data. The output is checked to identify the predicates, “*nsubj*” and “*xsubj*” so that the *subject* related information in the “*nsubj*” and “*xsubj*” predicates are considered as the probable candidates of event actors. Other dependency relations are filtered out. In the following example sentence along with parsed output and dependency relations, the “*nsubj*” and “*xsubj*” dependency relations include the event words “declined” and “comment” respectively. Phoenix is identified as the event actor (“eActor” tag) for both events.

“Phoenix **declined** to **comment**.”

nsubj (**declined-2**, Phoenix-1), xsubj (**comment-4**, Phoenix-1), aux(comment-4, to-3),
xcomp(declined-2, comment-4)

3.2 Syntax Based Model

The syntax of a sentence in terms of its argument structure or sub-categorization information of the associated verb plays an important role to identify the event actors of the events in a sentence. The VerbNet [12] subcategorization frames are used throughout this experiment for identifying the event actors.

3.1.1 Syntax Acquisition from VerbNet

The existing syntax for each event verb is extracted from the VerbNet. A separate rule based argument structure acquisition system is developed in the present task for identifying the event actors. The acquired argument structures are compared against the extracted VerbNet frame syntaxes. If the acquired argument structure matches with any of the extracted frame syntaxes, the event actor corresponding to each event verb is tagged with the actor information in the appropriate slot in the sentence.

3.3.2 Argument Structure Acquisition Framework

To acquire the argument structure, Stanford Parser [17] parsed event sentences are passed through a rule based *phrasal-head* extraction system to identify the *head part* of the phrase (well-structured and bracketed) level argument structure of the sentences corresponding to the event verbs. For example, the *head* parts of the phrases are extracted to make the phrase level pattern or argument structures of the following sentences. Sentence1: “*Ramen murdered Sharon with a knife.*”

Parsed Output: (ROOT (S (NP (NNP Ramen)) (VP (VBD murdered) (NP (NNS Sharon)) (PP (IN with) (NP (DT a) (NN knife)))))) (. .))

Acquired Argument Structure: [NP VP NP PP-with]

Simplified Extracted VerbNet Frame Syntax: [<NP value="Actor"> <VERB/> <NP patient> <PREP value="with">]

3.3 SRL for Event Actor Identification

Semantic Role Label (SRL) [18] [19] plays an important role to extract target argument relationship from the semantic role labeled sentences. Here, the argument is considered as an event actor and the target is identified as the corresponding event. Let us consider the following example:

[ARG1 All sites] were [TARGET inspected] to the satisfaction of the inspection team and with full cooperation of Iraqi authorities, [ARG0 Dacey] [TARGET said]

In the first trace, [All sites] is identified as the event actor <eActor> of the corresponding event word [inspected] and second trace, [Dacey] is the event actor <eActor> of the corresponding event [said]. So using the SRL technique, the event and the corresponding event actor is found. The original F-scores of the event actor identification systems for the subject based and syntax based models are 65.98% and

70%, respectively. Adding the SRL technique for event actor identification, the F-score of the system further improves to 73%.

4 Identification of Sentiment Holder

Sentiment analysis involves identifying its associated holder and the event or topic. A sentiment holder is the person or organization that expresses the positive or negative sentiment towards a specific event or topic [20]. Prior works in the identification of opinion holders deal either only with a single opinion per sentence [21] or with several [22]. The techniques that are employed to detect the holders are based on labeling the arguments of the verbs with their semantic roles [23] or syntactic models [6], emotional knowledge base [24], machine learning based classification [25] etc. The anaphor resolution based opinion holder identification method also exploits the lexical and syntactic information [26].

Thus, like event holder identification, the baseline system for identifying sentiment holder is also developed based on the *subject* information of the parsed sentences that contain sentiment events. The parsing was attempted using Stanford Dependency Parser [17]. The lexical pattern based phrase level similarity clues containing different Part-of-Speech (*PoS*) combinations, Name Entities (NEs) and noun phrases have been considered [27].

Another way to identify sentiment holder is based on the syntactical argument structure of the sentences. The pivotal hypothesis considered in the syntactic model is based on the hypothesis followed in [28]. In English, the *head* of each chunk in the dependency-parsed output helps in constructing the syntactic argument structure with respect to the key verb. Two separate techniques are adopted for extracting the argument structure [6]. The first technique is directly based on the parsed result and the second technique is based on the *PoS* tagged and chunked corpus. The available frames of the equivalent English verbs are extracted from the English *VerbNet* [12]. Each of the acquired syntactic argument structures is mapped to all the possible frame syntaxes present for the corresponding verb in the *VerbNet*. If the acquired syntactic argument structure of a sentence matches with any of the retrieved frame syntaxes of *VerbNet*, the holder roles (e.g., *Experiencer*, *Agent*, *Actor*, *Beneficiary* etc.) associated with the *VerbNet* frame syntaxes are then assigned in the appropriate slots in the syntactic arguments of the sentence [6] [27].

It is observed that the baseline model suffers from the identification problems of sentiment holders from the passive sentences. The dependency parser based method achieves better F-Score (66.98%) than the baseline method (F-Score of 62.39%) on a collection of 4,112 sentiment sentences. The dependency parser based method fails to disambiguate mostly the arguments from adjuncts [6]. The syntactic model outperforms over baseline significantly in terms of F-Score.

5 Association between Event Actor and Sentiment Holder

In the present work we attempt to analyze the roles of event actors and sentiment holders at document level. We randomly select 20 documents from the TempEval-2 corpus, out of which 10 each are used as the training and test sets. The event actors and sentiment holders are manually tagged on the sentiment event chains. The

sentiment event chains are identified from the documents based on the AFTER, BEFORE and OVERLAP temporal relations. The length of an event chain is measured by counting the number of events associated according to their temporal ordering. The detail is mentioned in Step I. It is observed that the maximum length of an event chain in the TempEval-2 corpus is 6~7 based on the relations between the main events [8]. The sentence clusters for each of the event chains are separated from the documents. The sentences of each cluster are processed to identify the event actors and the sentiment holders following the steps mentioned below.

Step I: Based on the temporal relations, the sentences are clustered in such a way that these carry a continuous chain of events with respect to main event relations [8]. For example, an event chain based on the temporal relations ($e2 \geq e8 > e10 \geq e7$) with respect to main event is shown as follows. The corresponding sentences of this event chain are clustered and separated from the whole document.

$e2$ $e7$ BEFORE, $e7$ $e8$ OVERLAP-OR-AFTER, $e8$ $e10$ BEFORE

Step II: The event actors and the sentiment holders are identified for each sentence in a cluster. The details of the identification of the event actors and the sentiment holders have been discussed in Section 3 and Section 4, respectively.

Step III. The Stanford Dependency Parser tool ² is used for identifying the coherent relations between an event actor and the sentiment holder from a sentence. The *direct* dependency relation is identified based on the simultaneous presence of the *event actor* and the *sentiment holder* in the same dependency relation whereas the *transitive* dependency is verified if they are connected via one or more intermediate direct dependency relations that contains either an event expression or a sentiment expression. If we consider the Example 2 and its parsed dependency relations, the following *direct* and *transitive* dependency relations containing event and sentiment expression (*killed* and *wounded*) supply the hints for the simultaneous presence of event actors as well as sentiment holders (*Hezbollah fighters*). We consider both *direct* and *transitive* dependency relations for associating the *event actors* and the *sentiment holders* (*amod(fighters-2, Hezbollah-1), nsubj(killed-3, fighters-2), nsubj(wounded-9, fighters-2)*)

Step IV. If no *direct* or *transitive* dependency relation exists between an event actor and a sentiment holder, sentiment events are identified in each sentence of a sentence cluster following the method described in Section 2. Based on the hypothesis that sentiments are associated with events in case of sentiment events, both the event actor and the sentiment holder are associated with the sentiment event.

Step V: The open source JavaRAP tool is considered that takes as input the sentence clusters of the Tempeval-2 corpus and generates a list of anaphor-antecedent pairs as output as well as an in-place annotation or substitution of the anaphors with their antecedents. In some cases (“*Time and again, he endures*”)., the anaphors are detected wrongly by the system and thus the next step is followed to reduce the errors.

Step VI: The Stanford Name Entity Recognizer ³ (NER) is used to identify the names of the *persons* and *organizations* from the sentences of a cluster. If an event expression and/or a sentiment expression are present with such NE tagged entities, like *person* or *organization* in the *direct* dependency relation, the entities are

² <http://nlp.stanford.edu/software/lex-parser.shtml>

³ <http://nlp.stanford.edu/ner/index.shtml>

identified as an event actor and/or sentiment holder, respectively. For example, if any *person* type NE is wrongly tagged by JavaRAP, the anaphoric presence of the person is resolved using the hints of the *person* type NE along with the *direct* dependency relation.

6 Evaluation

Two types of criteria are proposed for evaluating the association among event actors and sentiment holders. The results are shown in Table 1. In case of Type 1 criterion, it is observed that the clustering of the sentences with respect to the event chains performs better as the reduced scope helps to incorporate the knowledge of anaphors in comparison with the whole documents. The event actors are also identified as the sentiment holders when the lexical equivalence is present for the event and sentiment expressions. But, it has been observed that the system fails to identify the appositive cases (*Ram's excitement*) and anaphoric presence (*he*) of the event actors and sentiment holders. If the event and sentiment expressions are distributed in different fragments of text, the hand-crafted rules, as described in section 5, are used along with the open source JavaRAP (Resolution of Anaphora Procedure) tool, Named Entity Recognizer and dependency parser for associating the event actors and sentiment holders with respect to their corresponding event and sentiment expressions. But, the performance of the JavaRAP tool is 57%. Thus, it does not always result in the correct anaphora ("Time and again, he endures"). This sentence is resolved as "Time and again, <Time> endures" using the process that is mentioned in Step VI.

Table 1. Accuracy (in %) of the association between Event Actors and Sentiment Holders for two types of Evaluation criteria using different features

Type 1 Evaluation	
Association between Event Actor (EA) and Sentiment Holder (SH) with respect to event chains	Accuracy (in%)
Before Clustering+ Lexical Equivalence	57.09
After Clustering + Lexical Equivalence	65.87
Clustering + Lexical Equivalence + Named Entity Recognizer	68.02
Clustering + Lexical Equivalence + Named Entity Recognizer + Dependency Parser	73.44
Clustering + Lexical Equivalence + Named Entity Recognizer + Dependency Parser + JavaRAP	76.38
Type 2 Evaluation	
Classes (No. of Sentences#)	Accuracy (in %)
Single EA Vs. Single SH (43)	79.62
Single EA Vs. Multiple SH (9)	21.56
Multiple EA Vs. Single SH (11)	52.00
Multiple EA Vs. Multiple SH (7)	13.88

In addition to the Type 1 criterion, many to many relationships criteria is also considered for associating the event actors with their corresponding sentiment holders with respect to a particular event chain. A total of 67 event chains are identified from the test set. This is termed as the Type 2 criterion. It has been observed that the system achieves high accuracy in case of only one to one relationship which has enough instances in the development and the test sets. The other three types, such as Single event actor Vs Multiple sentiment holders, Multiple event actors Vs. Single sentiment holder and Multiple event actors Vs. Multiple sentiment holders are less frequent in the corpus.

7 Conclusion and Future Work

In this paper, we studied the roles of *event actors* and *sentiment holders* from the perspective of event sentiment relations within the TimeML framework. Lexical keyword and co-reference based bootstrapping algorithms have been used to associate the event and sentiment expression. We also proposed the knowledge of *event actors* and *sentiment holders* to identify their association in text based context. The difficulty arises when the event and sentiment expressions are present in different text segments. Named Entity recognizers, JavaRAP and other tools have been used for the identification of the *event actors* and *sentiment holders* in such cases.

There is a current research trend to develop and test the computational algorithms that will analyze how the world views between different countries compare from the perspectives of sentiment. Thus, in future work, the extraction of sentiment holders is important for discriminating sentiments that are viewed from different perspectives. By grouping the event actors and sentiment holders of different stance on diverse social and political issues, a better understanding of the relationships among countries or among organizations can be obtained. In our future task, we will try to evaluate our algorithm on blogs, twitter and other social media.

References

1. Das, D., Kolya, A.K., Ekbal, A., Bandyopadhyay, S.: Temporal Analysis of Sentiment Events – A Visual Realization and Tracking. In: Gelbukh, A.F. (ed.) CICLing 2011, Part I. LNCS, vol. 6608, pp. 417–428. Springer, Heidelberg (2011)
2. Seki, Y.: Opinion Holder Extraction from Author and Authority Viewpoints. In: SIGIR 2007. ACM, New York (2007) 978-1-59593-597-7/07/0007
3. Kim, S.-M., Hovy, E.: Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. ACL (2006)
4. Kolya, A., Das, D., Ekbal, A., Bandyopadhyay, S.: Identifying Event – Sentiment Association using Lexical Equivalence and Co-reference Approaches. In: RELMS Collocated with ACL 2011, Portland, Oregon, pp. 19–27 (June 23, 2011)
5. Kolya, A., Das, D., Ekbal, A., Bandyopadhyay, S.: A Hybrid Approach for Event Extraction and Event Actor. In: RANLP, Hissar, Bulgaria, September 12-14, pp. 592–597 (2011)

6. Das, D., Bandyopadhyay, S.: Emotion Holder for Emotional Verbs – The Role of Subject and Syntax. In: Gelbukh, A. (ed.) *CICLING 2010*. LNCS, vol. 6008, pp. 385–393. Springer, Heidelberg (2010)
7. Ekman, P.: Facial expression and emotion. *American Psychologist* 48(4), 384–392 (1993)
8. Verhagen, M., Saurí, R., Caselli, T., Pustejovsky, J.: SemEval-2010 Task 13: TempEval-2. In: *SemEval ACL 2010*, pp. 57–62 (2010)
9. Carmen, B., Rada, M., Janyce, W.: A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources. In: *The Sixth International Conference on Language Resources and Evaluation* (2008)
10. Stefano, B., Andrea, E., Fabrizio, S.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: *Proceedings of the 7th Conference on Language Resources and Evaluation*, pp. 2200–2204 (2010)
11. Strapparava, C., Valitutti, A.: Wordnet-affect: an affective extension of wordnet. In: *4th International Conference on Language Resources and Evaluation*, pp. 1083–1086 (2004)
12. Kipper-Schuler, K.: VerbNet: A broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA (2005)
13. Miller, G.A.: WordNet: An on-line lexical database. *International Journal of Lexicography* 3(4), 235–312 (1990)
14. Mann, W., Thompson, S.: Rhetorical Structure Theory: Description and Construction of Text Structure. In: Kempen, G. (ed.) *Natural Language Generation*, pp. 85–96. Martinus Nijhoff, The Hague (1987)
15. Wolf, F., Gibson, E.: Representing discourse coherence: A corpus-based study. *Computational Linguistics* 31(2), 249–287 (2005); Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge (2000)
16. Bostad, T.: Sentence Based Automatic Sentiment Classification. Ph. D. thesis, University of Cambridge, Computer Speech Text and Internet Technologies (CSTIT), Computer Laboratory (2003)
17. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: *5th International Conference on Language Resources and Evaluation* (2006)
18. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. *Computational Linguistics* 28(3), 245–288 (2002)
19. Pradhan, S., Ward, W., Hacioglu, K., Martin, J.H., Jurafsky, D.: Shallow Semantic Parsing using Support Vector Machines. In: *HLT/NAACL 2004*, Boston, MA (2004)
20. Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D.: Automatic Extraction of Opinion Propositions and their Holders. In: *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications* (2004)
21. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *LRE* 39(2-3), 165–210 (2005)
22. Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In: *Proceedings of HLT/EMNLP 2005* (2005)
23. Swier, R.S., Stevenson, S.: Unsupervised Semantic Role Labelling. In: *EMNLP* (2004)
24. Hu, J., Guan, C., Wang, M., Lin, F.: Model of Emotional Agent. In: Shi, Z.-Z., Sadananda, R. (eds.) *PRIMA 2006*. LNCS (LNAI), vol. 4088, pp. 534–539. Springer, Heidelberg (2006)

25. Evans, D.K.: A low-resources approach to Opinion Analysis: Machine Learning and Simple Approaches. NTCIR (2007)
26. Kim, Y., Jung, Y., Myaeng, S.-H.: Identifying Opinion Holders in Opinion Text from Online Newspapers. In: 2007 IEEE International Conference on Granular Computing, pp. 699–702 (2007), doi:10.1109/GrC.2007.45
27. Das, D., Bandyopadhyay, S.: Identifying Emotion Holder and Topic from Bengali Emotional Sentences. In: ICON 2010, IIT Kharagpur, India (2010)
28. Banerjee, S., Das, D., Bandyopadhyay, S.: Classification of Verbs—Towards Developing a Bengali Verb Subcategorization Lexicon. In: GWC 2010, Mumbai, India, pp. 76–83 (2010)

Applying Sentiment and Social Network Analysis in User Modeling

Mohammadreza Shams, Mohammadtaghi Saffar, Azadeh Shakery, and Heshaam Faili

School of Electrical and Computer Engineering, College of Engineering,
University of Tehran, Tehran, Iran

{m.shams,saffar,shakery,hfaili}@ut.ac.ir

Abstract. The idea of applying a conjunction of sentiment and social network analysis to improve the performance of applications has recently attracted attention of researchers. In widely used online shopping websites, customers can provide reviews about a product. Also a number of relations like friendship, trust and similarity between products or users are being formed. In this paper a combination of sentiment analysis and social network analysis is employed for extracting classification rules for each customer. These rules represent customers' preferences for each cluster of products and can be seen as a user model. The combination helps the system to classify products based on customers' interests. We compared the results of our proposed method with a baseline method with no social network analysis. The experiments on Amazon's meta-data collection show improvements in the performance of the classification rules compared to the baseline method.

Keywords: sentiment analysis, social network analysis, graph clustering, polarity classification, user model.

1 Introduction

The growing use of Internet as a tool to help people do their daily financial routines, gives the companies the opportunity to use this media to increase their sales rate by providing online shopping websites. The huge amount of reviews of products and items in online shopping solutions like Amazon.com [2] can help managers and decision makers to find out more about the preferences of customers, their needs, and also the rate and direction of change in their interests. Another source of knowledge available in online shopping websites is social networks formed between customers or products.

Sentiment analysis, also referred as opinion mining, is the process of analyzing the characteristics of opinions, feelings and emotions expressed in textual data provided for a certain topic or object [13]. In several and diverse applications, the knowledge about "what others think" is one of the most important pieces of knowledge in decision making process. The procedure of obtaining the knowledge is sentiment analysis. Consider a political campaign before an important election. The campaign can make strategic decisions to get more support from the people with the help of knowing what

the society thinks about the policies and political views of it. Another example would be online shopping solutions, like Amazon.com. Users of such systems can provide textual reviews about items. These data can be regarded as a valuable source of information for a proper sentiment analysis process. Outcomes can help recommender systems to understand a particular user's interests and preferences better, so the system would be able to offer more suitable items to the user.

A social network is a graph consisting of individuals (or organizations) called nodes and edges which interconnect them with different types of interdependency, such as friendship, affiliation relationship, communication mode or relationship of religions, interest or prestige [19]. Social network analysis is used to find the relationships resided in the graph via network theory. Basically graph theory is one of the most important tools used to conduct the analysis. Since the social network contains complicated interactions between individuals from different aspects the analysis is too convoluted which cannot be done only by the traditional methods, and the produced graph-based structure is often very complex.

However, recent research studies on opinion mining are mainly based on processing of textual data. Another source of knowledge, which was almost ignored in this area until recently, is the social network. Information about relationships among users or items can be used to improve the results of opinion mining. Recently, this idea is obtaining more attention. Some algorithms have been proposed to combine textual data with relationship links between objects or users to develop sentiment analysis results.

In this paper, we propose a method to find a Rule Based User Model (RBUM) by combining sentiment and social network analysis. The RBUM can be used to enhance different applications like recommender systems, customer relationship management systems or personal assistant agents. Although, methods that combine sentiment and social network analysis have been used in other domains, most of the previous work on constructing user models is using one of the data sources; textual data or relationships between entities. Our proposed method uses textual reviews and relationships between products together to make the user models more accurate.

The rest of the paper is organized as follows. In section 2, several relevant methods in the domain of sentiment analysis and social network analysis are introduced. In section 3, the overall RBUM process is presented. In subsection 3.1 the process of sentiment analysis applied in this study is described; followed by the description of formation and community detection of the social network of products in 3.2. Subsection 3.3 presents the classification rules generator method. After that, some explanations of the implementation and evaluation of the proposed system are introduced in section 4. Conclusions and future directions are explained in the last section, section 5.

2 Previous Work

In this section, we review the previous related research in three different areas: social network analysis, sentiment analysis, and combination of social network with sentiment analysis.

Social network analysis: One of the main tasks in social network analysis is to capture the community structure of networks. Community structure has been recognized as an important statistical feature of networked systems over the past decade. Generally, a community within a network is a sub-graph whose nodes are densely interconnected within the group but are loosely connected with the rest of the network. From this perspective, finding such groups of nodes in a network could be treated as a clustering problem, i.e. satisfying the criterion that the nodes within each cluster have dense connections with each other but weak connections with other nodes in other clusters. Two main classes of clustering algorithms are hierarchical clustering methods and partitioning methods. Hierarchical clustering methods create a hierarchy over the nodes. These methods can be classified as agglomerative or divisive. Agglomerative approaches start from the individual isolated nodes and aggregate those which are close or similar enough with each other, producing larger groups. Divisive methods consider all nodes in one group initially and try to divide that group to create smaller clusters until a termination condition is met. On the other hand, partitioning methods assign nodes to separate clusters continuously by following some optimization rules [8]. K-Means or K-Nodes are two examples.

Finding communities in social networks may seem a trivial task. But according to the complex nature of social graphs and their size, simple graph clustering algorithms perform poorly in this field [17]. A number of graph clustering algorithms have been introduced specially for detecting the communities in social networks [11], [12], [18] and [20].

Sentiment analysis: Several methods have been proposed in the domain of sentiment analysis for polarity classification task. Some of them focus on polarity-based feature selection methods [16], such as document frequency, Chi Square or polarity selection, and some others are combining rule-based [9], supervised or unsupervised classification methods [14], such as k-NN, Naive Bayes and Support Vector Machine (SVM).

Most of published evaluations show that the combination of sentiment-based feature selection and machine learning algorithms produces the best performance with respect to classification accuracy. However, almost all of the approaches employ an external resource in order to detect and extract polarity-related term features in text [13].

Combination of sentiment analysis and social network analysis: There have been a few research studies that use a combination of sentiment analysis with social network analysis in some domains. One of the most important works in this field is [7] doing political blog analysis. In this research, the authors studied modeling blogosphere sentiment focused on Barack Obama during the 2008 U.S. presidential election, and described a series of initial sentiment classification experiments on a dataset of 700 crowd-sourced posts labeled as ‘positive’, ‘negative’, ‘neutral’, or ‘not applicable’ with respect to president Obama. They used a hybrid machine learning and logic-based framework. The results show that the classification task in this environment is inherently complex, and learning features that exploit entity level sentiment and social network structure has a positive effect on the result. Another example of combining

sentiment analysis with social network analysis is a work done in [3]. In this research, the concept of online radicalization has been studied. They monitored users and interactions in a specific YouTube group using a combination of sentiment, lexical and social network analysis techniques. They made a number of interesting observations about the differences in the nature of the discussion and interactions between the male and female members of the group.

3 Rule Based User Modeling (RBUM) Method

In this research, we propose to use both sentiment analysis and social network analysis for user modeling. Specifically we used the costumers' reviews provided in an online shopping website (Amazon.com) and combined it with the relationships between products to predict costumers' interests in every product category. Our proposed algorithm consists of three main steps. In the first step, customers' reviews are fed into the sentiment analysis algorithm. The result of sentiment analysis is the polarity classification of customers' reviews. This set of tagged reviews will help find classification rules for each customer in order to predict his/her preferences about products.

In the second step, a product graph is constructed from the similar products. This graph is a weighted undirected network which represents products and their similarities. After that, a graph clustering algorithm is used to cluster the product's graph, to form different categories for products. This step is important because customers have different preferences in different groups of products and each category should be considered individually and independently.

In the third and final step, the results of sentiment analysis and the result of product's graph clustering are combined together for applying a rule generation algorithm called RIPPER. For each category of products and each customer this step should be done independently. The result of this step is a number of association rules for each user and each category, which describes the interests and preferences of the customer in that category. These rules can be used in enhancing user models in recommender systems, and also can help in prediction of someone's interest in a product. Each step is explained in detail in the following subsections. In Fig. 1, the overall steps of this procedure are shown.

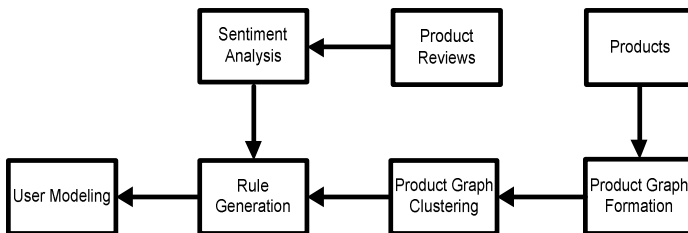


Fig. 1. RBUM steps

3.1 Applying Sentiment Analysis

Textual reviews provided by a customer about a product can be regarded as judgments and his/her interests on that product. But these reviews first need to get classified based on their polarity. To calculate the polarity of customers' reviews, initially we used a system that was developed for participation in TREC 2008 [4]. This system focuses on opinion polarity detection in a large scale blog corpus. It applies three modules called Lexicon, Surface and Syntactic. Each of these modules assigns a score for every blog post. This system fuses the results of the modules by calculating a weighted average on the given scores to decide the polarity of every blog post in its dataset. However differences in the nature of the data that we used in this paper and the blog corpus, made the results of applying [4] on our data, unreliable.

In Amazon.com, most reviews are considerably much shorter than the typical blog length in the TREC corpus of hundreds of sentences. Another major difference is that there is little evidence of subjectivity in the Amazon.com texts. Often, when Amazon users express their opinions, they simply state it rather than qualifying it with "I think ..." or "I feel ...". This behavior is not seen in the blog corpus where authors are keen to distinguish opinion from fact in their posts. All of this means that the components in the TREC system designed to detect subjectivity are not useful in the context of the Amazon data, and so hindered performance.

For these reasons, we decided to use a hybrid SVM method for polarity classification. Two types of features were employed in the classification process. The first type of features was generated by a lexicon module. This module uses a lexicon named SentiWordNet3 [6], which assigns positivity and negativity scores to synset entries in the WordNet. The average of positive, negative and neutral scores of words in each review are the features added by the lexicon module to that review. The second feature type was the term frequency of unigrams. For each review the term frequency of unigrams is a vector that contains term frequency of each vocabulary word [21].

Polarity detection is topic sensitive. Customer reviews are in different categories like books, electronics, etc. so a single SVM classifier should be used and trained individually for each of the categories. The features generated by the two parts of method are fed into SVM classifier so each review in each product category is classified either in positive, negative or neutral group.

3.2 Applying Social Network Analysis

The final goal of this paper is to model preferences of users for different products. Usually someone likes or dislikes some features in a category of products which can be quite the opposite for the same person in another group of products. A logical decision to handle this situation is to treat different classes separately. To do so, first the category of each product should be determined. The products and their similarity relations can be modeled as a graph. Each node in the graph represents a single product and an undirected weighted edge between two nodes represents the similarity between the nodes. In order to compute a similarity weight for two products, an assumption can be used: the nodes with more connections have stronger relations with

each other than the nodes with fewer connections. A product is connected to another one, if customers have bought these two items together frequently. Because of the direct influence of customers' behaviors in the connectedness relation between products, the products graph can be regarded as a social network.

Building the Social Graph for Products. In many applications, the relationship scores between entities are available in a zero/one format. For example ‘friendship’ or ‘who trust whom’ relations have an assigned binary score. But these concepts are inherently weighted. As someone can easily talk about how much he/she trusts someone else.

In order to compute a weight for the relation between two entities, the size of common connected nodes can be used. We compute the weight of an edge connecting nodes D_i and D_j in the products graph as:

$$w(D_i, D_j) = \begin{cases} 0 & D_i \text{ and } D_j \text{ are not connected} \\ \frac{|connected(D_i) \cap connected(D_j)|}{|connected(D_i) \cup connected(D_j)|} & D_i \text{ and } D_j \text{ are connected} \end{cases} \quad (1)$$

Where $connected(D_i)$ is the set of entities connected to the node D_i . The assumption of this formula is that when two items have more common connected items, they are more similar to each other.

People usually have different tastes and preferences about the products in each category. For example a person might like romantic novels among the books and like music with blues genre. This observation points out the importance of analyzing different categories independently to discover user’s preferences and interests in each of the classes. Granularity of categories is very important for the task of obtaining user interests. For instance, the general category of books can have several sub-categories like novels, historical, scientific, etc. If sub-categories were analyzed instead, more specific and precise rules could be generated within that smaller sub-category. In order to find these sub-categories with adjustable granularities, one can use a suitable clustering algorithm on the social graph of products.

Clustering the Graph of Products. In this work, we propose to use Markov CLustering algorithm (MCL) to cluster the products graph. MCL is a fast and scalable unsupervised clustering algorithm for networks based on simulation of stochastic flow in graphs [10]. It has been widely used in social network analysis context. This algorithm is suitable for our case, because it works perfectly with large graphs and the clustering results granularity can be controlled by input parameters to the algorithm. The result of applying MCL on the products graph is a categorization of similar products with a customized granularity. If the parameters are properly calibrated, the results of clustering are the same as the category labels in the dataset. If a different granularity is chosen, sub-categories can be discovered.

MCL uses a stochastic matrix to represent the probability of moving from one node to the other. It also uses two operations called inflation and expansion. Inflation (Γ_r) is an operator which powers each cell with $r > 1$ then it normalizes each column of the matrix. Inflation is defined in equation 2.

$$\Gamma_r = \frac{M_{ij}r}{\sum_{r,j}(M)} \quad (2)$$

The expansion operator is simply the square of stochastic matrix. It represents the probability of moving from a node to another by moving exactly two edges. MCL alternatively applies inflation and expansion operations until a stopping criterion is reached. The parameter r in the inflation operator, determines the granularity of the resulting clusters. A detailed explanation of MCL is presented in [17].

3.3 Preference Rule Extraction

In the two previous steps, a set of categories of products was formed from products graph analysis and the reviews of users were categorized in three different groups of positive, neutral and negative by sentiment analysis.

In this step, the results of these two steps are used together in order to obtain association rules for each category of products. These association rules are considered as the user model that represents interests and preferences of the user in each category of products. Since there are different types of products available for the user, applying a rule generation algorithm on the whole dataset is not a smart choice, because it may result in generation of rules that have attributes from different categories of products that are not related to each other. That is why we propose to apply a clustering algorithm on the products and to consider each category of products independently for rule generation. This step can be repeated for every user in the system.

To generate RBUM, first all the reviews posted from the user on the system are extracted from the dataset. In the next step, each review is analyzed using the procedure explained in subsection 3.1 and its polarity class is determined. Each review is assigned to a category from the set positive, neutral and negative.

After finishing polarity classification, reviews further get spliced into the categories of products they belong to. These sets of reviews are then used as the input of a rule generation algorithm. Before passing these reviews to the rule generation algorithm, there is another major problem that needs to get resolved. Some users are not likely to provide enough reviews for some category of products to help the next step in generating accurate classification rules. This data sparseness in product reviews of a user may make the resulting preference rules untrustworthy and of low quality. To deal with this problem an augmentation approach is introduced that tries to add related reviews from the same user to the set of his/her reviews for a product category, if the set is too small.

In each sub-class of products there can be a number of product reviews for a user. If there are a limited number of reviews in a sub-class the preference rules can get biased and may not represent the real preferences of the user. To tackle this problem the set of product reviews of a user for a sub-class are augmented by adding other product reviews of that user from another sub-class with respect to a calculated similarity of the two sub-classes: $sim(C_i, C_j)$.

$$sim(C_i, C_j) = \begin{cases} 1 & \text{if } i = j \\ \frac{\sum_{p_i \in C_i} \sum_{p_j \in C_j} sim(p_i, p_j)}{|C_i| \times |C_j|} & \text{otherwise} \end{cases} \quad (3)$$

$sim(C_i, C_j)$ is simply the average of pairwise products' similarities in two sub-classes where C_i and C_j are two sub-classes and p_i and p_j are products from C_i and C_j respectively. The product reviews of the desired sub-class is augmented by the reviews of other classes based on these computed similarities. A minimum threshold (θ) represents the minimum number of product reviews needed before the rule generation phase begins. When a user has fewer product reviews in a category of products, other reviews from the same user in other categories are added to this category until the threshold condition is satisfied. The number of additional reviews from a category is calculated by the function $additional_u(C_i, C_j)$:

$$additional_u(C_i, C_j) = \begin{cases} 0 & \text{if } |reviews(C_i)| \leq \theta \\ \left\lceil \frac{sim(C_i, C_j)}{\sum_{k \in [1, m], k \neq i} sim(C_i, C_k)} \times (\theta - |reviews(C_i)|) \right\rceil & \text{otherwise} \end{cases} \quad (4)$$

Where m is the number of subclasses, $reviews(C_i)$ is the set of reviews for category C_i and $additional(C_i, C_j)$ is the number of reviews from category C_j which are added to category C_i . This function simply calculates the distribution of the number of reviews from sub-classes needed for augmentation in accordance to their relative similarity with the desired sub-class. The augmenting process then selects this number of reviews randomly and adds them to the reviews of the desired category of products so that the rule generation method can work on a reasonable number of reviews. If there are less number of reviews available in a category than it is needed for augmenting we have to choose all reviews from that category and then recalculate the additional function for the rest of the categories.

Finally for each set of reviews, a rule generation algorithm like RIPPER [5] is used to extract classification rules for products based on user's reviews. After applying the whole process, we obtain a set of association rules for each user.

RIPPER is a sequential covering algorithm. Unlike decision tree algorithms which try to produce a complete set of rules simultaneously, sequential covering algorithms find the best possible rule for each class, one by one. After finding a rule, all the sample tuples which are matched by that rule are removed from training set and the process of rule generation repeats until there is no more sample tuples left or a stopping criterion is reached. In each round of sequential covering algorithms, the most accurate rule is extracted. A rule is said to be accurate if it covers the most possible tuples from the desired class and the least possible tuples from other classes.

4 Experiment Results

To implement our RBUM method, a dataset called Amazon-metadata [1] obtained from Stanford Large Network Dataset Collection [15] (SNAP) has been used. The data was collected by crawling Amazon website and contains product metadata and

review information about 548,552 different products (Books, music CDs, DVDs and VHS video tapes). Table 1 shows some features of this dataset.

Table 1. Amazon-metadata features

Products	548,552
Product-Project Edges	1,788,725
Reviews	7,781,990
Product category memberships	2,509,699
Products by product group	Books 393561
	DVDs 19828
	Music CDs 103144
	VHS video tapes 26132

In Table 2 the overall information about the clustering phase is presented along with the minimum and maximum size of the detected clusters.

Table 2. A general description of the clustering phase

Category	Number of Clusters	Maximum Cluster Size	Minimum Cluster Size
Books	53	47,371	802
DVDs	17	3231	189
Music CDs	25	6299	1001
VHS	16	3788	206

Table 3. A sample of clustering results in the Amazon dataset

Book		VHS	Music
Cluster 1.Book	Cluster 2.Book	Cluster 1.VHS	Cluster 1.Music
Data Mining: Concepts and Techniques	The Catcher in the Rye	The Jungle Book (Disney)	Every Teardrop Is A Waterfall
Data Mining: Practical Machine Learning Tools and Techniques, Second Edition	To Kill A Mockingbird: 50th Anniversary edition	101 Dalmatians Platinum Edition	X & Y
Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management	Catch-22	Pinocchio	The A Team
Handbook of Statistical Analysis and Data Mining Applications	Nineteen Eighty-four	Sleeping Beauty	Save The World
Data Mining Techniques in CRM: Inside Customer Segmentation	Animal Farm: A Fairy Story	Robin Hood	A Rush Of Blood To The Head

In Table 3 a sample of the clusters found in the Amazon metadata dataset is presented. As it is shown, the items in each cluster are very similar and the general categories of products like books are divided to finer clusters.

The list of features used for a product in this research is the list of sub-categories that the dataset presents for that product. These sub-categories can be seen as properties of the product. For example, as shown in Fig. 2, in the book category, an item can be in different sub-categories like literature and art, drama, etc. and the author or publish dates are other sub-categories for the books. These features comprise the feature vector used by the rule extraction algorithm.

```

Id: 15
ASIN: 1559362022
title: Wake Up and Smell the Coffee
group: Book salesrank: 518927
similar: 5 1559360968 1559361247 1559360828 1559361018 0743214552
categories: 3
  | Books[283155] | Subjects[1000] | Literature & Fiction[17] | Drama[2159] | United States[2160]
  | Books[283155] | Subjects[1000] | Arts & Photography[1] | Performing Arts[521000] | Theater[2154] | General[2218]
  | Books[283155] | Subjects[1000] | Literature & Fiction[17] | Authors, A-Z[70021] | ( B ) [70023] | Bosnian, Eric[70116]
reviews: total: 8 downloaded: 8 avg rating: 4
2002-5-13 customer: A2IGOA66Y6O8TQ rating: 5 votes: 3 helpful: 2
2002-6-17 customer: A2OIN4AUH84KNE rating: 5 votes: 2 helpful: 1
2003-1-2 customer: A2HN382JNT1CIU rating: 1 votes: 6 helpful: 1
2003-6-7 customer: A2FDJ79LUDU4O18 rating: 4 votes: 1 helpful: 1
2003-6-27 customer: A39QMV9ZKRJXO5 rating: 4 votes: 1 helpful: 1
2004-2-17 customer: AUJVMSTQ1TXDI rating: 1 votes: 2 helpful: 0
2004-2-24 customer: A2C5K0QTL9UAT rating: 5 votes: 2 helpful: 2
2004-10-13 customer: ASXYF0Z3UH4HB rating: 5 votes: 1 helpful: 1

```

Fig. 2. A sample tuple from Amazon-metadata corpus

In order to evaluate the results of the sentiment analysis step, we applied SentiWordNet3, SVM and Hybrid-SVM methods to compare the accuracy of each method according to polarity classification task. We used the set of already rated reviews available in the dataset as the baseline for comparing the methods by 5-fold cross validation. Each review might have a rate from 1 to 5. We further spliced this rating into three categories. The ratings 1 and 2 were assigned to negative class, the rating 3 was assigned to neutral class, and the ratings 4 and 5 were assigned to positive class. The results are shown in Table 4. As the results suggest the Hybrid-SVM method performs better compared to the other methods. The accuracy measure is defined in Equation 5.

$$accuracy = \frac{\text{number of items classified correctly}}{\text{number of all items}} \quad (5)$$

Table 4. Accuracy of polarity classifiers

Category	Book	VHS	DVD	Music
SentiWordNet3	0.64	0.63	0.61	0.65
SVM	0.69	0.61	0.62	0.69
Hybrid-SVM	0.75	0.68	0.62	0.71

To determine the threshold of the minimum number of comments required for rule generation, we randomly selected 100 users for each major category of products who provided less than five reviews in that category. We then augmented the reviews with the proposed method with different values of threshold and calculated the precision of the generated rules. The results, presented in Fig. 3, show that for the Amazon meta-data collection the value 17 is a reasonable choice for this threshold. We continued to evaluate our method with the threshold of 17 reviews for the augmentation phase.

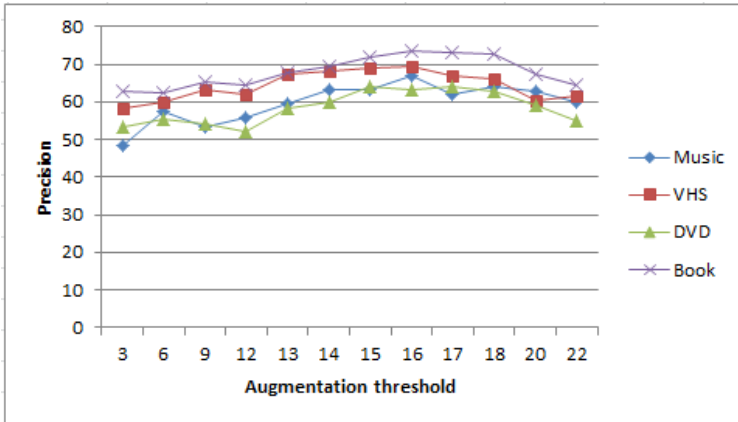


Fig. 3. Precision of the generated rules for different values of augmentation threshold

The rules extracted by RIPPER explain what users like in each category of products, what products they feel neutral about and what items they don't like in that category. Based on these rules, a recommender system can augment its user model and offer more relevant items to the users. A number of example extracted rules are shown in Table 5.

Table 5. Extracted rules for a random user

Category	Rule
Book	$subject \in Drama \wedge publish\ date > 1990 \rightarrow like$
Book	$author = Gabriel\ Garcia\ Marquez \rightarrow like$
VHS	$Genre = Drama \wedge Actors \ \& \ Actresses = Daniell, Henry \rightarrow dislike$
DVD	$Country = United\ Kingdom \wedge Type = TV \rightarrow dislike$
Music	$Category = Music\ for\ travelers \wedge Style = International \wedge Country = Greece \rightarrow like$

To evaluate the performance of classification rules, 100 users were selected randomly and then the precision and recall of the rules obtained in the final stage for these users were calculated. These measurements were compared to a baseline method that is our proposed method without the clustering of products step. The baseline method has two stages identical to sentiment analysis and classification rule extraction steps in our method and uses the four general classes of products (Book, VHS,

DVD and Music) as its product categories. The precision and recall measures are defined in Equation 6 and the comparison of the two methods is shown in Table 6. $Precision(c, user_i)$ and $recall(c, user_i)$ are the precision and recall of generated rules for $user_i$ and the category c respectively.

$$precision(user_i) = \sum_{c \in category} \frac{precision(c, user_i)}{|category|} \quad (6)$$

$$recall(user_i) = \sum_{c \in category} \frac{recall(c, user_i)}{|category|}$$

$$Precision = \sum_{for\ every\ user} \frac{precision(user)}{|user|}$$

$$Recall = \sum_{for\ every\ user} \frac{recall(user)}{|user|}$$

Table 6. Comparing precision and recall of extracted rules

	Precision	Recall
RBUM method	78.16	72.25
Baseline method	73.05	68.86

The precision and recall values of the two methods for different categories of products are shown in Fig. 4 and Fig. 5 respectively.

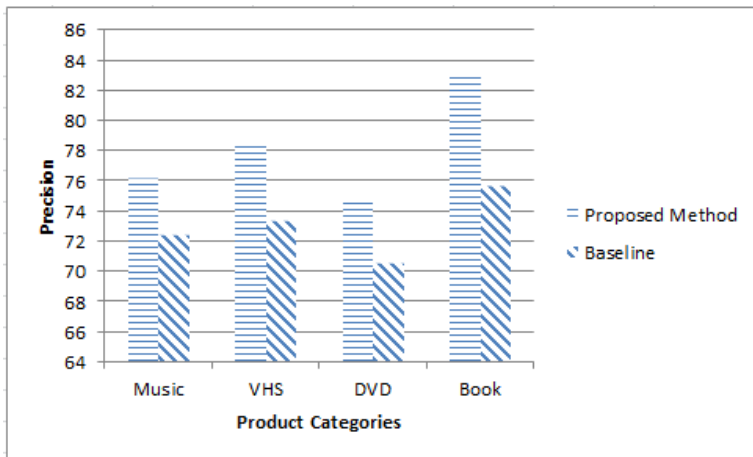


Fig. 4. Comparing precision for different categories

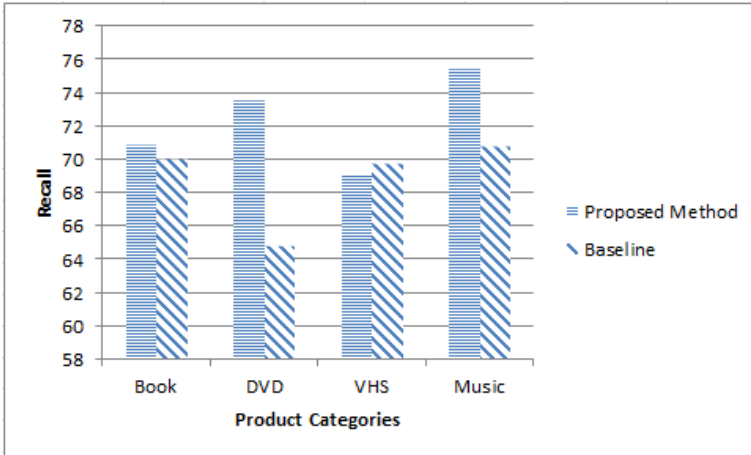


Fig. 5. Comparing recall for different categories

As the results show, our proposed method performs better than the baseline method in all categories of products. The only feature of the proposed method missing in the baseline method is the product's graph analysis. It can be concluded from this experiment that a user has different tastes and preferences about items in different categories and the categories like books, DVDs, VHSs and music are too general for a customer to have a consistent taste in them. In order to extract the preferences of a user it is more beneficial to consider more specific categories separately.

5 Conclusions and Future Directions

In this paper, a new method for combining the results of pure textual sentiment analysis and social network analysis has been presented which improves the user modeling process. Experiment results on the Amazon's metadata dataset show that the proposed method which uses social networks can improve the performance compared to a system based only on textual processing of reviews.

We can improve this work by taking into account the relations like 'friendship' or 'who trust whom' between users to further improve the user model. Also it can get better by considering other features of the reviews like the time and the location of reviewers. Some interesting classification rules can also get extracted for people living in the same country or city. Finally an interesting study that can be done on the results of this work is to try to find the similarities in interests and preferences of the users and cluster the users to find the groups with similar tastes in different groups of products. This clustering of users can also help to fill the gaps of user model of newly arrived customers who haven't provided enough reviews.

Acknowledgments. This research is partially supported by Research Institute for ICT (ITRC).

References

1. Amazon metadata dataset, <http://snap.stanford.edu/data/amazon-meta.html>
2. Amazon online shopping website, <http://www.amazon.com>
3. Bermingham, A., Conway, M., McInerney, L., O'Hare, N., Smeaton, A.F.: Combining Social Network Analysis and Sentiment Analysis to Explore the Potential for Online Radicalisation. In: IEEE International Conference on Advances in Social Network Analysis and Mining, Washington, DC, USA, pp. 231–236 (2009)
4. Bermingham, A., Smeaton, A., Foster, J., Hogan, D.: DCU at the TREC 2008 blog track. In: TREC 2008: Proceedings of the Sixteenth Text Retrieval Conference, Gaithersburg, Maryland, USA (2008)
5. Cohen, W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, San Francisco, pp. 115–123 (1995)
6. Esuli, A., Sebastiani, F., Baccianella, S.: SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010), Valletta, Malta (2010)
7. Gryc, W., Moilanen, K.: Leveraging Textual Sentiment Analysis with Social Network Modeling: Sentiment analysis of political blogs in the 2008 U.S. presidential election. In: Proceedings of the From Text to Political Positions Workshop. Vrije Universiteit, Amsterdam (2010)
8. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers (2006)
9. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22(2), 110–125 (2006)
10. MCL graph clustering tool, <http://www.micans.org/mcl>
11. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* 69, 026113 (2004)
12. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* 69, 066133 (2004)
13. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations Trends Inf. Retrieval* 2(1-2), 1–135 (2008)
14. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. *J. Informetrics* 3(2), 143–157 (2009)
15. Stanford large network dataset collection, <http://snap.stanford.edu/data/>
16. Tan, S., Zhang, J.: An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications* 34(4), 2622–2629 (2008)
17. Van Dongen, S.: Graph Clustering by Flow Simulation. Ph.D. thesis, University of Utrecht (2000)
18. Wei, F., Qian, W., Wang, C., Zhou, A.: Detecting overlapping community structures in networks. *World Wide Web* 12(2), 235–261 (2009)
19. Yang, S., Knoke, D.: *Social Network Analysis (Quantitative Applications in the Social Sciences)*, 2nd edn. Sage Publications (2007)
20. Zhang, S., Wang, R., Zhang, X.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications* 374(1), 483–490 (2007)
21. Zhe Xu, B.S.: A Sentiment Analysis Model Integrating Multiple Algorithms and Diverse Features. M.Sc. Thesis, Ohio State University (2010)

The 5W Structure for Sentiment Summarization-Visualization-Tracking

Amitava Das¹, Sivaji Bandyopadhyay², and Björn Gambäck¹

¹ Department of Computer and Information Science (IDI)
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
Sem Sælands Vei 7-9, NO - 7491, Trondheim, Norway

² Department of Computer and Engineering
Jadavpur University
Kolkata-700032, India
amiava.santu@gmail.com, sivaji_cse_ju@yahoo.com,
gamback@idi.ntnu.no

Abstract. In this paper we address the Sentiment Analysis problem from the end user's perspective. An end user might desire an automated at-a-glance presentation of the main points made in a single review or how opinion changes time to time over multiple documents. To meet the requirement we propose a relatively generic opinion 5Ws structurization, further used for *textual and visual summary and tracking*. The 5W task seeks to extract the semantic constituents in a natural language sentence by distilling it into the answers to the 5W questions: **Who**, **What**, **When**, **Where** and **Why**. The visualization system facilitates users to generate sentiment tracking with textual summary and sentiment polarity wise graph based on any dimension or combination of dimensions as they want i.e. "**Who**" are the actors and "**What**" are their sentiment regarding any topic, changes in sentiment during "**When**" and "**Where**" and the reasons for change in sentiment as "**Why**".

Keywords: 5W Sentiment Structurization, Sentiment Summarization, Sentiment Visualization and Sentiment Tracking.

1 What Previous Studies Suggest, Opinion Summary: Topic-Wise, Polarity-Wise or other-Wise?

Aggregation of information is the necessity from the end user's perspective but it is nearly impossible to make consensus about the output format or how the data should be aggregated. Researchers tried with various types of output format like textual or visual summary or overall tracking with time dimension. The next key issue is "**How the data should be aggregated?**" and "**What is the End User's requirement?**". Dasgupta and Ng [1] throw an important question: "**Topic-wise, Sentiment-wise, or Otherwise?**" about the opinion summary generation techniques. Instead of digging for the answer of the unresolved debate we experimented with multiple outputs formats. At first we will look into the topic-wise, polarity-wise and other-wise summarization

systems proposed by various previous researchers and then will describe the systems developed by us.

Topic-Wise: There is clearly a tight connection between extraction of topic-based information from a single document and topic-based summarization of that document, since the information that is pulled out can serve as a summary; see [2] for a brief review (Section 5.1). Obviously, this connection between extraction and summarization holds in the case of sentiment-based summarization, as well. There are various topic-opinion [4], [5] summarization systems, proposed by the previous researchers. Leveraging existing topic-based technologies is the most common practice for sentiment summarization. One line of practice is to adapt existing topic-based multi-document summarization algorithms to the sentiment setting. Sometimes the adaptation consists simply of modifying [3], [6] the input to these pre-existing algorithms.

Polarity-Wise: Indeed the topic-opinion model is the most popular one but there could be a requirement at the end user's perspective that they might look into an at-a-glance presentation of opinion-oriented summary. For example: One market surveyor from company *A* might be interested in the root cause for why their product *X* (suppose camera) become less popular day by day. And for this particular case *A* may want to look into for the *negative* reviews only. Therefore opinion-oriented summary is the end user's requirement here. Relatively a few research efforts could be found on the polarity-wise summarization in the literature than the popular topic-opinion model. There are a few important related works [8], [9] which are significant in both the aspects: problem definition and solution architecture with best of our knowledge.

Visualization: To convey all the automatically extracted knowledge to the end user concisely the graphical or visualized output format is one of the trusted and well acceptable methods. Thus a number of researchers tried to leverage the existing or newly developed graphical visualization methods for the opinion summary presentation. Some noteworthy related previous works on opinion summary visualization techniques are by Gamon et al., [11], Yi and Niblack, [12], Carenini et al. [14]¹ and [15].

Tracking: In many applications, analysts and other users are interested in tracking changes in sentiment about a product, political candidate, company or other issues over time. The tracking system could be a good measure to understand the people's sentiment changes or it could be helpful sociological survey also. In general sense tracking means plotting of sentiment values over time into a graphical visualization. Some significant research efforts on opinion tracking are Lydia² project (also called TextMap) [16], Ku et al., [17] Mishne and Rijke, [18] and Fukuhara et al., [19].

2 The Proposed 5W Rationalism

We mentioned a few (Due to space complexity) of noteworthy related works in this section. During the literature survey we realized that there is no consensus among the

¹ <http://www.cs.ubc.ca/~carenini/storage/SEA/demo.html>

² <http://www.textmap.com/>

researchers could be found on the output format of any sentiment summarization system.

Instead of digging for the answer of the unresolved debate we experimented with multiple output formats: multi-document topic-opinion textual summary but realizing the end user's requirement and to less their effort and to present an *at-a-glance* representation we devise a 5W constituent based textual summarization-visualization-tracking system. The 5W constituent based summarization system is a multi-genre system. The system facilitates users to generate sentiment tracking with textual summary and sentiment polarity wise graph based on any dimension or combination of dimensions as they want i.e. "Who" are the actors and "What" are their sentiment regarding any topic, changes in sentiment during "When" and "Where" and the reasons for change in sentiment as "Why". During the related work discussion we categorize the previous systems in "Topic-Wise", "Polarity-Wise" or "Other-Wise" genres. In the "Other-Wise" genre we described the necessity of the visualization and tracking systems. As per our understanding the 5W constituent based summarization system fall into every genre and the supportive argumentations from our side are as follows:

Topic-Wise: The 5W system facilitates users to generate sentiment summary based on any customized topic like Who, What, When, Where and Why and based on any dimension or combination of dimensions as they want.

Polarity-Wise: The system produces an overall gnat chart, could be treated as an overall polarity wise summary. An interested user can still look into the summary text to find out more details.

Visualization and Tracking: The visualization facilitates users to generate visual sentiment tracking with polarity wise graph based on any dimension or combination of dimensions as they want i.e. "Who" are the actors and "What" are their sentiment regarding any topic, changes in sentiment during "When" and "Where" and the reasons for change in sentiment as "Why". The final graph for tracking is been generated with a timeline.

There are very few research attempts where 5W structurization have been attempted. The ideas of 5Ws have been used successfully for a machine translation evaluation methodology [20]. The methodology addresses the cross-lingual 5W task: given a source language sentence and the corresponding target language sentence, it evaluates whether the 5Ws in the source have been comprehensibly translated into the target language. In addition we previously tried the 5W extraction task from Bengali [21].

From the next section we describe the development process of our 5W constituent based textual and visual summarization and tracking system.

3 Corpus Collections and Annotation

The present system has been developed for the Bengali language. Resource acquisition is one of the most challenging obstacles to work with resource-constrained languages like Bengali. Bengali is the fifth popular language³ in the World, second in India and the national language in Bangladesh.

³ http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

The details of corpus development could be found in [22] for Bengali. We obtained the corpus from the authors. For the present task a portion of the corpus from the editorial pages, i.e., Reader's opinion section or Letters to the Editor Section containing 28K word-forms have been manually annotated with sentence level opinion constituents. The detail statistics about the corpus is reported in Table 1.

Table 1. Bengali News Corpus Statistics

Statistics	NEWS
Total number of documents	100
Total number of sentences	2234
Average number of sentences in a document	22
Total number of wordforms	28807
Average number of wordforms in a document	288
Total number of distinct wordforms	17176

Annotators were asked to annotate 5Ws in Bengali sentences in terms of Bengali noun chunks. Instructions have been given to annotators to find out the principle opinionated verb in a sentence and successively extract 5W components by asking 5W questions to the principle verb.

Table 2. Agreement of annotators at each 5W level

Tag	Annotators X and Y Agree percentage
Who	88.45%
What	64.66%
When	76.45%
Where	75.23%
Why	56.23%

Table 3. Agreement of annotators at sentence level

Annotators	X vs. Y	X Vs. Z	Y Vs. Z	Avg.
Percentage	73.87%	69.06%	60.44%	67.8%
All Agree	58.66%			

The agreement of annotations between two annotators (Mr. X and Mr. Y) has been evaluated. The agreements of tag values at each 5W level are listed in Tables 2. For the evaluation of the extractive summarization system gold standard data has been prepared and three annotators took part. The inter-annotator agreement for the identification of subjective sentences for opinion summary is reported in Table 3.

It has been observed that in the present task the inter-annotator agreement is better for Who, When and Where level annotation rather than What and Why level though a small number of documents have been considered.

Further discussion with annotators reveals that the psychology of annotators is to grasp all 5Ws in every sentence, whereas in general all 5Ws are not present in every sentence. But the same groups of annotators are more cautious during sentence identification for summary as they are very conscious to find out the most concise set of sentences that best describe the opinionated snapshot of any document. The

annotators were working independent of each other and they were not trained linguists. As observed, the most ambiguous tag to identify is “Why”. The overall annotation has been done on 2234 sentences as mentioned in the Table 1. Generally each W type presents in a sentence only once but sometime it may twice. For example in the following sentence there are two “Who” tags. A post statistical analysis revealed that only in 3-5% cases each W tag repeats in a sentence and the percentage vary tag wise. Another important observation is every Ws are not present in every sentence. To better understand the distribution pattern of 5Ws in a corpus we gather a statistics for each 5W tag level as listed in Table 4

Table 4. Sentence wise co-occurrence pattern of 5Ws

Tags	Percentage						Total No. of Occurrence Of Each Ws in the Corpus
	Who	What	When	Where	Why	Overall	
Who	-	58.56%	73.34%	78.01%	28.33%	73.50%	1642
What	58.56%	-	62.89%	70.63%	64.91%	64.23%	1435
When	73.34%	62.89%	-	48.63%	23.66%	57.23%	1278
Where	78.0%	70.63%	48.63%	-	12.02%	68.65%	1533
Why	28.33%	64.91%	23.66%	12.02%	-	32.00%	714

The Gopal Krishna Gandhiiy/**Who**, expressed his grief for the rail accident and Smt. Mamata Banerjee/**Who** followed the same line of act to express her own feelings.

Sentiment tagging is always very ambiguous because it differs from the writer to reader’s perspective [23]. Therefore it is very hard to achieve high agreement score in sentiment data.

Another important observation is that 5W annotation task takes very little time for annotation. Annotation is a vital tedious task for any new experiment, but 5W annotation task is easy to adopt for any new language.

4 The 5W Extraction

The 5Ws semantic role labeling task demands and addressing various NLP issues such as: predicate identification, argument extraction, attachment disambiguation, location and time expression recognition. To solve these issues the present system architecture relies on Machine Learning technique and rule-based methodologies simultaneously.

One of the most important milestones in SRL literature is CoNLL-2005 Shared Task⁴ on Semantic Role Labeling. All most all SRL research group participated in the shared task. System reports of those participated systems eminently prove that

⁴ <http://www.lsi.upc.es/~srlconll/st05/st05.html>

Maximum Entropy⁵ (ME) based models work well in this problem domain as 8 among 19 systems used ME as the solution architecture. The second best performing system [24] uses ME model uses only syntactic information without using any pre or post processing. For the present system we did a number of experiments and finally choose an n -gram (where $n=4$) window with the best-identified features to train the classifier.

Table 4 presents the distribution pattern of 5Ws in overall corpus. It is very clear that 5Ws are not very regular jointly in the corpus. Hence sequence labeling with 5Ws tags using ME will lead a label biased problem (as we reported in Section 7) and may not be an acceptable solution for present problem definition as concluded in [24] (although in a different SRL task).

We apply both rule-based and statistical techniques jointly to the final system. The rules are being captured by acquired statistics on training set and linguistic analysis of standard Bengali grammar. The features used in the present system are reported in the following section.

4.1 The Feature Organization for MEMM

The features to be found most effective are chosen experimentally. Bengali is an electronically resource scarce language, thus our aim was to find out the less number of features but the features should be effective. Involving more number of features will demand more linguistic tools, which are not readily available for the language. All the features that have been used to develop the present system are categorized as Lexical, Morphological and Syntactic features. These are listed in the Table 5 below and have been described in the subsequent subsections.

Table 5. Features

Types	Features	
Lexical	POS	
	Root Word	
Morphological	Noun	Gender
		Number
		Person
		Case
	Verb	Voice
		Modality
Syntactic	Head Noun	
	Chunk Type	
	Dependency Relations	

Part of Speech (POS): POS of any word cannot be treated as direct clue of its semantic but it definitely helps to identify it. Finding out the POS of any word can reduce the search space for semantic meaning. It has been shown by [25], [26] etc.

⁵ <http://maxent.sourceforge.net/>

that the part of speech of any word in sentences is a vital clue to identify semantic role of that word.

Root Word: Root word is a good feature to identify word level semantic role especially for those types of 5Ws where dictionaries have been made like “When”, “Where” and “Why”. There are various conjuncts and postpositions, which directly indicate the type of predicate present in any sentence. As example জন্য, হেতু give clue that the next predicate is causative (“Why”).

Gender: Gender information is essential to relate any chunk to the principle verb modality. In the case of “What”/“Whom” ambiguities gender information help significantly. For inanimate objects it will be null and for animates it has definitely a value. Bengali is not a gender sensitive language hence this feature is not such significant linguistically rather number and person features. But the statistical co-occurrence of gender information with the number and person information is significant.

Number: Number information helps to identify specially for “Who”/“What” ambiguities. As we reported in inter-annotator agreement section “Who” has been identified first by matching modality information of principle verb with corresponding number information of noun chunks.

Person: Person information is as important as number information. It helps to relate any head of noun chunks to principle verb in any sentence.

Case: Case markers are generally described as karaka relations of any noun chunks with main verb. It has been described that semantically karaka is the ancestor of all semantic role interpretations. Case markers are categorized as Nominative, Accusative, Genitive and Locative. Case markers are very helpful for almost in every 5W semantic role identification task.

Voice: The distinction between active and passive verbs plays an important role in the connection between semantic role and grammatical function, since direct objects of active verbs often correspond in semantic role to subjects of passive verbs as suggested by various researchers [24]. A set of hand-written rules helps to identify the voice of any verb chunk. The rules rely on presence auxiliary verbs like হয়েছে, হোক etc indicate that the main verb in that particular chunk is in passive form.

Modality: Honorific markers are very distinctly used in Bengali and it directly reflects by the modality marker of any verb. As example the honorific variation করা/do are as কর (used with তুমি: 2nd person either of same age or younger), করো (used with তুমি: 2nd person either of same age or slightly elder) and করুন (used with আপনি: 2nd person generally for aged or honorable person). Verb Modality information helps to identify especially the “Who” tag. “Who” is identified first by matching modality information of principle verb with corresponding number information of noun chunks.

Head Noun: The present SRL system identifies chunk level semantic roles. Therefore morphological features of chunk head is only important rather other chunk members. Head words of noun phrases can be used to express selectional restrictions on the semantic role types of the noun chunks. For example, in a communication frame, noun phrases headed by Ram, brother, or he are more likely to be the SPEAKER

(Who), while those headed by proposal, story, or question are more likely to be the TOPIC (What).

Chunk Type: Present SRL system identifies noun chunk level semantic roles. Hence chunk level information is effectively used as a feature in supervised classifier and in rule-based post processor.

Dependency Relations: It has been profoundly established that dependency phrase-structures are most crucial to understand semantic contribution of every syntactic nodes in a sentence [25], [26]. A statistical dependency parser has been used for Bengali as described in [27]. Shallow parsers⁶ for Indian languages developed under a Government of India funded consortium project named Indian Language to Indian Language Machine Translation System (IL-ILMT) are now publicly available.

4.2 Rule-Based Post-processing

As described earlier post-processing is necessary in this setup. The rules developed here are either based on syntactic grammar, manually augmented dictionary or corpus heuristic.

In order to apply rule-based post-processor for “When” tag we developed a manually augmented list with pre defined categories as described in Table 6. Similar to “When”, we categorized “Where” and “Why” as general and relative as listed in Table 7.

Table 6. Time Expressions

	Bengali		English Gloss
	General	সকাল/সন্ধ্যা/রাত...	
টার সময়/ঘটিকায়/মিনিট		O clock/hour/minute	
সোমবার/মঙ্গলবার		Monday/Tuesday	
Relative	আগে/পরে...		Before/After...
	সামনে/পেছনে...		Upcoming/
	Special Cases	উঠলে/খামলে	When rise/When stop

Table 7. Locative Expressions

Type	Locative	
General	Bengali	English Gloss
	মার্চে/ঘাটে/রাস্তায়	Morning/evening/night/dawn
Relative	আগে/পরে...	Before/After...
	সামনে/পেছনে...	Front/Behind
Causative		
General	জন্য/কারণে/হেতু...	Hence/Reason/Reason
Relative	যদি_তবে	If_else
	যদিও_তবুও	If_else

⁶ http://lrc.iit.ac.in/showfile.php?filename=downloads/shallow_parser.php

5 Performance of the 5Ws Extraction

The performance result of ML (1) technique has been reported in Table 9. After using rule-based postprocessor the system (2) performance increases as listed in the following Table 8.

It is noticeable that the performance of the MEMM-based model differs tag-wise. For such heterogeneous problem nature we propose a hybrid system as rule-based post processor followed by Machine Learning. The rule-based post processor can identify those cases missed by ML method and can reduce false hits generated by statistical system.

Table 8. Performance of 5Ws Opinion Constituents by MEMM + Rule Based-Post Processing

Tag	Precision (%)		Recall (%)		F-measure (%)		Avg. F-Measure (%)	
	1	2	1	2	1	2	1	2
Who	76.2	79.6	64.3	72.6	69.8	75.9	62.2	68.1
What	61.2	65.5	51.3	59.6	55.9	62.4		
When	69.2	73.4	58.6	66.0	63.4	69.5		
Where	70.0	77.7	60.0	69.7	64.6	73.4		
Why	76.2	63.5	53.9	55.6	57.4	59.2		

6 The Summarization Methodologies

The present system is a multi-document extractive opinion summarization system for Bengali. Documents are preprocessed with the subjectivity identifier (as described in [28]) followed by the polarity classifier (as described in [29]). All the 5W constituents extracted from each sentence and clustered depending upon common constituents present at document level. The document clusters are then formed as tightly coupled network. The node of the network is the extracted sentiment constituent and the edges represent the relationship among them.

The next major step is to extract relevant sentences from each constituent cluster that reflects the contextual concise content of the current constituent cluster. Our summarization system is a dynamic one and the output depends on user's dimension choices. To adopt this kind of special need we used Information Retrieval (IR) based technique to identify the most "informed" sentences from the constituents cluster and it can be termed as IR based cluster center for that particular cluster. With the adaptation of ideas from page rank algorithms [30], it can be easily observed that a text fragment (sentence) in a document is relevant if it is highly related to many relevant text fragments of other documents in the same cluster. The basic idea is to cover all the constituents' node in the network by the shortest path algorithm as given by user. The adaptive page rank algorithm helps to find out the shortest distance, which covers all the desired constituents' node and maximizes the accumulated edge scores among them. Accordingly sentences are chosen based on the presence of those particular constituents. The detail description could be found in the following subsection.

6.1 Constituent Based Document Clustering

Constituent clustering algorithms (*K-Means*) partition a set of documents into finite number of groups or clusters in terms of 5W opinion constituents. Documents are represented as a vector of 5W constituents present in the opinionated sentences within the document into various subjective sentences.

The similarity between vectors is calculated by assigning numerical weights to 5W opinion constituents and then using the cosine similarity measure as specified in the following equation.

$$s(\vec{d}_k, \vec{d}_j) = \vec{d}_k \cdot \vec{d}_j = \sum_{i=0}^N w_{i,k} \times w_{i,j}$$

where \vec{d}_k and \vec{d}_j are the document vectors. N is the total number of unique 5Ws that exist in the document set \vec{d}_k and \vec{d}_j . The $w_{i,k}$ and $w_{i,j}$ are the 5W opinion constituents that exist in the documents \vec{d}_k and \vec{d}_j respectively. An example of inter-document theme cluster has been reported in Table 9. The numeric scores are the similarity association value assigned by the clustering technique. A threshold value of greater than 0.5 has been chosen experimentally to construct the inter-document theme relational graph in the next level.

Table 9. Theme Clusters by 5W Dimensions

Generated Clusters						
5Ws	Constituents	Doc1	Doc2	Doc3	Doc4	Doc5
Who	Mamata Banerjee	0.63	0.01	0.55	0.93	0.02
	West Bengal CM	0.00	0.12	0.37	0.10	0.17
What	Gyaneswari Express	0.98	0.79	0.58	0.47	0.36
	Derailment	0.98	0.76	0.35	0.23	0.15
When	24/05/2010	0.94	0.01	0.01	0.01	0.01
	Midnight	0.68	0.78	0.01	0.01	0.01
Where	Jhargram	0.76	0.25	0.01	0.13	0.76
	Khemasoli	0.87	0.01	0.01	0.01	0.01
Why	Maoist	0.78	0.89	0.06	0.10	0.14
	Bomb Blast	0.13	0.78	0.01	0.01	0.78

To better aid our understanding of the automatically determined category relationships we visualized this network using the Fruchterman-Reingold force directed graph layout algorithm [31] and the NodeXL network analysis tool [32]⁷ as shown in Fig. 1. In the following graphical representation one color depict one cluster.

⁷ Available from <http://www.codeplex.com/NodeXL>

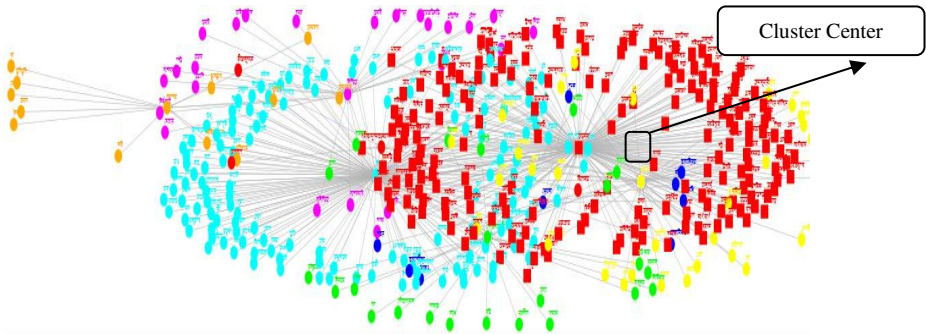


Fig. 1. Document Level Theme Relational Graph by NodeXL

6.2 Constituent Relevance Calculation

In the generated constituent network all the lexicons are connected with weighted vertex either directly or indirectly. Semantic lexicon inference could be identified by network distance of any two constituent nodes by calculating the distance in terms of weighted vertex. We computed the relevance of semantic lexicon nodes by summing up the edge scores of those edges connecting the node with other nodes in the same cluster. As cluster centers are also interconnected with weighted vertex so inter-cluster relations could be also calculated in terms of weighted network distance between two nodes within two separate clusters. As an example: suppose we have the following two clusters: *A* and *B*. *A* has *m* numbers of nodes while *B* consists of *n* numbers of nodes. a_x and b_y are the clusters centers of *A* and *B*.

$$A = \{a_1, a_2, a_3, a_4, \dots, a_x, \dots, a_m\} \quad B = \{b_1, b_2, b_3, b_4, \dots, b_y, \dots, b_n\}$$

The lexicon semantic affinity inference between a_x and b_y could be calculated as follows:

$$S_d(a_x, b_y) = \frac{\sum_{k=0}^n v_k}{k} \quad \text{---(3)} \quad \text{or} \quad \sum_{c=0}^m \frac{\sum_{k=0}^n v_k}{k} \times \prod_{c=0}^m l_c \quad \text{---(4)}$$

where $S_d(a_x, b_y)$ = semantic affinity distance between two constituent a_x and b_y .

Equation (3) and (4) are for intra-cluster and inter-cluster semantic distance measure respectively. k =number of weighted vertex between two constituent a_x and b_y . v_k is the weighted vertex between two lexicons. m =number of cluster centers between two lexicons. l_c is the distance between cluster centers between two lexicons.

6.3 Dimension Wise Opinion Summary-Visualization-Tracking

The working principle of the present system is as follows.

- The system identifies all the desired nodes in the developed semantic constituent network as given by user in the form of 5W.

- Inter-constituents distances have been calculated from the developed semantic constituent network. For example, suppose user gave the following input. Therefore the calculated inter-constituents distances may look like the Table 10.\

Input:	<u>Who</u> মমতা ব্যানার্জী (Mamata Banerjee)	<u>What</u> জানেশ্বরী এক্সপ্রেস (Gyaneshwari Express)	<u>When</u> মধ্যরাত (Midnight)	<u>Where</u> ঝাড়গ্রাম (Jhargram)	<u>Why</u> মাওবাদী (Maoist)
---------------	---	--	---	--	--

Table 10. Calculated inter-constituents distances

Type	Inter-Constituents Distances				
	Who	What	When	Where	Why
Who	-	0.86	0.02	0.34	0.74
What	0.86	-	0.80	0.89	0.67
When	0.02	0.80	-	0.58	0.23
Where	0.34	0.89	0.58	-	0.20
Why	0.74	0.67	0.23	0.20	-

- All the sentences consist of at least one of the user-defined constituents are extracted from all the documents.
- Extracted sentences are then ranked with the adaptive Page-Rank algorithm based on the constituent present in that sentence. In the first iteration the standard IR based Page-Rank algorithm assign a score to each sentence based on keyword (constituents are treated as keyword in this stage) presence. In the second iteration the calculated rank by the Page-Rank algorithm are multiplied with the inter-constituents distances for those sentences where more than one constituent present. For example: in the next sentence two Ws: “Who” and “What” are present jointly as constituent. Suppose the assigned rank for the following sentence by the basic Page-Rank algorithm is n . Then in the next iteration the modified score will be $n*0.86$, because the inter-constituents distances for “Who” (মমতা বন্দ্যোপাধ্যায়) and “What” (জানেশ্বরী এক্সপ্রেস) is 0.86.

মমতা_বন্দ্যোপাধ্যায়/**Who** জানেশ্বরী_এক্সপ্রেস_ঘটনাকে/**What** রাজনৈতিক চক্রান্ত বলে মন্তব্য করেন।

English **Gloss:** Mamta_Bandyopadhyay/**Who** commented that the Gyaneshwari_Express_incident/**What** is a political conspiracy.

- The ranked sentences are then sorted by descending order and top-ranked 30% sentences (from all retrieved sentences) are shown as a summary.

Ordering of sentences is very important is very important in case of summarization. We prefer the temporal order of sentences as they occurred in original document, when it published.

The visual tracking system consists of five drop down boxes. The drop down boxes give options for individual 5W dimension of each unique Ws that exist in the corpus. The present visual tracking system facilitates users to generate opinion polarity wise graph based visualization and summary on any 5W dimension and combination of 5W dimensions, as they want. (Shown in Fig. 2).

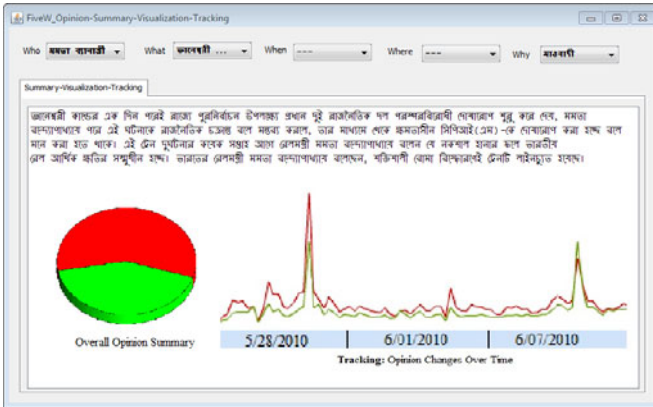


Fig. 2. A Snapshot of the Present Summarization System

7 Experimental Result

To evaluate the present system we follow a two-fold evaluation mechanism. The first-fold evaluation is to understand the system performance to detect relative sentences prior to generate final summary (as mentioned in the third step of the Summary process). For this evaluation we check system-identified sentences with every human annotator’s gold standard sentences and finally we calculated the overall accuracy of the system as reported in Table 11.

Table 11. Final Results subjective sentence identification for summary

Metrics	X	Y	Z	Avg.
Precision	77.65%	67.22%	71.57%	72.15%
Recall	68.76%	64.53%	68.68%	67.32%
F-Score	72.94%	65.85%	70.10%	69.65%

Table 12. Human Evaluation on 5W Dimension Specific Summaries

Tags	Average Scores				
	Who	What	When	Where	Why
Who	-	3.20	3.30	3.30	2.50
What	3.20	-	3.33	3.80	2.6
When	3.30	3.33	-	2.0	2.5
Where	3.30	3.80	2.0	-	2.0
Why	2.50	2.6	2.5	2.0	-
Overall	3.08	3.23	3.00	2.77	2.40

It was a challenge to evaluate the accuracy of the dimension specific summaries. It is hardly possible to make a human extracted gold summary set for every dimension combinations; therefore we propose a direct human evaluation technique. Two evaluators have been involved in the present task and they are asked to give evaluative score to each system-generated summaries. We use a 1-5 scoring technique whereas 1 denotes very poor, 2 denotes poor, 3 denotes acceptable, 4 denotes good and 5 denotes excellent. The final evaluation result of the dimension specific summarization system is reported in Table 12.

8 Conclusion

The present paper started with a very basic question “*What is the End User’s Requirement?*”. To answer this question we do believe that our proposed 5W Summarization –Visualization-Tracking system could be treated as a qualitative and acceptable solution. To compare our suggestion we presented a vivid description of the previous works. Another self-contributory remark should be mentioned that according to best of our knowledge this is the first attempt on opinion summarization or visual tracking for the language Bengali. Moreover the 5W structurization is new to the community and proposed by us.

Acknowledgments. The work reported in this paper is supported by a grant from the India-Japan Cooperative Programme (DST-JST) 2009 Research project entitled “Sentiment Analysis where AI meets Psychology” funded by Department of Science and Technology (DST), Government of India.

References

1. Dasgupta, S., Ng, V.: Topic-wise, Sentiment wise, or Otherwise? Identifying the Hidden Dimension for Unsupervised Text Classification. In: EMNLP 2009 (2009)
2. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* (2008)
3. Seki, Y., Eguchi, K., Kando, N.: Analysis of multi-document viewpoint summarization using multi-dimensional genres, pp. 142–145. *AAAI* (2004)
4. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: *Proceedings of ACL* (2004)
5. Ku, L.-W., Li, L.-Y., Wu, T.-H., Chen, H.-H.: Major topic detection and its application to opinion summarization. In: *Proceedings of the SIGIR*, pp. 627–628 (2005)
6. Liang, Z., Eduard, H.: On the summarization of dynamically introduced information: Online discussions and blogs. In: *AAAI-CAAW*, pp. 237–242 (2006)
7. Kawai, Y., Kumamoto, T., Tanaka, K.: Fair News Reader: Recommending News Articles with Different Sentiments Based on User Preference. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) *KES 2007, Part I. LNCS (LNAI)*, vol. 4692, pp. 612–622. Springer, Heidelberg (2007)

8. Hu, M., Liu, B.: Mining and summarizing-customer reviews. In: Proc. of the 10th ACM-SIGKDD Conf., pp. 168–177. ACM Press, New York (2004)
9. Zhuang, L., Jing, F., Zhu, X., Zhang, L.: Movie review mining and summarization. In: ACM-SIGIR-(CIKM) (2006)
10. Das, S.R., Chen, M.Y.: Yahoo! for Amazon: Sentiment extraction from small talk on the Web. *Management Science* 53(9), 1375–1388 (2007)
11. Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining Customer Opinions from Free Text. In: Famili, A.F., Kok, J.N., Peña, J.M., Siebes, A., Feelders, A. (eds.) IDA 2005. LNCS, vol. 3646, pp. 121–132. Springer, Heidelberg (2005)
12. Yi, J., Niblack, W.: Sentiment mining in WebFountain. In: Proceedings of the International Conference on Data Engineering, ICDE (2005)
13. Gruhl, D., Chavet, L., Gibson, D., Meyer, J., Pattanayak, P., Tomkins, A., Zien, J.: How to build a Webfountain: architecture for very large-scale text analytics. *IBM Systems Journal* 43(1), 64–77 (2004)
14. Carenini, G., Ng, R.T., Pauls, A.: Interactive multimedia summaries of evaluative text. In: Proceedings of Intelligent User Interfaces (IUI), pp. 124–131. ACM Press (2006)
15. Gregory, M.L., Chinchor, N., Whitney, P., Carter, R., Hetzler, E., Turner, A.: User-directed sentiment analysis: Visualizing the affective content of documents. In: Proceedings of the Workshop on Sentiment and Subjectivity in Text, pp. 23–30. ACL (2006)
16. Lloyd, L., Kechagias, D., Skiena, S.S.: Lydia: A System for Large-Scale News Analysis. In: Consens, M.P., Navarro, G. (eds.) SPIRE 2005. LNCS, vol. 3772, pp. 161–166. Springer, Heidelberg (2005)
17. Ku, L.-W., Liang, Y.-T., Chen, H.-H.: Opinion extraction, summarization and tracking in news and blog corpora. In: AAAI-CAAW, pp. 100–107 (2006)
18. Mishne, G., de Rijke, M.: Moodviews: Tools for blog mood analysis. In: AAAI-CAAW, pp. 153–154 (2006)
19. Fukuhara, T., Nakagawa, H., Nishida, T.: Understanding sentiment of people from news articles: Temporal sentiment analysis of social events. In: ICWSM (2007)
20. Parton, K., McKeown, K., Coyne, B., Diab, M., Grishman, R., Hakkani-Tür, D., Harper, M., Ji, H., Wei, Y.M., Meyers, A., Stolbach, S., Sun, A., Tur, G., Wei, X., Sibel, Y.: Who, What, When, Where, Why? Comparing Multiple Approaches to the Cross-Lingual 5W Task. In: The Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 423–431 (2009)
21. Das, A., Ghosh, A., Bandyopadhyay, S.: Semantic Role Labeling for Bengali Noun using 5Ws: Who, What, When, Where and Why. In: The Proceeding of the International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLPKE2010), Beijing, China, pp. 1–8 (2010)
22. Ekbal, A., Bandyopadhyay, S.: A Web-based Bengali News Corpus for Named Entity Recognition. *LRE Journal* 42(2), 173–182 (2008)
23. Tang, Y.-J., Chen, H.-H.: Emotion Modeling from Writer/Reader Perspectives Using a Microblog Dataset. In: Proceeding of the Workshop Sentiment Analysis Where AI Meets Psychology (2011)
24. Haghghi, A., Toutanova, K., Manning, C.D.: A Joint Model for Semantic Role Labeling. In: CoNLL-2005 Shared Task (2005)
25. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. In: Association for Computational Linguistics (2002)

26. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal* 31(1) (2005)
27. Ghosh, A., Das, A., Bhaskar, P., Bandyopadhyay, S.: Dependency Parser for Bengali: the JU System at ICON 2009. In: *NLP Tool Contest ICON 2009* (2009)
28. Das, A., Bandyopadhyay, S.: Subjectivity Detection using Genetic Algorithm. In: *WASSA 2010*, Lisbon, Portugal, August 16-20 (2010)
29. Das, A., Bandyopadhyay, S.: Phrase-level Polarity Identification for Bengali. *IJCLA* 1(1-2), 169–182 (2010) ISSN 0976-0962s
30. Page, L.: PageRank: Bringing Order to the Web. *Stanford Digital Library Project* (1997)
31. Fruchterman Thomas, M.J., Reingold Edward, M.: Graph drawing by force-directed placement. *Software: Practice and Experience* 21(11), 1129–1164 (1991)
32. Marc, S., Shneiderman, B., Milic-Frayling, N., Rodrigues, E.M., Barash, V., Dunne, C., Capone, T., Perer, A., Gleave, E.: Analyzing (social media) networks with NodeXL. In: *C&T 2009: Proc. Fourth International Conference on Communities and Technologies* (2009)

The Naive Bayes Classifier in Opinion Mining: In Search of the Best Feature Set

Liviu P. Dinu and Iulia Iuga

University of Bucharest, Faculty of Mathematics and Computer Science,
Center for Computational Linguistics,
14 Academiei, RO-010014, Bucharest, Romania
ldinu@fmi.unibuc.ro, iuliaiuga1211@gmail.com

Abstract. This paper focuses on how naive Bayes classifiers work in opinion mining applications. The first question asked is what are the feature sets to choose when training such a classifier in order to obtain the best results in the classification of objects (in this case, texts). The second question is whether combining the results of Naive Bayes classifiers trained on different feature sets has a positive effect on the final results. Two data bases consisting of negative and positive movie reviews were used when training and testing the classifiers for testing purposes.

1 Introduction

During the last decade, data text mining [15] has received a lot of attention, due to the explosion of available data (over 80% of information is stored as text). Typical text mining tasks include text categorization and text clustering [4], humor characterization [9], coherence texts investigation [3], or opinion mining and sentiment analysis [12], [8].

This paper focuses on how naive Bayes classifiers work in opinion mining field, which has had a boost as the on-line social media which has had a boost as the on-line social media (blogs [2], social networks [10], etc.) has risen and the interest in quickly determining the general opinions on certain topics has increased.

Given a set of subjective texts that express opinions about a certain object, the purpose is to extract those attributes (features) of the object that have been commented on in the given texts and to determine whether these texts are positive, negative or neutral.

A couple of interesting applications are sentiment analysis tools for Twitter status updates (<http://www.tweetfeel.com/> , <http://twittersentiment.appspot.com/>) are relevant examples, but not the only ones) or the analysis of short comments on film reviews [16], [11].

1.1 Preliminaries

In the "bag-of-words" model [7], we begin with making the simplifying assumption about a text that it can be represented as collections of words in which grammar rules are negligible and even the word order is unimportant.

Bayes classifiers [6] assign the most likely class to a given example described by its feature vector. Training such classifiers can be significantly simplified by assuming that features are independent classes, that is:

$$P(X|C) = \prod_{i=1, \overline{n}} (P(X_i|C)) \quad (1)$$

where $X = (X_1, X_2, \dots, X_n)$ is a feature vector and C is a class. Despite the unrealistic assumption that the features are independent from each other, the resulting classifier, called naive Bayes classifier, is very successful in practice and this technique has the great advantages of being much faster than more sophisticated ones and much more approachable, besides generating competitive results. The naive Bayes has proven effective and is often used in many practical applications besides the text classification that we are going to discuss further, such as medical diagnosis [5] and system performance management [14]. In order to determine the probability of a word to appear in a positive versus a negative context, we will first train a classifier, in this case a naive Bayes classifier, on a set of annotated features (here the labels will be "positive" and "negative").

In the field of statistics, the accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's actual (true) value. The precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved.

1.2 Motivation and Personal Contributions

The purpose of our study was determining what is the best way of selecting features from texts that are to be analysed in opinion mining applications that use the naive Bayes classifier as a decision taker. We were interested in performance from the point of view of accuracy as well as of the time efficiency, which is important in practical applications.

We first ran a series of tests on features from the input texts that we considered relevant in the classification of the texts from the opinion mining point of view and then we proposed two combining techniques for individual classifiers trained on different feature sets.

In the closing of this paper we will briefly present an application that we have developed, application that implements an algorithm for determining the percentage of positive, negative and neutral NewsGroup reviews and user comments for films listed on the Internet Movie Database, disregarding their star-scores and using only on naive Bayes classification as the analyses tool.

2 Tests

In this section we will present the results obtained when training and testing naive Bayes classifiers on ten different feature sets. Each feature set will be discussed in the following.

The data set used for training the naive Bayes classifiers was *Polarity Dataset v2.0* [12]. It consists of 1000 positive movie reviews and 1000 negative ones (we can assume that the classifiers receives equal numbers of positive and negative features when trained). This corpus is included in NLTK (Natural Language ToolKit, a tool that we used when programming these tests) under the name *movie_reviews*.

For testing data we used *Polarity Dataset v1.0* [12] which consist from 700 positive movie reviews and 700 negative ones.

Both these data bases can be found and downloaded at the following link:

<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

2.1 Results Obtained for Testing the Classification Given by the Naive Bayes Classifier When Different Features Were Taken into Consideration

The features which were considered for testing the classification given by the naive Bayes classifier are listed below. The results of all the tests can be read in the table found after the enumeration of the feature sets:

1. Test no. 1: The first test was run when considering all the words.

For this test, the most informative features were:

avoids = True	pos : neg = 13.0 : 1.0
astounding = True	pos : neg = 12.3 : 1.0
slip = True	pos : neg = 11.7 : 1.0
outstanding = True	pos : neg = 11.5 : 1.0
ludicrous = True	neg : pos = 11.0 : 1.0
fascination = True	pos : neg = 11.0 : 1.0
3000 = True	neg : pos = 11.0 : 1.0
insulting = True	neg : pos = 11.0 : 1.0
sucks = True	neg : pos = 10.6 : 1.0
hudson = True	neg : pos = 10.3 : 1.0

2. Test no. 2: For the second test, we eliminated the stopwords from the texts, hoping that they don't weight much in the subjectivity department. It turned out it made no difference if we filter or not tyhe stopwords.

Same as before, the most informative features were:

avoids = True	pos : neg = 13.0 : 1.0
astounding = True	pos : neg = 12.3 : 1.0
slip = True	pos : neg = 11.7 : 1.0
outstanding = True	pos : neg = 11.5 : 1.0
ludicrous = True	neg : pos = 11.0 : 1.0
fascination = True	pos : neg = 11.0 : 1.0
3000 = True	neg : pos = 11.0 : 1.0
insulting = True	neg : pos = 11.0 : 1.0
sucks = True	neg : pos = 10.6 : 1.0
hudson = True	neg : pos = 10.3 : 1.0

3. Test no. 3: For this test, we applied a stemmer to the words, trying to find out if maybe only the roots of the words of the texts would be sufficient to obtain the information we are looking for. According to this test, different forms of the words are relevant when expressing opinions.

For this test, the most informative features were:

plod = True	neg : pos = 13.7 : 1.0
misfir = True	neg : pos = 11.7 : 1.0
outstand = True	neg : pos = 11.5 : 1.0
incoher = True	neg : pos = 11.0 : 1.0
3000 = True	neg : pos = 11.0 : 1.0
predat = True	neg : pos = 10.3 : 1.0
hudson = True	neg : pos = 10.3 : 1.0
seamless = True	pos : neg = 10.3 : 1.0
hatr = True	pos : neg = 10.3 : 1.0
ideolog = True	pos : neg = 10.3 : 1.0

4. Test no. 4: For the next test, we took into consideration the bigrams (pairs of words) from the texts, plus all the words. It appears that collocations of words from the text help determine polarity.

The most informative features were:

('give', 'us') = True	pos : neg = 14.3 : 1.0
avoids = True	pos : neg = 13.0 : 1.0
('quite', 'frankly') = True	pos : neg = 12.3 : 1.0
astounding = True	pos : neg = 12.3 : 1.0
('does', 'so') = True	neg : pos = 12.3 : 1.0
slip = True	pos : neg = 11.7 : 1.0
('&', 'robin') = True	neg : pos = 11.7 : 1.0
('fairy', 'tale') = True	neg : pos = 11.7 : 1.0
outstanding = True	neg : pos = 11.5 : 1.0
ludicrous = True	neg : pos = 11.0 : 1.0

5. Test no. 5: For this test we considered as the feature set for training and testing the most frequent 10.000 words. The words that have the highest frequencies are the most relevant, but still, there is room for improvement and, according to the previous test, it appears that the collocations provide slightly more information. That leads to the next idea: to combine these two, that is to train and test a classifier on all the bigrams from the texts and most frequent 10000 words too.

For this test, the most informative features were:

avoids = True	pos : neg = 13.0 : 1.0
astounding = True	pos : neg = 12.3 : 1.0
slip = True	pos : neg = 11.7 : 1.0
outstanding = True	pos : neg = 11.5 : 1.0
ludicrous = True	neg : pos = 11.0 : 1.0
fascination = True	pos : neg = 11.0 : 1.0
3000 = True	neg : pos = 11.0 : 1.0
insulting = True	neg : pos = 11.0 : 1.0
sucks = True	neg : pos = 10.6 : 1.0
thematic = True	neg : pos = 10.3 : 1.0

6. Test no. 6: For all bigrams and most frequent words. When using the best words and the bigrams as feature set for training and testing, there is no improvement comparing to the test ran just on bigrams.

The most informative features were:

('give', 'us') = True	pos : neg = 14.3 : 1.0
avoids = True	pos : neg = 13.0 : 1.0
('quite', 'frankly') = True	pos : neg = 12.3 : 1.0
astounding = True	pos : neg = 12.3 : 1.0
('does', 'so') = True	neg : pos = 12.3 : 1.0
slip = True	pos : neg = 11.7 : 1.0
(&, 'robin') = True	neg : pos = 11.7 : 1.0
('fairy', 'tale') = True	neg : pos = 11.7 : 1.0
outstanding = True	neg : pos = 11.5 : 1.0
ludicrous = True	neg : pos = 11.0 : 1.0

7. Test no. 7: A different approach to take that came to mind was using as features those parts of speech from the text that seem to express the most subjectivity, that being the adjectives and the adverbs. Test number 7 was done on adjectives only.

In order to extract the adjectives from the text we used the WordNet thesaurus (also included as a package in a the Natural Language ToolKit) and extracted the words of the movie reviews that appeared at least once in WN as adjectives. We did not use a part of speech tagger, but that is a technique that is worth being investigated. The same tactic was used in the next test, for extracting the adverbs.

8. Test no. 8: This test was done on both the adjectives and the adverbs from the texts.
9. Test no. 9: Going in this direction, another idea came to mind, that being that we might benefit from adding to the adjectives, extracted from the texts in the same manner as presented before, their WordNet synonyms.

We can notice that this has not improved our results, but the contrary and the reason that happened could be that we did not determine the meaning of those adjectives in their contexts and therefore we added to the training feature sets all the possible synonyms of those words, disregarding their actual meaning in the context. An interesting direction to go from this point

NaiveBayes	Accuracy	Neg precision	Neg recall	Pos precision	Pos recall
Test 1	86.07	97.37	74.14	79.12	98.00
Test 2	86.07	97.37	74.14	79.12	98.00
Test 3	83.79	97.02	69.71	76.37	97.86
Test 4	91.00	95.13	86.43	87.57	95.57
Test 5	89.71	92.64	86.29	87.17	93.14
Test 6	91.00	95.13	86.43	87.57	95.57
Test 7	81.5	96.03	65.71	73.94	97.29
Test 8	82.36	95.94	67.57	74.97	97.14
Test 9	70.43	94.97	43.14	63.22	97.71
Test 10	53.50	54.56	41.86	52.84	65.14

Fig. 1. Precision of feature sets

would be using a disambiguation algorithm for establishing the meaning of the adjectives in their contexts and only adding to the training feature sets the synonyms of those meanings that we obtained. After that, applying the classifier. This might lead to better results.

10. Test no. 10: Another question raised was how much information is provided by parts of speech in the text studied. Meaning, how much could one find out about the subjectivity of a text when only looking at the parts of speech (nouns, adjectives, adverbs and verbs) that the texts consist of. So, for this test, the feature sets were the parts of speech of the reviews:

As it turns out, this test is not very relevant for our purposes and the results are just as good as flipping a coin. However, another idea to be pursued is whether the way parts of speech are enchainned is relevant in this type of classification. Maybe it is more likely to have a certain taxis in a positive text then in a negative one.

For every features we computed the accuracy, negation precision, negation recall, positive precision and positive recall. The Results obtained for testing the classification given by the naive Bayes classifier when the previous 10 features were taken into consideration are summarized in Figure [11](#)

Example 1. We show in Figure [12](#) examples of classification of the documents on which the testing was done into the "positive" and "negative" categories.

	NEG : POS		NEG : POS
cv000_tok-11609.txt	0.0 : 1.0	cv000_tok-9611.txt	1.0 : 0.0
cv001_tok-10180.txt	0.0 : 1.0	cv003_tok-13044.txt	0.9916 : 0.0084
cv001_tok-19324.txt	0.0 : 1.0	cv005_tok-24602.txt	1.0 : 0.0
cv002_tok-12931.txt	0.0 : 1.0	cv006_tok-29539.txt	1.0 : 0.0
cv002_tok-3321.txt	0.0229 : 0.9771	cv009_tok-19587.txt	1.0 : 0.0
cv003_tok-8338.txt	0.0 : 1.0	cv011_tok-7845.txt	1.0 : 0.0
cv004_tok-25944.txt	0.0 : 1.0	cv012_tok-26965.txt	1.0 : 0.0
cv004_tok-29856.txt	0.0 : 1.0	cv013_tok-14854.txt	1.0 : 0.0
cv005_tok-26110.txt	0.0 : 1.0	cv014_tok-12391.txt	0.9997 : 0.0003
cv006_tok-28887.txt	0.0 : 1.0	cv016_tok-16970.txt	0.9628 : 0.0372

Fig. 2. Example of classification

In the left side you can see the list of the documents classified by naive Bayes as positive: for example, the document *cv002_tok-12931.txt* has a probability of being negative of 0.0229 and a positive probability of 0.9771, therefore is included in the positive documents list.

Remark 1. We split the texts in thirds and ran the same tests described before on these parts. The accuracy decreased slightly for each of the thirds. This indicates that there isn't a rule about having more information about sentiment polarity in the beginning as opposed to the end or the middle.

2.2 Combining Classifiers

As discussed in the previous subsection, for each feature set, we have built a naive Bayes classifier that we trained on the first data set and test on the second one. The results for each of the feature sets listed before can be read in the Table 2.1 from the previous subsection. In the following we will provide two combining classifiers methods which we applied on the previous features.

Each classifier calculates a certain probability for the documents to be positive or negative ones. If the probability of a text to be positive is bigger then the one of it being negative, then the classifier assigns it the positive class and the other way around. We will therefore have a resulting list that looks like this: $\{(neg > pos), (pos > neg), \dots\}$.

The Majority Rule. For each document, we will assign the class that appears a majority of times in the list generated by the classifiers.

Probability Aggregation. We calculate the sum of the positive/negative probabilities given by each classifier. Then, if the sum of the positive probabilities is bigger then the negative one, we will assign that document the positive class and vice versa.

Table 1. The majority rule

Method 1	Accuracy	Neg precision	Neg recall	Pos precision	Pos recall
c1 - c5	86.57	97.23	75.29	79.84	97.86
c1 - c7	87.79	97.32	77.71	81.45	97.86
c1 - c9	86.79	97.42	75.57	80.05	98.00

Table 2. Probability aggregation

Method 2	Accuracy	Neg precision	Neg recall	Pos precision	Pos recall
c1 - c5	87.29	97.11	76.86	80.85	97.71
c1 - c7	87.79	97.32	77.71	81.45	97.86
c1 - c9	87.14	97.27	76.43	80.59	97.86

As seen in the result tables [1](#) and [2](#) figured before, we did not find a combining method that increases the performance for combined classifiers as opposed to the individual ones. An idea would be to take advantage of the difference in the recall and precision measurements obtained on different feature sets. That is, suppose we have a number of classifiers trained on a certain feature set for which it generates big positive precision values and another series of classifiers that have a big positive recall (and the other way around for the negative values). By combining them, they might balance each other out and conduct to better results. In our case, we did not have good examples of independent feature sets to implement this idea, but we consider it worth investigating.

3 An Application: Opinion Mining for IMDb User Comments and NewsGroup Reviews

The conclusion drawn after running the tests in section 2 of this paper is that the best results are obtained for the feature set consisting of bigram collocations found in the texts. However, there is not such big of a difference in performance between this method and the one using the most frequent words feature sets – in this particular case, we had a difference of approximately 1%. But there is a significant difference when thinking about the time consumption aspect, the later being a much faster method (runs 2-3 times faster than the former). This aspect is very important in practical applications that are meant for the use of non-specialists. That is the reason why we chose extracting the most frequent words from the texts analyzed. This application is basically an in-browser application consisting of two buttons placed in the toolbar of the browser, buttons that, when pressed, if the user is visiting the page corresponding to a movie or TV series listed on the imdb.com website, will calculate the percentage of positive, negative and neutral user comments, respectively reviews posted in the NewsGroup. (When the user is on a different page, they will receive error messages; also, if there aren't any comments/reviews, a message will be displayed that will let them know about this, when the buttons are pressed.) This categorization

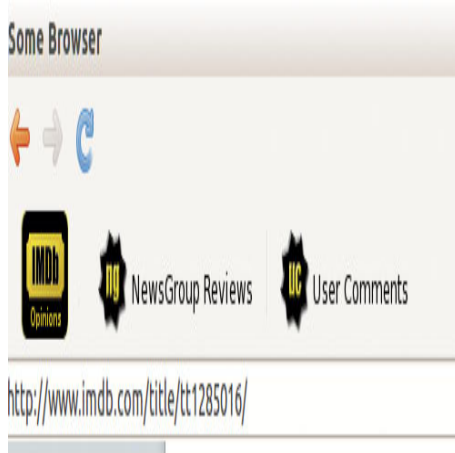


Fig. 3. The toolbar of a browser



Fig. 4. The dialog boxes

will be made exclusively based on the algorithm discussed previously and not taking into consideration the scores the commentators might have given for that particular title. The algorithm gives scores for the comments tested for opinions that give the positive/negative probabilities; so, if this probability of a certain review to be positive is somewhere between 0.45 and 0.55, the review will be considered to be neutral (same for negative probability).

We show three print-screens, the first one (Figure 3) shows the toolbar of a browser, where the two buttons are to be found, the second and third one (Figure 4) show the dialog boxes displayed after pressing the two buttons when the user is visiting the page of a particular title listed on the IMDb website.

In order to minimize the time spent on calculations, we will start with an already trained classifier. For this training, we used the same data base used in

the training step from the tests presented in section 2 (1000 positive and 1000 negative movie reviews).

Remark 2. When the number of user comments assigned to a movie increases, the scores percentage of positive/negative result generated with this algorithm is very close to the the star-score of that title on IMDb, which makes sense (bare in mind that any logged user is allowed to rate a title on IMDb, they don't have to necessarily leave a comment in order to do so). Not the same thing happens for the reviews listed in the NewsGroup; this is not so bizarre, as these reviews are generally posted by critics, who might not share the opinion of the general public.

By developing this application, we were able to observe how the classification with naive Bayes works in real case scenarios. The fact that, when having a large number of user comments, the results that the algorithm gave were increasingly closer to the star rating is proof enough that this is a good path and is worth investigating and improving in the future.

4 Conclusions

Nowadays, due to the explosion of the internet, we deal with an unprecedented amount of data published by people all over the world who express their opinions on different topics. In most cases, when in need to access and determine general opinions on certain topics, we do not have a rating system available (such as the one provided by the IMDb) and developing opinion mining methods that are fast and as efficient as possible is a current issue.

This study was focused on developing such a method that uses a time-efficient classification algorithm. We decided to use the naive Bayes classifier. After performing a series of tests to determine its performance when running on different feature sets extracted from the analysed texts, we came to the conclusion that the best option for selecting the feature set for real time applications is extracting a relevant number of the most frequent words from the texts used for training the classifier. For such a simple and apparently overly-simplifying technique, it performs very well and extremely fast.

As we were trying to find those feature sets that make the most sense to be relevant in opinion mining, we made a series of assumptions, such as that groups of words might give more information (correct), that the most frequent words weight more (also correct), but also found new leads that we think are worth further investigating; some parts of speech provide more information than others (adjectives, adverbs); this seems like a sensible assumption, but there is information lost when extracting the adjectives from the texts (we used the WordNet thesaurus to determine if each word has at least one adjective/adverb meaning; a better solution might be using a part of speech tagger). Then, we tried to add to the training feature set the synonyms of the adjectives selected that way; this method also lead to decreasing the performance, a reason for this might be that we did not use a disambiguation technique to determine the sense

of the words in their contexts before adding the synonyms, but we added all synonyms of the words instead. That would be the second most important idea to investigate in a future work.

Of course, this method, even if will give good accuracy for classification, will be extremely slow: applying a part of speech tagger and a disambiguation algorithm are both time consuming. And in the end it might turn out not have even been worth the effort, not gaining those many percent points in final results of classification.

Secondly, we tried to find a combining method for classifiers that had been trained on different feature sets in order to increase the final accuracy level. We did not manage to find such a method, but a disadvantage that we had was not having a balanced set of classifiers, that is, all the classifiers were wrong in the same way: similar values for precision and recall values. We think that by combining classifiers that even each other out, we would have a much better chance with the combining methods we proposed.

Acknowledgments. Research supported by the CNCS, IDEI - PCE project 311/2011, "The Structure and Interpretation of the Romanian Nominal Phrase in Discourse Representation Theory: the Determiners."

References

1. Chaovalit, P., Zhou, L.: Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. In: 38th Hawaii International Conference on System Sciences, HICSS 2005 (2005)
2. Conrad, J.G., Schilder, F.: Opinion mining in legal blogs. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL 2007, pp. 231–236 (2007)
3. Dinu, A.: Short Text Categorization via Coherence Constraints. In: Proc. 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2011, Timisoara, Romania, September 26-29, pp. 247–251 (2011)
4. Feldman, R., Sanger, J.: The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)
5. Kononenko, I.: Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine* 23(1), 89–109 (2001)
6. Langley, P., Iba, W., Thompson, K.: An Analysis of Bayesian Classifiers. In: Proc. AAAI 1992, pp. 223–228 (1992)
7. Lewis, D.D.: Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In: Proc. Machine Learning: ECML-1998, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, pp. 4–15 (1998)
8. Mihalcea, R., Banea, C., Wiebe, J.: Learning Multilingual Subjective Language via Cross-Lingual Projections. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL 2007, Prague, Czech Republic, June 23-30 (2007)
9. Mihalcea, R., Pulman, S.: Characterizing Humour: An Exploration of Features in Humorous Texts. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 337–347. Springer, Heidelberg (2007)

10. Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, May 17-23 (2010)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
12. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval (FTIR) 2(1-2), 1–135 (2007)
13. Perkins, J.: Python Text Processing with NLTK 2.0 Cookbook. Packt Publishing (2010)
14. Rish, I., Watson, T.J.: An empirical study of the naive Bayes classifier. Research Center (2001), http://domino.research.ibm.com/comm/research_people.nsf/pages/rish.pubs.html
15. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques. Elsevier (2005)
16. Yessenov, K., Misailovic, S.: Sentiment Analysis of Movie Review Comments, Report on Spring 2009 final project (2009), <http://people.csail.mit.edu/kuat/courses/6.863/>
17. Beautiful Soup - HTML/XML parser for Python, <http://www.crummy.com/software/BeautifulSoup/>
18. IMDbPY - package for manipulating IMDb data for Python, <http://imdbpy.sourceforge.net/>
19. NLTK - Natural Language ToolKit, <http://www.nltk.org/>
20. PyGTK - library for implementing graphic user interfaces in Python, <http://www.pygtk.org/>
21. WebKit - web browser web, <http://www.webkit.org/>

A Domain Independent Framework to Extract and Aggregate Analogous Features in Online Reviews

Archana Bhattarai, Nobal Niraula, Vasile Rus, and King-Ip Lin

Department of Computer Science, The University of Memphis
209 Dunn Hall
Memphis, TN 38152, USA
{abhattachar, nbnraula, vrus, davidlin}@memphis.edu

Abstract. Extracting and detecting features from online reviews is both important and challenging, especially when domain knowledge is not explicitly available. Moreover, opinions about the same feature of a product or service are frequently expressed in various lexical forms. In this paper, we present a novel framework to automatically detect, extract and aggregate semantically related features of reviewed products and services. Our model uses sentence level syntactic and lexical information to detect candidate feature words, and corpus level co-occurrence statistics to perform grouping of semantically similar features to obtain high precision feature detection. The high precision feature assembly capability of our model has a distinct advantage over state of the art approaches, like double propagation, by producing short and succinct sets of features compared to potential thousands of features that are generated by existing approaches. We evaluate our model in two completely unrelated domains, restaurant and camera online reviews, to verify its domain independence. The results of our model outperformed existing state of the art probabilistic models.

Keywords: Information Retrieval, Natural Language Processing, Text Analysis.

1 Introduction

The usability and reliability of reviews have propelled them as a standard resource to assess the quality of products and services. Various studies [2], [3] have shown that online reviews have real economic values for the products they target. However, the sheer volume of the reviews spread over many sites makes it almost impossible to go through every one of them manually. A need arises to develop automated tools that can process the reviews and produce useful, aggregated information for users for quick and accurate decision making. Some of the review sites, such as amazon.com, have tried to solve this problem by presenting an average rating of the product based on the numerical value given by the users. The average rating indicates the overall sentiment/opinion orientation towards the product. In addition, they show “The most helpful favorable (and negative) review”. This, however, does not solve the problem completely as potential consumers tend to go through the few reviews at the top

which may give them a biased view of particular features of the product. Moreover, different customers may be interested in different features of the product. For example, a frequently traveling person may want a light and small camera whereas a person who is a professional photographer may be interested in the resolution and other advanced features. Thus, it is hard for the potential consumer to grasp the overall picture of existing consumers' sentiments toward the product.

An automated review mining system could offer a more detailed opinion summary for products based on existing reviews by automatically identifying features in textual reviews and the corresponding sentiment of the reviews' authors. The two tasks are called opinion detection and classification (e.g. positive, or negative). For example, "*I like this camera*" is a positive opinion and "*this camera did not produce good picture*" is a negative opinion. This is a much studied problem [4], [5], [6], [7], [8], [9]. These works mostly focus on detecting the overall opinion of the user towards the product. However, instead of detecting overall sentiment, it would be more beneficial to detect the opinion at a more fine grain level since a single review could have disparate opinions on different features of a product. For example, the sentence "*Although the picture quality of the camera is good, it is too bulky to carry*", expresses a positive opinion on the **picture quality** feature of the product and a negative opinion about the **size** feature of the product. One could argue that ratable features are already mentioned in some reviews. However, most reviews tend to be free-form text and people tend to express opinions on different product features even in a single sentence like in the previous example.

Secondly, users tend to express their opinions about a product's features in a variety of forms. For example, the opinion sentences "*I loved their sushi*" and "*the bread was really fresh*" are talking about the targets *sushi* and *bread*. Although these sentences are not expressing opinions on exactly the same subject/target, they seem to be positive about the feature *food* of some restaurant. The example above illustrates two challenging issues for an automated system; one is to identify the target of the sentence and the other is to identify what feature of the target the sentence is rating. A fully automated method would identify both the features of a particular target product and the opinions polarity. The idea of using topic models and other dimension reduction algorithms such as LSA [10], PLSA [11] and LDA [12] appear to be most suitable in unsupervised aggregation of semantically similar features in reviews and have been used in recent research work. However, these models are not precise enough in their naïve form for practical usage and also do not cover most of the sentences as they are completely uninformed of the domain, i.e. products' features, and ignore syntactic information at the sentence level. To alleviate these drawbacks, heuristics have been introduced in sentence-level LDA or focusing only on nouns at sentence level [14], [15] for better prediction of features. However, still a lot of textual content that are not relevant to ratable features reside in sentences of reviews which introduce impurities in the prediction of features of the product.

Another thread of research has exploited the lexical information and co-occurrence information at the sentence level to detect features. Some of these approaches are association rule based filtering [13] and the double propagation algorithm [1]. With the lexical/syntactic information, product/service targets and features can be extracted

with very high precision and recall. However, these methods can only extract features that are explicitly identified by certain words. As an example, from our analysis in the camera dataset [1], only 36% of sentences have explicitly mentioned features, which is very natural and frequent in human languages. Another difficult issue of feature detection is that people by nature use different words/phrases to refer to single feature, which makes it virtually impossible to only use this method to extract most prominent features from reviews. As the number of reviews increase, the number of lexical expressions referring the feature (target word) increase exponentially too. As an example, in the restaurant dataset, we extracted 10633 targets from 275512 sentences in 52624 reviews. Usually, reviews are in large number for most of the products and services and an effective review mining system should be able to provide the collective impression in the shortest possible way. Thus, the real use of these extracted targets is only seen when they are grouped into few important features.

In our approach, we propose a unified model that incorporates target and feature detection together at sentence level. With such a model, we also tackle the problem of implicit and explicit mention of features in reviews. To further explain this point, let us analyze the following sentence, “*I’ve had a lot of cameras, but this one is ACTUALLY being used, and not stored in a desk: it’s always in my shirt pocket*”. The target of this sentence is possibly *usage* whereas the sentence is actually talking about the feature *size*. Thus, if we try to extract the target first and then try to relate that target to ratable features separately, the feature *size* can never be inferred from the sentence. However, our inference model has the capability to classify sentences to one of the suitable feature classes even when feature is not explicitly mentioned in the sentence thus being able to detect the feature *size*. Both the topic model based approaches and lexical/syntactic information based approaches, as explained above, have their own pros and cons. However, no work has been done to combine both the information to get the best of both. Hence, in this work, we tackle the problem exploiting both the local and global attributes preserving domain independency and least supervision. We came up with a strategy to filter the words in SLDA (sentence LDA) so only relevant words are left behind. These filtered sentences are then fed to LDA to get co-occurrence based grouping of features. The only human intervention in our framework is the assignment of meaningful labels to LDA discovered topics. Our results show a distinct advantage of our method over the standard methods combining the advantages of least supervision (LDA based systems), high recall (sentence level feature extraction systems) and mitigating the disadvantages (requirement of explicit feature mention, inability to group similar feature words to refer single feature).

2 Related Work

Most work on review mining involves tasks of identifying related product features and classifying the sentiment of the review, either as a whole or by individual sentences/phrases. An example is the work by Hu and Liu [13]. They apply association rule mining to extract product features. To determine sentiment polarity,

they use a seed set of adjectives which are expanded via WordNet [16]. Then they select a subset or rewrite some of the original sentences to capture main points as done in traditional text summarization.

Another line of research is using a supervised framework [17]. However, these methods are only limited to a certain domain and are highly dependent on the training data. Morinaga and others [18] use a search engine to collect statements regarding target products and then extract opinions using syntactic and linguistic rules derived by human experts. The result is then used to generate statistically meaningful information. Work has also been done to make the whole process unsupervised. One example is OPINE [19]. It uses the concept of relaxation-labeling to determine the semantic orientation of potential opinion words in the context of the extracted product features and specific review sentences. Another example is in [20] where the authors proposed to generate rated aspect summary of short comments based on the overall rating provided. They propose to use unstructured and structured PLSA to identify group of words representing aspects.

One of steps in automated opinion mining is seed sentiment word detection, which are used as a bootstrapping mechanism to later identify candidate target words (words that refer some ratable property of the product/service). Many such techniques depend on some external knowledge in the form of pre-classified word list or training set to predict the polarity [21], [7]. Our method only depends on the overall rating (numerical value associate with most of the reviews that indicate overall semantic orientation of the opinion holder) of the review which is provided almost all of the time. This makes our method flexible enough for cross domain applications. Our work more closely follows the concept of unsupervised aspect classification presented in [14] and [15].

3 Methodology

The overall methodology can be seen as a combination of serial tasks such as domain specific seed sentiment words extraction, explicit feature candidate extraction based on sentence level features, assembly of target words to ratable features and classification of sentences to ratable features. In this section, we will explain each of the tasks in detail.

Figure 1 depicts the overall framework and flow of our system. Customer reviews about a product are collected from review sites on the web. Most of the reviews have ratings (a number indicating overall opinion of the review) with them. We use this number to extract seed sentiment words. In the process, we do not consider the reviews that do not have ratings associated as this step explicitly needs a rating value. The obtained seed words are then fed in the target extraction module which essentially extracts candidate target words and more sentiment words in a recursive manner. A target word/phrase indicates one of the features/properties of the product whereas a sentiment word indicates the customers sentiment (like/dislike) towards the product. These candidate feature words are then grouped by the LDA based feature identification module to identify ratable features of the product/service.

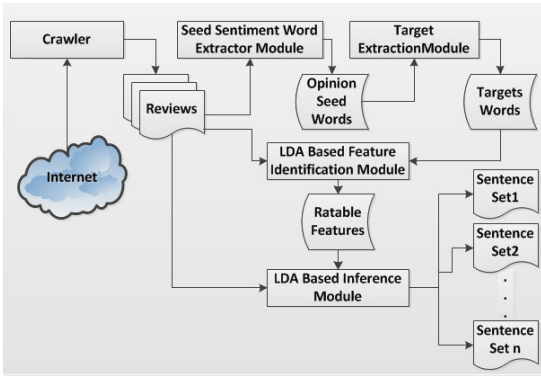


Fig. 1. Overall system framework

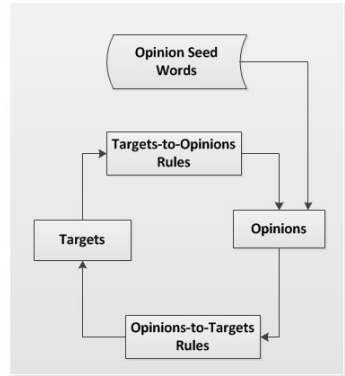


Fig. 2. Conceptual framework for target extraction module

3.1 Domain Specific Seed Sentiment Words Extraction

While the original double propagation method requires an opinion seed set to start with, our method does not require any external information. Since opinion words are completely domain dependent, it might not be so easy to obtain domain dependent opinion words for hundreds of products and services. We exploit user given ratings to collect domain dependent subjective opinions. Sentiment is expressed most of the time with adjectives. Thus, we only work with adjectives in a sentence. For each adjective in a review, we count its frequency in reviews which have positive rating and negative rating. A review is considered positive if it has a 4 or 5 star and is considered negative if it has rating of 2 or fewer stars. Here, we use majority voting to detect the underlying polarity of these sentiment words. We define the semantic orientation of a sentiment word based on its minimum frequency in positive/negative reviews and the ratio of its occurrence in them. The semantic orientation is defined by the following equation.

$$SO(w) = \frac{|w_p|}{(|w_p| + |w_n|)} \tag{1}$$

Here, w_p is the frequency of word w in positive reviews and w_n is the frequency of it in negative reviews.

If the semantic orientation of a sentiment word w (i.e. $SO(w)$) is greater than a given confidence value and it appears at least in a given support value of positive documents, the word is considered to have a positive semantic orientation. Similarly, we identify negative sentiment words. We changed the threshold values to obtain best precision and recall and hence found 60% of total documents as confidence and 2% of total documents as support are the optimal values, respectively. We also experimented with adverb phrase chunks and adjective phrase chunks, but unigrams gave best results.

3.2 Explicit Target Candidate Extraction Based on Sentence Level Features

The target words extraction methodology is based on the syntactic relation between opinion words and feature words. The process recursively discovers product/service target feature words based on known opinion words and discovers more unknown opinion words based on known target feature words.

Figure 2 above shows the conceptual framework of target extraction module. Initially it requires a seed opinion word set, the extraction method of which has been described in the previous section. The process employs a rule based strategy over dependency trees of sentences. For example, if we know **great** is an opinion word and are given a rule like “*a noun on which an opinion word directly depends through mod is taken as the target,*” we can easily extract **pictures** as the target. Similarly, if we know **picture** is a target, we could extract the adjective **great** as an opinion word using a similar rule. The paper [1] describes the extraction rules in detail.

3.3 Assembly of Target Words to Ratable Features

Opinions expressed in reviews are mostly associated with particular features of entities, i.e. products or services. Each opinion sentence mentions at least a feature along with a corresponding opinion towards the feature. So the next step is to identify representative features of the entity as sets of target words that indicate the feature.

Latent Dirichlet Allocation (LDA) and its modifications [14], [15] have recently been applied to uncover the latent topics which are not directly observable in a document. We also followed the same idea of using bag of words in documents to identify the aspects in the reviews. LDA is a generative probabilistic model well suited for discrete distinct data such as text corpora. LDA can be seen as a three-level hierarchical Bayesian model which models each item of a collection in terms of finite mixture over latent set of aspects. Each aspect is then modeled as an infinite mixture of aspect probabilities. This allows it to capture significant intra-document statistical structure. Documents are represented as random mixtures over latent aspects, where each aspect is characterized by a distribution over words. Since we are interested in the fine grained features of the entity, we assume each sentence to be a single document [14]. Thus, the output of the model is a distribution over inferred aspects for each sentence in the data. We skip the details of LDA since it is out of scope in this work. Intuitively, features are mostly presented in noun forms.

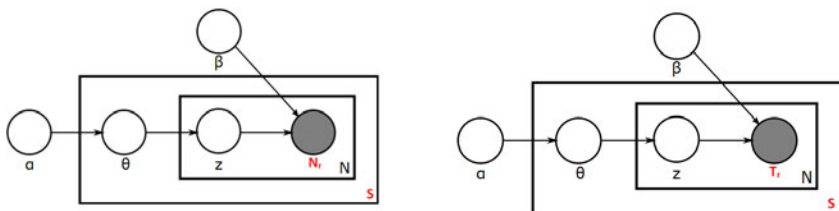


Fig. 3. (left) Noun word based LDA and (right) target word based LDA

Thus, as a preprocessing step, Brody et al [12], filtered nouns s from the sentences in reviews and use them as candidate to generate hidden features of the entity. We

extend this filtering of words to another level in our work. Instead of using all nouns in the sentence, we filter the words to only keep candidate target words obtained from the previous step as input to LDA. LDA groups a set of representative words into pre-defined number of aspects. The following diagrams and the corresponding mathematical formulation represent sentence level noun filtered LDA and sentence level target filtered LDA.

We used the Gibbs sampling based LDA for estimating the parameters [21]. Let \vec{N}_r and \vec{z} be the vectors of all nouns and their topic assignment in the collection. Then, the topic assignment for a particular noun in the review is computed as follows:

$$p(z_i = k | \vec{z}_{-i}, \vec{N}_r) = \frac{n_{k,-i}^{(k)} + \beta_k}{[\sum_{v=1}^V n_{k,-i}^{(v)} + \beta_v] - 1} \frac{n_{m,-i}^{(k)} + \alpha_k}{[\sum_{j=1}^K n_{m,-i}^{(j)} + \alpha_j] - 1} \tag{2}$$

Similarly, let \vec{T}_r and \vec{z} be the vectors of all targets and their topic assignment in the collection. The topic assignment for a particular target in the review is computed as:

$$p(z_i = k | \vec{z}_{-i}, \vec{T}_r) = \frac{n_{k,-i}^{(k)} + \beta_k}{[\sum_{v=1}^V n_{k,-i}^{(v)} + \beta_v] - 1} \frac{n_{m,-i}^{(k)} + \alpha_k}{[\sum_{j=1}^K n_{m,-i}^{(j)} + \alpha_j] - 1} \tag{3}$$

where $\vec{\alpha}$ and $\vec{\beta}$ are the Dirichlet parameters, V is the vocabulary size, $n_k^{(v)}$ is the number of times a topic k is assigned to the word v, $-i$ refers to the assignments excluding the current assignment.

3.4 Classification of Sentences Based on Features

Each test sentence is given to the LDA model to get its topic distribution. The topic distribution is then used to classify the given sentence into one of the aspects. To this end, we say that a sentence belongs to a feature ‘f’ if $P(f) > \text{threshold}$, where $P(f)$ is the probability of topic ‘f’ in the sentence.

4 Experimental Results

In this section, we will present our result in camera and restaurant domain for different tasks.

4.1 Dataset

To demonstrate the proof of concept of our model and its domain independence, we chose two completely disparate domains, restaurant (service) and camera (product). For the restaurant dataset, we used the existing dataset from [22]. It consists of 652 reviews of restaurants in New York City, of which a subset is annotated with features and orientation, which we use extensively during evaluation. For the camera dataset, we collected reviews from amazon.com to develop the framework. For the evaluation purpose, we used the data from Bing et al. We manually annotated the sentences in the dataset to 9 ratable features. The dataset was annotated by 2 graduate students of computer science. Some useful statistics of the dataset is shown in the following table.

Table 1. Dataset statistics

Dataset	Restaurant	Camera
# of Reviews	52624	3999
# of positive reviews	38653	3212
# of negative reviews	9305	522
Average rating	3.9	4.17

4.2 Seed Sentiment Words Extraction

Table 2 below shows some sample sentiment words that were detected as positive and negative based only on provided user rating using all the reviews in both the dataset. We evaluate our method using the manually annotated dataset provided by [22]. Each sentence in the dataset is positive, negative, neutral or has conflict. Among 3000, 2603 of them sentences are either positive or negative and we only use these for evaluation.

Table 2. Sample seed sentiment words

Polarity	Sample Words
Positive	absolute, luscious, golden, refreshingly, cozy, amazing
Negative	unidentifiable, tasteless, unfriendly, unprepared, unaccommodating

Table 3. Evaluation result on seed sentiment word extraction

Accuracy	Precision	Recall
76.46	83.74	89.04
73.69	100	71.48

4.3 Candidate Target Extraction Based on Double Propagation

Since we did not have any conceptual contribution on the double propagation algorithm, we did not explicitly evaluate this task. However, we did apply the algorithm in a bigger and more realistic size dataset. In the following table, we present the statistics on number of targets generated by the algorithm on camera and restaurant domain. We also show sample target words extracted in both domains.

Table 4. Target extraction statistics and sample target words

Domain	Sample Candidate Target Words	# of Targets	# of Sentences
Restaurant	Medallions , dining, roco, halibut, rock, steamy, souffles, omakase, tolerance, addict, border, caipirinas, sterling, vibes, vingar, vegetarian, shrimps, rigatoncini, soufflee	10633	275512
Camera	Plastic, cravings, incentive, zoom, competitors, desktop, parts, video, purchases, amateur, photo, chance, party, viola, pocket, aspect, touch, washed, frames, nephew	1459	20012

4.4 Grouping of Targets to Ratable Features

Tables 5 show some sample words that were identified by the LDA model.

Table 5. Sample words representing ratable features in restaurant domain

Topics	Noun Filtering Based grouping	Target Filtering Based Grouping
Price	Food, service, prices, price, quality, ambience, atmosphere, bit, ok, nothing, size,	worth, prices, bit, price, quality, portions, service, cheap, reasonable, average
Ambie nce	Food, service, staff, atmosphere, Service, wait, décor, waiters, Very, attentive, bit	Bar, room, dining, music, décor, tables, area, space, atmosphere, cozy, nice, seating
Food	Menu, sushi, dishes, Food, fish, chef, everything, variety, dish, items, specials	Pizza, steak, cheese, meat, side, burger, taste, fries, plate, chicken, burgers, bread
Anecd ote	Dinner, night, time, friend, lunch, friends, day, brunch, birthday, party, evening,	Place, recommend, love, fun, date, nice, friends, perfect, people, eat, enjoy, spot
Misc	Restaurant, experience, times, time, reviews, dining, years, place, meal, visit, couple, NYC	restaurants, experience, places, favorite, city, top, dining, years, neighborhood
Service	Table, waiter, minutes, time, order, people, reservation, hour, waitress, manager, hostess	Table, wait, minutes, order, reservation, bar, waiter, hour, waiting, reservations

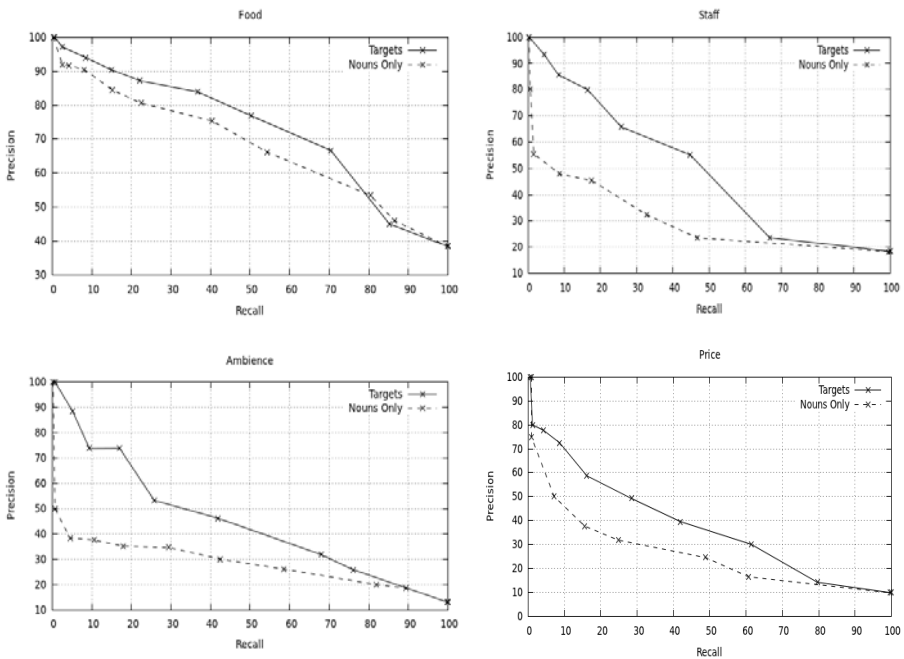


Fig. 4. Precision/Recall curves for ratable aspects food, staff, ambience and price in restaurant domain with noun filtering and target filtering

For the evaluation purpose, we used the manually annotated dataset by [22]. They have annotated around 3000 sentences from 652 restaurants for its sentiment polarity and aspect. For each sentence, they assign one or more aspects among six aspects; food, staff, ambience, anecdotes, miscellaneous and price. For the evaluation on camera domain, we used Bing data which we annotated for 9 features and we used amazon.com collected data for target extraction and grouping.

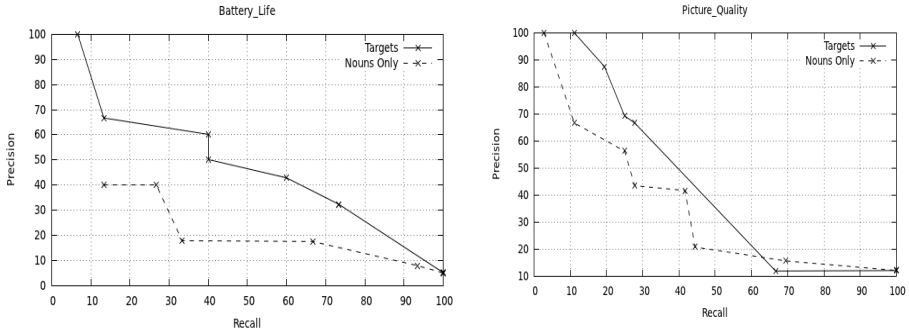


Fig. 5. Precision/Recall curves for ratable aspects battery life and picture quality with noun filtering and target filtering

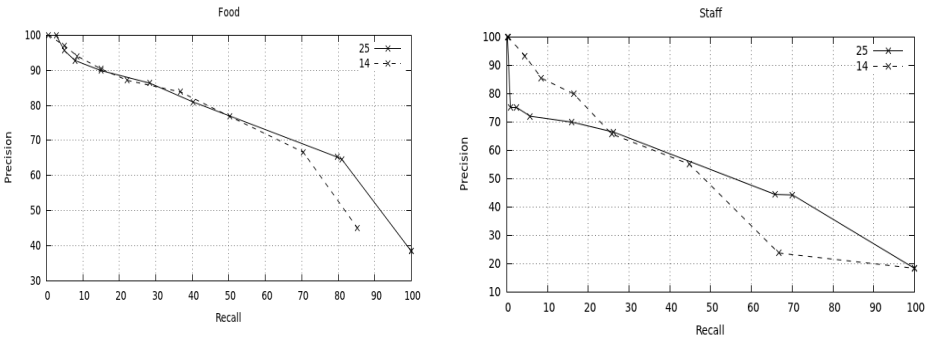


Fig. 6. Precision/Recall curves with 14 and 25 topics for ratable aspects, food and staff

Figures 4 and 5 above show the precision/recall curve for some of the ratable aspects in restaurant and camera domain. A threshold t_f for each feature is defined to classify sentence to one of the inferred features. We assume a sentence as associated to feature f if $P(f) > t_f$. By varying the threshold t_f we created precision-recall curves. This is similar to [12]. For a direct comparison to their method, we also implemented noun based filtering along with target based filtering and drew the curve that depicts significant improvement in precision/recall with the target based filtering method.

The problem of determining the model order (number of features) still persists in our method as in most unsupervised learning scenario. We thus performed a small experiment varying the number of features in precision/recall graph as shown in figure 6. As the result shows, although choosing right number of features (14 in case of restaurant domain) did seem to improve the result, the improvement was not

significant. Hence we decided to use some practical number of features that users would think of while reviewing a product/feature. For the restaurant domain, to be able to evaluate our system we chose to use 14 features with the mapping of 6 manually annotated features. For the camera domain, we tried to manually annotate the dataset from [15] and found that there are nine practical features. So we used nine features for camera dataset.

5 Discussion and Future Work

We presented a framework to identify and aggregate ratable features with minimal supervision. Our method has shown significant improvement on the identification and grouping of features in reviews. We also introduce the idea of using the star rating as a way to classify sentiments without an external corpus. Since all the steps were performed in a domain-independent way, the system is flexible enough to be equally applicable to any other entity of any domain. Though, the recall of aspect identification system is not high, in real life scenario, most of the products/services have sizable amount of reviews and hence even a low recall result could be representative and helpful to customers.

As future work, we intend to extend the system to collect all the reviews for a particular entity from the web and produce succinct information. This work can be seen as a milestone to build a system that does not perform keyword based document retrieval but actually processes relevant documents to produce precise information.

Current work ignored the requirement of identifying opinion holder of the opinion. However, if the same concept needs to be applied in more general contexts such as blog to generate a summarized form of information for an entity such as a person or a company or location, opinion holder identification cannot be ignored. Moreover, in such a scenario, we would not even have explicit sentiment polarity clue such as rating. Thus an entity based faceted opinion summarization in such case would be more challenging. We intend to explore in that direction down the line.

Acknowledgments. This research was supported in part by Institute for Education Sciences under award R305A100875. 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 sponsoring agencies.

References

1. Qiu, G., et al.: Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics* 37, 9–27 (2011)
2. Dhar, V., Chang, E., Stern, L.N.: Does Chatter Matter? The Impact of User-Generated Content on Music. CeDER Working Paper. New York University (2007)
3. Ipeirotis, P.G.: Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 99 (2010)

4. Bansal, M., Cardie, C., Lee, L.: The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In: Proceedings of COLING: Companion volume: Posters, pp. 13–16 (2008)
5. Eguchi, K., Lavrenko, V.: Sentiment Retrieval using Generative Models. In: Jurafsky, D., Gaussier, É. (eds.) ACL, pp. 345–354 (2006) ISBN: 1-932432-73-6
6. Esuli, A., Sebastiani, F.: Determining the semantic orientation of terms through gloss classification, pp. 617–624 (2005)
7. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques, pp. 79–86. Association for Computational Linguistics, Stroudsburg (2002)
8. Pang, B., Lee, L.: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, pp. 271–278 (2004)
9. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, pp. 417–424. Association for Computational Linguistics, Stroudsburg (2002), doi:<http://dx.doi.org/10.3115/1073083.1073153>
10. Dumais, S.T., et al.: Using Latent Semantic Analysis To Improve Access To Textual Information, pp. 281–285. ACM (1988)
11. Hofmann, T.: Probabilistic latent semantic indexing, pp. 50–57. ACM, New York (1999), doi:<http://doi.acm.org/10.1145/312624.312649>, ISBN: 1-58113-096-1
12. Blei, D.M., et al.: Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003)
13. Hu, M., Liu, B.: Mining opinion features in customer reviews, pp. 755–760. AAAI Press (2004) ISBN: 0-262-51183-5
14. Brody, S., Elhadad, N.: An unsupervised aspect-sentiment model for online reviews, pp. 804–812. Association for Computational Linguistics, Stroudsburg (2010) ISBN: 1-932432-65-5
15. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models, pp. 111–120. ACM, New York (2008) ISBN: 978-1-60558-085-2
16. Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM* 38, 39–41 (1995)
17. Zhuang, L., Jing, F., Zhu, X.-Y.: Movie review mining and summarization, pp. 43–50. ACM, New York (2006) ISBN: 1-59593-433-2
18. Morinaga, S., et al.: Mining product reputations on the Web, pp. 341–349. ACM, New York (2002) ISBN: 1-58113-567-X
19. Popescu, A.-M., Etzioni, O.: Extracting product features and opinions from reviews, pp. 339–346. Association for Computational Linguistics, Stroudsburg (2005)
20. Lu, Y., Zhai, C.X., Sundaresan, N.: Rated aspect summarization of short comments, pp. 131–140. ACM, New York (2009), doi:
<http://doi.acm.org/10.1145/1526709.1526728>, ISBN: 978-1-60558-487-4
21. Wiebe, J.M.: Learning Subjective Adjectives from Corpora, pp. 735–740 (2000)
22. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In: Proceedings of the Twelfth International Workshop on the Web and Databases (2009)

Learning Lexical Subjectivity Strength for Chinese Opinionated Sentence Identification

Xin Wang and Guohong Fu

School of Computer Science and Technology, Heilongjiang University
Harbin 150080, China
wangxincs@hotmail.com, ghfu@hlju.edu.cn

Abstract. Lexical subjectivity strength has proven to be of great value to subjectivity classification. However, the quantitative calculation of lexical subjectivity strength has not yet been much explored. This paper presents a fuzzy set based approach to automatically learn lexical subjectivity strength for Chinese opinionated sentence identification. To approach this task, log-linear probabilities are employed to extract a set of subjective words from opinionated sentences, and three fuzzy sets, namely low-strength subjectivity, medium-strength subjectivity and high-strength subjectivity, are then defined to represent their respective classes of subjectivity strength. Furthermore, three membership functions are built to indicate the degrees of subjective words in different fuzzy sets. Finally, the acquired lexical subjectivity strength is further exploited to perform subjectivity classification. The experimental results on the NTCIR-7 MOAT data demonstrate that the introduction of lexical subjectivity strength is beneficial to subjectivity classification.

Keywords: Opinion mining, subjectivity classification, lexical subjectivity strength, fuzzy sets, membership functions.

1 Introduction

As an important subtask of opinion mining, opinionated sentence identification (OSI), also referred to as subjectivity classification, aims to classify sentences as subjective or objective, and thus benefits many opinion mining applications, such as product review mining, opinion question answering and opinion summarization [1][2][3].

Recent years have seen a great progress in subjectivity classification, and a variety of approaches have been attempted, including machine learning methods [4][5][6] [7] and rule-based methods [8][9]. However, subjectivity classification is still a challenge. This is largely due to the particularities of subjective languages. On the one hand, unlike factual text, opinionated documents are usually expressed in a more subtle or arbitrary manner [2], which makes it very difficult to distinguish subjective sentences from objective sentences in some cases. On the other hand, many words that have proven to be good indicators of subjectivity may have both subjective and objective senses [10]. Such ambiguity raises another challenge and could produce a significance source of errors in subjectivity classification.

Most previous studies focus on exploring lexical cues for subjectivity classification [9][10][11][12][13][14]. However, this is not a trivial task. Firstly, different words in opinion expressions may have different subjective strength, and thus have different contributions to subjectivity classification [8]. Secondly, a same word may have different subjective strength in different contexts. Therefore, there is a need to discriminate the subjective strength of words for subjectivity classification. Up to this point, however, to date, there has been relatively little research conducted on lexical subjectivity strength. While lexical subjectivity strength has proven to be of great value to subjectivity classification [8][15], the quantitative calculation of lexical subjectivity strength has not yet been much explored. Finally, a subjective sentence may contain two or more subjective words, but they play different roles in expressing opinions. Opinion indicators such as ‘think’, ‘indicate’ and ‘claim’ are used for eliciting opinions but do not bear any sentiment orientations. In contrast, sentiment words like ‘satisfied’ and ‘fearful’ contain particular semantic orientations or polarities by themselves. Common sense seems to indicate that they should be considered in a different way while using them as indicators of subjectivity or opinionated sentences, but we lack empirical support.

To address the aforementioned issues, in this paper we exploit the fuzzy set theory [16][17] to learn lexical subjective strength for Chinese subjectivity classification at sentence level. To approach this task, we first employ log-linear probabilities to weight and extract subjective words from the training data. Then, we define three fuzzy sets to represent words of different subjective strength and thus construct three membership functions to further divide the extracted subjective words into low-strength subjective words, medium-strength subjective words or high-strength subjective words. To examine the effects of different subjective words on sentence-level subjectivity classification, we consider both opinion indicators and sentiment words as potential cues for subjectivity. The acquired lexical strength is finally used to perform sentence-level subjectivity classification. Compared with existing studies, the proposed technique provides a straightforward way to measure the subjective strength of words in terms of their membership functions of different fuzzy sets constructed from corpora and thus can handle in a sense the fuzzy characteristic of subjective languages. We show that the proposed approach can achieve a promising performance on the NTCIR-7 MOAT test data [18].

The rests of this paper proceed as follows. In section 2, we present a log-linear probability based scheme for Chinese subjective word weighting and extraction. Section 3 describes a fuzzy-set based technique for learning lexical subjective strength. Section 4 presents the experimental results. Finally in Section 5, we give our conclusions and some possible directions for future research.

2 Subjective Word Extraction

We extract subjective words from the training data in three steps: Firstly, all verbs and adjectives or adverbs within opinionated sentences are considered as potential opinion indicators and sentiment words, respectively. Then, the association of these subjective word candidates with subjectivity is weighed with log-linear probabilities [19]. Finally, some noise words that cannot be opinion indicators or sentiment words are filtered out in terms of their weights.

2.1 Potential Subjective Words in Chinese

In general, an opinionated sentence involves an opinion holder, an opinion indicator and sentiment expression information. In Chinese, opinion indicators refer to a set of special verbs such as 認為 ‘consider’, 说 ‘say’ and 主張 ‘advocate’, which are used by opinion holders to express their reviews on a special opinion target. Sentiment expression information is often represented as a combination of various words, mainly including polar words bearing positive or negative orientations (e.g. 美麗 ‘beautiful’), some adversative conjunctions like 但是 ‘but’, some predict adverbs like 可能 ‘may be’, and some negation words like 不 ‘no’. For convenient, we refer these sentiment-associated words as sentiment words. To examine the effects of different types of subjective words on subjectivity classification, we consider both verbal opinion indicators and sentiment words as potential indicators of subjectivity.

2.2 Weighing Subjective Words

To weigh subjective words within opinionated sentences, we calculate log-linear probabilities. The advantage of log-linear probabilities over other weighing methods like Chi-square and mutual information is that it can handle the interaction between words and their associated subjectivity classes (viz. subjectivity or objectivity) as well as the interaction between words themselves during computing lexical subjectivity weights. The details of the log-linear probability can be seen in [19].

To begin with, we need to count the frequencies and probabilities of candidate subjective words and their sentences from the training data, and store them in a contingency table shown in Table 1.

Table 1. Contingency tables of frequencies and probabilities for weighting subjective words

Words	Frequency			Probability		
	Subjective	Objective	\sum_j	Subjective	Objective	\sum_j
W_1	n_{11}	n_{12}	$n_{1\cdot}$	p_{11}	p_{12}	$p_{1\cdot}$
...
w_k	n_{k1}	n_{k2}	$n_{k\cdot}$	p_{k1}	p_{k2}	$p_{k\cdot}$
\sum_i	$n_{\cdot 1}$	$n_{\cdot 2}$	N	$p_{\cdot 1}$	$p_{\cdot 2}$	P

Where, W denotes subjective words in the training corpus, and C denotes the binary class set, namely {subjective sentences, objective sentences}, n_{ij} denotes the frequency of the subjective word w_i ($1 \leq i \leq k$) within a sentence of a special subjectivity class c_j ($j = 1, 2$), the corresponding probability $p_{ij} = n_{ij}/n$, and n is the sum of all n_{ij} .

As shown in Equation (1), the probabilities in Table 1 can be further represented as a log formulation.

$$\eta_{ij} = \ln p_{ij} = \ln(p_{i\cdot} p_{\cdot j} \frac{p_{ij}}{p_{i\cdot} p_{\cdot j}}) = \ln p_{i\cdot} + \ln p_{\cdot j} + \ln \frac{p_{ij}}{p_{i\cdot} p_{\cdot j}}. \tag{1}$$

Let $\eta_{i\bullet} = \sum_{j=1}^2 \eta_{ij}$, $\eta_{\bullet j} = \sum_{i=1}^k \eta_{ij}$ and $\eta_{\bullet\bullet} = \sum_{i=1}^k \sum_{j=1}^2 \eta_{ij}$. Thus, the average log-probabilities can be calculated by the following formulas, respectively.

$$\bar{\eta}_{i\bullet} = \frac{1}{2} \sum_{j=1}^2 \eta_{ij}, \bar{\eta}_{\bullet j} = \frac{1}{k} \sum_{i=1}^k \eta_{ij}, \bar{\eta}_{\bullet\bullet} = \frac{1}{2 \times k} \sum_{i=1}^k \sum_{j=1}^2 \eta_{ij}. \tag{2}$$

Let $\gamma_{ij} = \eta_{ij} - \bar{\eta}_{i\bullet} - \bar{\eta}_{\bullet j} + \bar{\eta}_{\bullet\bullet}$, $\hat{p}_{1j} = n_{1j} / n$, $\hat{p}_{i\bullet} = n_{i\bullet} / n$, $\hat{p}_{\bullet j} = n_{\bullet j} / n$. Where γ_{ij} is the interaction between the word w_i and the subjectivity category C_j . $\gamma_{ij} > 0$ indicates that w_i and C_j have a positive interaction, while $\gamma_{ij} < 0$ means they have an anti-interaction effect. w_i would not have any relationship with C_j when $\gamma_{ij} = 0$. Furthermore, we can define $\hat{\eta}_{ij}$, $\hat{\eta}_{i\bullet}$, $\hat{\eta}_{\bullet j}$, and $\hat{\eta}_{\bullet\bullet}$ using formulas (3), respectively.

$$\left\{ \begin{aligned} \hat{\eta}_{ij} &= \ln \hat{p}_{ij} = \ln n_{ij} - \ln n \\ \hat{\eta}_{i\bullet} &= \frac{1}{2} \sum_{j=1}^2 \ln \eta_{ij} = \frac{1}{2} \sum_{j=1}^2 \ln \frac{n_{ij}}{n} = \frac{1}{2} \sum_{j=1}^2 (\ln n_{ij} - \ln n) \\ \hat{\eta}_{\bullet j} &= \frac{1}{k} \sum_{i=1}^k \ln \eta_{ij} = \frac{1}{k} \sum_{i=1}^k \ln \frac{n_{ij}}{n} = \frac{1}{k} \sum_{i=1}^k (\ln n_{ij} - \ln n) \\ \hat{\eta}_{\bullet\bullet} &= \frac{1}{2k} \sum_{i=1}^k \sum_{j=1}^2 \eta_{ij} = \frac{1}{2k} \sum_{i=1}^k \sum_{j=1}^2 \ln \frac{n_{ij}}{n} = \frac{1}{2k} \sum_{i=1}^k \sum_{j=1}^2 (\ln n_{ij} - \ln n) \end{aligned} \right. \tag{3}$$

Thus, γ_{ij} can be estimated by the following equation.

$$\hat{\gamma}_{ij} = \hat{\eta}_{ij} - \hat{\eta}_{i\bullet} - \hat{\eta}_{\bullet j} + \hat{\eta}_{\bullet\bullet}. \tag{4}$$

Actually, $\hat{\gamma}_{ij}$ measures the contribution of the selected word w_i within a subjective sentence in C_j . In the present study, we use $\hat{\gamma}_{ij}$ to weight subjective words. Table 2 illustrates some subjective words and their $\hat{\gamma}_{ij}$ values from the training data.

Table 2. The weight of some typical opinion indicators and sentiment words

Opinion indicators	$\hat{\gamma}_{ij}$	Sentiment words	$\hat{\gamma}_{ij}$
認為 ‘consider’	3.6310	必然 ‘inevitable’	2.3510
指出 ‘indicate’	3.6042	看好 ‘satisfied’	2.2740
宣稱 ‘claim’	2.3900	害怕 ‘scare’	2.0715
預測 ‘predict’	2.3233	污染 ‘pollution’	1.9279
報導 ‘report’	-1.7848	接見 ‘receive’	-0.5214
記錄 ‘record’	-2.1916	發行 ‘release’	-0.5389

3 Learning Lexical Subjective Strength Using Fuzzy Sets

Based on lexical subjectivity weights, in this section we continue to apply the fuzzy set theory to discriminate subjective words in terms of their subjective strength. To do this, we first divide lexical subjective strength into three levels, namely *high subjective strength*, *medium subjective strength* and *low subjective strength*. Then, we define three fuzzy sets to represent the three levels of subjective strength and further construct their member functions from the weighting data shown in Table 2. These three member functions will be used to determine which subset a subjective word should belong to.

3.1 Membership Functions

In this paper, three triangular membership functions are defined to represent the three levels of lexical subjective strength. In fact, triangular membership functions are appropriate for distribution with different stages of a liner increasing. For example, people of different ages, namely “*young people*”, “*middle-aged people*”, “*old people*”, are always represented by triangular membership functions. We choose triangular membership functions in that there is a similar distribution with respect to subjective strength discrimination of three different stages.

Let $M = \{m_1, m_2, m_3\}$ be the center set of the three levels of subjective strength, then their triangular membership functions can be defined by formulas (5), (6) and (7), respectively.

$$T_{low}(x) = \begin{cases} 1 & x \leq m_1 \\ \frac{m_2 - x}{m_2 - m_1} & m_1 < x < m_2 \\ 0 & x \geq m_2 \end{cases} \quad (5)$$

$$T_{high}(x) = \begin{cases} 1 & x \geq m_3 \\ \frac{x - m_2}{m_3 - m_2} & m_2 < x < m_3 \\ 0 & x \leq m_2 \end{cases} \quad (6)$$

$$T_{medium}(x) = \begin{cases} 0 & x \leq m_1 \\ \frac{x - m_1}{m_2 - m_1} & m_1 < x < m_2 \\ \frac{m_3 - x}{m_3 - m_2} & m_2 < x < m_3 \\ 0 & x \geq m_3 \end{cases} \quad (7)$$

Algorithm 1. The SOM algorithm for clustering words based on their subjectivity strength.

Input: The subjective word weight set X .

Output: The cluster centers m_1, m_2 and m_3 .

Initialize

$$m_i[0] = \min(x) + (\max(x) - \min(x)) \times (i - 1) / (k - 1). \text{ where, } x \in X, i=1,2,3.$$

Iteration

$$t \leftarrow 0;$$

Repeat until $D(X, M)^t = D(X, M)^{t-1}$

Randomly select $x \in X$, denoted as $x[t]$;

Compute the nearest center $m_c[t]$ to $x[t]$

$$\|x[t] - m_c[t]\| = \min_i(\|x[t] - m_i[t]\|);$$

$m_c[t+1] \leftarrow m_c[t] + \eta[t] \times (x[t] - m_c[t])$, where $\eta[t] = 1/t$ is a given learning rate;

$$m_i[t+1] \leftarrow m_i[t], i \neq c;$$

$$t \leftarrow t + 1;$$

To determine the center set for each membership function, we use the self-organizing feature mapping (SOM) algorithm [19], as shown in Algorithm 1. The reason is due to the fact that the uneven distribution of words in the training corpus may cause data sparseness during the estimation of subjectivity weights. Furthermore, in comparison with other clustering methods like k-means, SOM is not sensitive to the selection of initial center and offers a good solution for isolated points during clustering. In fact, the SOM algorithm utilizes an error propagation to correct the distance from samples to the center, until convergence is achieved.

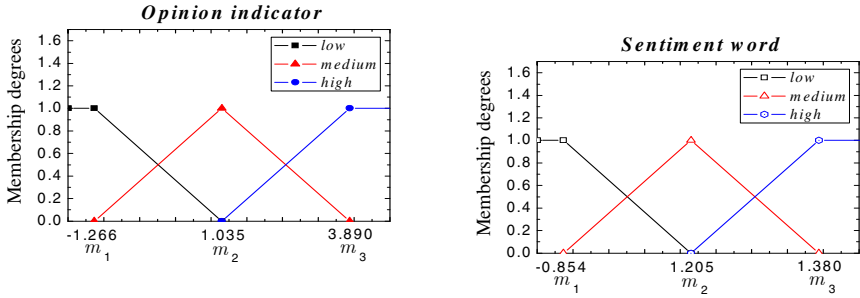
Let X be a set of subjective word weights, and $M = (m_1, m_2, \dots, m_k)$ be the cluster center of the membership functions, where, $k=1, 2,$ and 3 . Then the distance between X and M can be defined by Equation (8).

$$D(X, M) = \sum_{x \in X} \min_i \|x - m_i\|. \tag{8}$$

At the beginning of clustering, the cluster centers m_1, m_2 and m_3 should be initialized with Equation (9).

$$m_i[0] = \min(x) + (\max(x) - \min(x)) \times (i - 1) / (k - 1). \tag{9}$$

During the first iteration (viz. $t = 0$, t is the times of iteration), the clustering procedure will randomly select a score (denoted as $x[t]$) from X . Then, it will find the nearest center $m_c[t]$ to $x[t]$, and update the nearest center. The process iterates until $D(X, M)$ converges. Algorithm 1 details the clustering procedure.



(a) Membership functions for opinion indicators (b) Membership functions for sentiment words

Fig. 1. Membership functions of the three fuzzy sets for subjective words in terms of subjective strength. For opinion indicators, we have $m_1 = -1.226$, $m_2 = 1.035$ and $m_3 = 3.890$, while for sentiment words, we have $m_1 = -0.854$, $m_2 = 1.205$, and $m_3 = 3.114$.

Using Algorithm 1, we can calculate the clustering centers for opinion indicators and sentiment words in terms of their subjective weights, and thus construct their membership functions, as shown in Fig.1.

3.2 Lexical Subjective Strength Determination

Given a subjective word and its subjectivity weight, we can thus apply the above membership functions to compute its membership degree for each fuzzy set, and further determine their corresponding subjective strength under the principle of maximum membership. The basic idea is as follows: Let $\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n$ be the fuzzy sets of X , $\exists x_0 \in X$, if $\tilde{A}_k(x_0) = \max(\tilde{A}_1(x_0), \tilde{A}_2(x_0), \dots, \tilde{A}_n(x_0))$, then x_0 is a membership of the fuzzy set \tilde{A}_k .

Thus, the subjective words in Table 2 can be classified into different fuzzy subsets in terms of their lexical subjective strength, as shown in Table 3 and Table 4, respectively.

Table 3. Fuzzy classification of some opinion indicators in terms of membership degrees

Opinion indicators	Membership degrees of words for each fuzzy set			Final fuzzy set
	Low	Medium	High	
認為 ‘consider’	0	0.0907	0.9093	High
指出 ‘indicate’	0	0.1001	0.8999	High
宣稱 ‘claim’	0	0.5253	0.4747	Medium
預測 ‘predict’	0	0.5487	0.4513	Medium
報導 ‘report’	1	0	0	Low
記錄 ‘record’	1	0	0	Low

Table 4. Fuzzy classification of some sentiment words in terms of membership degrees

Sentiment words	Membership degrees of words for each fuzzy set			Final fuzzy set
	Low	Medium	High	
必然 ‘inevitable’	0	0.5809	0.4191	High
看好 ‘satisfied’	0	0.4399	0.5601	High
害怕 ‘scare’	0	0.6212	0.3788	Medium
污染 ‘pollution’	0	0.6689	0.3311	Medium
接見 ‘receive’	0.8383	0.1617	0	Low
發行 ‘release’	0.8468	0.1532	0	Low

4 Experimental Results and Discussions

To assess our approach, we developed a rule-based subjectivity classifier and conducted experiments on the NTCIR-7 MOAT data for traditional Chinese opinion analysis [18]. This section reports our experimental results.

4.1 The Subjectivity Classifier

To investigate the effect of lexical subjective strength on subjectivity classification, we develop a rule-based classifier for Chinese sentence-level subjectivity classification, which classify a given sentence as subjective or objective by looking for the occurrences of medium-strength and/or high-strength subjective words. Unlike [8] that uses a single rule for subjective sentence detection, we employ a set of more comprehensive rules to handle both opinion indicators and sentiment words for subjectivity classification. The rules are defined as follows.

Rule 1: *IF* the number of high-strength indicators or medium-strength indicators within a sentence is larger than a given threshold δ , *THEN* it is a subjective sentence.

Rule 2: *IF* the number of high-strength sentiment words or medium-strength sentiment words within a sentence is larger than a given threshold μ , *THEN* it is a subjective sentence.

Rule 3: *IF* the number of high-strength indicators or medium-strength indicators within a sentence is larger than a given threshold δ , *THEN* it is a subjective sentence; *ELSE IF* the number of high-strength sentiment words or medium-strength sentiment words within a sentence is larger than a given threshold μ , *THEN* it is a subjective sentence.

Where, δ and μ are two thresholds that can be empirically determined through experiments.

Obviously, Rule 1 and Rule 2 take into account opinion indicators and sentiment words, respectively for subjective sentence detection, while Rule 3 can handle both of them simultaneously for subjectivity identification.

4.2 Experimental Setup

The experimental data mainly come from the NTCIR-7 MOAT data for traditional Chinese opinion analysis. To achieve a reliable estimation for lexical subjective weight and strength, the NTCIR-6 data for traditional Chinese opinion analysis are also added to the training set. Table 5 presents the basic statistics of the experimental data.

Table 5. Basic statistics of the experimental data

Item	Training data	Test data
Topic	35	14
Document	901	188
Sentence	13415	4655

For comparison, in this paper the performance is reported in terms of the same metrics as used in NTCIR-7. They are F-score (F), recall (R), precision (P) under the LWK evaluation approach with the lenient standards [18].

4.3 Experimental Results

4.3.1 Opinion Indicators vs. Sentiment Words

Our first experiment intends to examine the contribution of different types of subjective words for subjectivity classification. This experiment is conducted by applying the above three rules to the test data and evaluating the output, respectively. The results are summarized in Table 6.

Table 6. Evaluation results for different classifiers using different rules with $\delta=1$ and $\mu=1$

Classifier	P	R	F
Rule 1	0.7085	0.5551	0.6225
Rule 2	0.8144	0.5363	0.6467
Rule 3	0.7175	0.8006	0.7567

It can be observed from Table 6 that the classifiers based on Rule 1 and Rule 2 obtained a high precision but a relatively low recall, which is similar to that of (Riloff and Wiebe, 2003). In addition, the classifier based on Rule 3 achieved the best performance as a whole, showing the positive effect of combining opinion indicators and sentiment words on subjectivity classification. Another interesting observation is that the classifier based on Rule 2 yielded the best precision (81.44%) but the worse recall (53.63%). The reason might be that most subjective sentences in real opinionated documents may contain sentiment words but their subjective strength is not high. As a result, a number of opinionated sentences would not be successfully identified by the classifier only using Rule 2.

4.3.2 Effects of Lexical Subjectivity Strength

The goal of our second experiment is to investigate the effects of using lexical subjectivity strength on opinionated sentence identification. This experiment is conducted by comparing our system with a baseline system and the WIA-Opinmine system [20] for NTCIR-7 MOAT. The baseline system involves a lexicon of subjective words, which contains a total of 8596 opinion words extracted from the CUHK sentiment lexicon and the NTU sentiment dictionary [18]. The lexicon is used to identify the appearance of subjective words for opinionated sentence detection. It should be noted

that the baseline system does not discriminate subjective words of different subjective strength. In other words, it completely ignores lexical subjective strength during opinionated sentence detection. The WIA-Opinmine system used a coarse-fine strategy to explore multiple complicated features for subjectivity classification [20]. It achieved the best performance among all participants of the NTCIR-7 MOAT. The results are summarized in Table 7.

Table 7. Comparison of our system with other systems under the lenient standard

Systems	P	R	F
Base line	0.5288	0.8511	0.6523
WIA-Opinmine	0.6520	0.8698	0.7453
Our system	0.7175	0.8006	0.7567

As can be seen from Table 7, our system performs better than WIA-Opinmine in precision and F-score, but worse in recall. This might be due to the fact that WIA-Opinmine has used more features than our system, and is thus capable of identifying more subjective sentences. Furthermore, our system outperform the baseline system without using lexical subjectivity strength by more than ten percents, showing in a sense that lexical subjectivity strength play an important role in opinionated sentence identification.

5 Conclusions and Future Work

In this paper, we have presented a log-linear probability-based scheme for weighing subjective words and a fuzzy set based technique for learning lexical subjectivity strength. We have also evaluated our approach on the NTCIR-7 MOAT data. The experimental results show that both opinion indicators and sentiment words are very important indicators of subjectivity while their contributions for subjectivity classification may be different. We have also demonstrated that lexical subjectivity strength can result in a significant performance improvement in subjectivity classification.

The encouraging results of the present study suggest several possibilities for future research. To further enhance our system, in future work we intend to exploit a more tailored method, to optimize the membership functions for subjective words. Also, we plan to exploit fuzzy decision trees for subjectivity analysis. The present study focuses on Chinese subjectivity recognition. In future, we also plan to extend our current system and apply it to other languages like English.

Acknowledgments. The authors would like to thank Chinese University of Hong Kong, National Taiwan University and NTCIR for their data. This study was supported by National Natural Science Foundation of China under Grant No.60973081 and No.61170148, the Returned Scholar Foundation of Educational Department of Heilongjiang Province under Grant No.1154hz26, and Harbin Innovative Foundation for Returnees under Grant No.2009RFLXG007, respectively.

References

1. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning Subjective Language. *Computational Linguistics* 30(3), 277–308 (2004)
2. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
3. Liu, B.: Sentiment Analysis and Subjectivity. In: Indurkha, N., Damerau, F.J. (eds.) *Handbook of Natural Language Processing* (2010)
4. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: *Proceedings of ACL 2004*, pp. 271–278 (2004)
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: *Proceedings of HLT/EMNLP 2005* (2005)
6. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics* 35, 399–433 (2009)
7. Breck, E., Choi, Y., Cardie, C.: Identifying Expressions of Opinion in Context. In: *Proceedings of IJCAI 2007* (2007)
8. Riloff, E., Wiebe, J.: Learning Extraction Patterns for Subjective Expressions. In: *Proceedings of EMNLP 2003* (2003)
9. Riloff, E., Wiebe, J., Phillips, W.: Exploiting Subjectivity Classification to Improve Information Extraction. In: *Proceedings of AAAI 2005* (2005)
10. Akkaya, C., Wiebe, J., Mihalcea, R.: Subjectivity Word Sense Disambiguation. In: *Proceedings of EMNLP 2009* (2009)
11. Yu, H., Hatzivassiloglou, V.: Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In: *Proceedings of EMNLP 2003* (2003)
12. Wiebe, J., Mihalcea, R.: Word Sense and Subjectivity. In: *Proceedings of ACL 2006* (2006)
13. Jijkoun, V., Rijke, M., Weerkamp, W.: Generating Focused Topic-Specific Sentiment Lexicons. In: *Proceedings of ACL 2010*, pp. 585–594 (2010)
14. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In: *Proceedings of EMNLP 2010*, pp. 56–65 (2010)
15. Paltoglou, G., Thelwall, M.: A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In: *Proceedings of ACL 2010*, pp. 1386–1395 (2010)
16. Fu, G., Wang, X.: Chinese Sentence-level Sentiment Classification Based on Fuzzy Sets. In: *COLING 2010 (Poster)*, pp. 312–319 (2010)
17. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
18. Seki, Y., Evans, D.K., Ku, L.-W., Sun, L., Chen, H.-H., Kando, N.: Overview of Multilingual Opinion Analysis Task at NTCIR-7. In: *Proceedings of NTCIR-7*, pp. 185–203 (2008)
19. He, X.: *Modern Statistical Analysis Methods and Applications*. Renmin University of China Publishing House (2008)
20. Kohonen, T.: *Self-organization and associative memory*. Springer, Berlin (1998)
21. Xu, R.F., Wong, K.F., Xia, Y.Q.: Coarse-Fine Opinion Mining – WIA in NTCIR-7 MOAT Task. In: *Proceedings of NTCIR-7* (2008)

Building Subjectivity Lexicon(s) from Scratch for Essay Data

Beata Beigman Klebanov¹, Jill Burstein¹, Nitin Madnani¹,
Adam Faulkner², and Joel Tetreault¹

¹ Educational Testing Service

{bbeigmanklebanov, jburstein, nmadnani, jtetreault}@ets.org

² Graduate Center, The City University of New York
adamflkr@gmail.com

Abstract. While there are a number of subjectivity lexicons available for research purposes, none can be used commercially. We describe the process of constructing subjectivity lexicon(s) for recognizing sentiment polarity in essays written by test-takers, to be used within a commercial essay-scoring system. We discuss ways of expanding a manually-built seed lexicon using dictionary-based, distributional in-domain and out-of-domain information, as well as using Amazon Mechanical Turk to help “clean up” the expansions. We show the feasibility of constructing a family of subjectivity lexicons from scratch using a combination of methods to attain competitive performance with state-of-art research-only lexicons. Furthermore, this is the first use, to our knowledge, of a paraphrase generation system for expanding a subjectivity lexicon.

Keywords: essay writing, sentiment analysis, sentiment polarity, subjectivity lexicon, C5.0, lexicon expansion, paraphrase generation, thesaurus resources.

1 Introduction

For commercial applications of sentiment analysis, an in-house subjectivity lexicon needs to be constructed, since existing lexicons, such as MPQA [1] and GI [2], are available either for research and education only¹ or under GNU GPL license that disallows the incorporation of the resource into proprietary materials.² In this article, we describe a methodology for creating a family of subjectivity lexicons from scratch through the following phases: (1) a lexicon of about 400 words was manually constructed based on materials in our domain of interest (test-taker essays), (2) a small-scale annotation was conducted to augment the lexicon to 750 words, and (3) a variety of expansion methods with subsequent *human* and *automated* clean-up were implemented. We show that this process results in subjectivity lexicons that are comparable to state-of-art lexicons in terms of sentiment classification performance

¹ “This version of the General Inquirer is made available exclusively for educational and research purposes.” From http://www.wjh.harvard.edu/~inquirer/j1_1/manual

² “The GNU General Public License does not permit incorporating your program into proprietary programs.” From <http://www.gnu.org/copyleft/gpl.html>

on our data as well as in terms of effective coverage (the number of words in a lexicon that appear in our data).

The article is organized as follows. Section 2 details the process of lexicon construction, starting from the 750-word seed lexicon (section 2.1), then discussing the automatic lexicon expansions (section 2.2), proceeding to the manual clean-up using Amazon Mechanical Turk (AMT) (section 2.3) and automatic clean-up through lexicon combination (section 2.4). Section 3 details the evaluation of the lexicons; the setup for evaluation is described in sections 3.1 and 3.2, section 3.3 compares the lexicons in terms of effective coverage of our data, section 3.4 provides the comparative evaluation of the lexicons on the sentence-level sentiment classification task. Table 4 in section 3.4 presents our main results. Section 4 surveys related work. We discuss our results and conclude in section 5.

2 Building Subjectivity Lexicons

2.1 Seed Lexicon

First, we randomly sample 5,000 essays from a corpus of about 100,000 essays containing writing samples across many topics. Essays were responses to several different writing assignments, including graduate school entrance exams, non-native English speaker proficiency exams, and accounting exams. We manually selected positive and negative sentiment words from the full list of word types in these data; these constitute our **Lexicon 0**, which contains 407 words.

We sampled 878 sentences containing at least one word from Lexicon 0, thus biasing the sample towards sentiment-bearing sentences. The motivation for the bias was increasing the incidence of sentiment-bearing – positive (**POS**) and negative (**NEG**) – sentences, under the assumption that sentiment-bearing sentences had more positive and negative words, and hence, were more effective for lexicon development. Using these sentences, we proceeded with an annotation task as follows. Two research assistants annotated 878 sentences with sentence-level sentiment polarity; 248 of these were also annotated for all words that contribute to the sentiment of the sentence or go against it. We refer to the 248 sentence set as **L-1**, and to the 630 sentence set **L-2**. For example, the following sentence was labeled as positive; words contributing to the positive sentiment are bold-faced and words (and phrases) going against it are underlined.

Some may even be **impressed** that we are **confident** enough to risk showing a lower net income.

In addition, positive and negative sentences from the **T-1** dataset (to be described in section 3.2) were annotated using AMT for words that most contribute to the overall sentiment of the sentence (marking words that go against the dominant sentiment was omitted to simplify the protocol). Each sentence was assigned to 5 AMT annotators; all words marked by at least 3 AMT annotators were selected.

Finally, the **Seed Lexicon** was created by adding to Lexicon 0 all words³ marked in L-1 annotations and all words selected from the AMT annotations; the authors then

³ When annotators could not attribute sentiment to single words, they marked phrases. Our current lexicons make no use of multi-word annotations.

performed a manual clean-up. The resulting Seed Lexicon contains 749 words, 406 positive and 343 negative. L-2 was not used in lexicon creation. However, the labeled sentences in that set were used in the evaluation experiments described in section 3.2.

2.2 Automatically Expanding the Seed Lexicon

We used three sources to automatically expand the Seed Lexicon: WordNet [3], Lin’s distributional thesaurus [4], and a pivot-based paraphrase generation tool [5]. The resulting lexicons will be called Raw WN, Raw Lin, and Raw Para, respectively; they were created as follows. Please see column 2 of Table 3 for sizes of these lexicons.

Raw WN. We used WordNet to extract the first three synonyms of the first sense of each word in the Seed Lexicon, restricting returned words to those with the same part-of-speech as the original word. The selection process was based on previous research [6] that showed that these constraints are likely to produce strong synonym candidates without a large number of false positives.

RAW Lin. Lin’s proximity-based thesaurus trained on our in-house essay data as well as on well-formed newswire texts provided an additional resource for lexicon expansion. All words with the proximity score > 1.80 to any of the Seed Lexicon words were included in the expansion.

RAW Para. We used a pivot-based lexical and phrasal paraphrase generation system. This system operates by extracting bilingual correspondences from a bilingual corpus, identifying phrases and words in Language A that all correspond to the same phrase or word in Language B and pivoting on the common phrase or word to extract all Language A words and phrases as paraphrases of each other. More details can be found in [7]. We use the French-English parallel corpus (approx. 1.2 mln sentences) from the corpus of European parliamentary proceedings [8]. The base paraphrase system is susceptible to noise due to the imperfect bilingual word alignments. Therefore, we implement additional heuristics in order to minimize the number of noisy paraphrase pairs [5]. For example, one such heuristic filters out any pairs where a function word may have been inferred as a paraphrase of a content word. For lexicon expansion experiments, we use the top 15 single-word paraphrases for every word from the Seed Lexicon, excluding morphological variants of the seed word.

Table 1. Examples of words added through various expansion methods

Seed Word	WN expansion	LIN expansion	Para expansion
abuse	ill-treatment	harassment	exploitation
accuse	incriminate	indict	reproach
anxiety	anxiousness	anguish	disquiet
conflict	battle	clash	crisis
costly	dearly-won	burdensome	onerous
dangerous	unsafe	deadly	toxic
improve	amend	enhance	reinforce
invaluable	priceless	valuable	precious

Table 1 shows some examples for each expansion method. We note that *only* the distributional thesaurus is based on in-domain data, while the other expansions use either a general-purpose dictionary (WordNet) or out-of-domain training materials (parliamentary proceedings). It therefore remains to be seen whether the generated expansions are sufficiently general to apply to our domain.

2.3 Manual Clean-Up of Expanded Lexicons

The expanded lexicons did require some clean-up. For example, antonyms of positive words ended up in the positive lexicon. This could happen, especially when using the distributional thesaurus, since antonyms and their synonyms tend to appear in the same distributions (for example, a *good* paper; a *bad* paper). In order to narrow down the expansions to words that carry positive or negative sentiment, we employ Amazon Mechanical Turk again. All words generated by at least one expansion mechanism were included in this new task. This time, AMT annotators were asked to label single words as positive, negative or neutral. Each word received three annotations. Two filtering criteria were used to generate the Cleaned versions based on the original raw expanded lexicon: (a) Each word had to be tagged as POS or NEG polarity by the majority of the three annotators, and (b) The majority polarity had to match the expected polarity of the word, that is, the polarity of the word from the Seed Lexicon for which the current word had been generated as an expansion. The clean-up procedures resulted in 16% to 41% reduction in the Raw lexicon sizes (see Table 3 for the sizes of the Raw and Cleaned lexicons), although the Cleaned lexicons are still at least twice the size of the Seed Lexicon.

2.4 Automatic Clean-UP of Expanded Lexicons through Lexicon Combination

We also experimented with an alternative strategy for noise reduction in the expanded lexicons. This strategy is based on the assumption that the three sources and mechanisms of expansion are sufficiently different to provide independent evidence of a word's relatedness to the seed word it was expanded from. Therefore, in the generation of new lexicons, where automated clean-up was applied, we introduced into the new lexicon *only* words that were produced by both of the two expansion methods of choice. Thus, we created the Raw WN + Raw Lin lexicon that contains the Seed Lexicon expanded with words with the same polarity that were both in Raw WN and in Raw Lin. Raw WN + Raw Para lexicon and Raw Lin + Raw Para lexicon were generated in a similar fashion. This procedure resulted in the elimination of up to 63% of the larger of the two raw lexicons; for the lexicon sizes, see Table 3. Still, these new lexicons are 29%-55% larger than the Seed Lexicon, providing significant expansion without human intervention.

3 Evaluating the Quality of the Lexicons

3.1 Evaluation Methodology

We used C5.0 [9], a decision-tree classifier, to evaluate the effectiveness of the different lexicons for classifying sentiment polarity of sentences. Each lexicon is represented by just two features: (1) the number of positive words in the sentence, and

(2) the number of negative words in the sentence. A number of experiments were run with additional features, but using only these two features produced the highest accuracies. For instance, an additional feature that was tried was the difference between the number of positive and negative words in a sentence. This is relevant since a sentence with a positive polarity, for instance, can contain negative words. We hypothesized that a large difference in counts might help to predict the dominant polarity. However, adding this difference feature did not boost performance.

3.2 Data Sets for Training and Testing

To generate the data for training and testing C5.0, the following strategies were employed. We used our pool of 100,000 essays to sample a second, non-overlapping set of 5,000 essays. From these essays, we randomly sampled 550 sentences, and submitted them to sentiment polarity annotation by two research assistants, both of whom had experience doing linguistic annotation. Fifty of the sentences were annotated by both annotators, with a resulting Kappa of 0.8. Sentences labeled as incomprehensible or as containing both negative and positive sentiment in equal measure were removed. The remaining sentences were randomly split between **T-1** and **TEST** sets, except for the 43 sentences out of the 50 double-annotated ones for which the annotators were in agreement. These sentences were all added to the **TEST** set. **T-1** contains 247 sentences. **TEST** contains 281 sentences; this is the set used for the blind testing reported in Table 4.

As a second step, in order to augment the training set, we utilized the data initially used for lexicon development. Recall that L-1 and L-2 had been created with a bias towards positive and negative polarity sentences (see section 2.1); this resulted in a significantly smaller proportion of NEU sentences in L-1 and L-2 (11%) than in the T-1 set (39%). To mitigate the risk of a significantly different category distribution between training and testing materials, we wanted to create a larger training set that matched the distribution of T-1. We implemented the following procedure to create **T-2**. We sampled data from L-1 and L-2 such as to match the category distributions in T-1. This resulted in the utilization of all NEU sentences in L-1 and L-2, and of only a small proportion of POS and NEG sentences in these sets. Adding the new items to T-1, we now have T-2 (482 sentences), doubling the amount of training data and retaining the T-1 distribution of categories.

In order to further expand the training data without changing its category distribution, we used the remaining POS and NEG annotated sentences in L-1 and L-2 and undertook the following procedure to collect more neutral sentences. Using a different essay pool with the same sub-genres of essays, we randomly sampled 1000 sentences with a condition that was complementary to the one used to produce L-1 and L-2, that is, with the condition that *none* of the sampled sentences match any word in Lexicon 0 (see section 2.1). This way, we obtained a higher percentage of NEU sentences in the sample. This new 1000 sentence set was submitted to AMT, and all sentences with a majority vote of NEU out of 3 AMT annotator labels were considered to be acceptable neutral sentences. We then added these NEU sentences,

along with the appropriate number of POS and NEG sentences from L-1 and L-2⁴ to maintain the category distribution of T-1 and T-2, and produced the final training set, **T-3**. Table 2 summarizes the sizes of all three training sets and of the test set.

Table 2. Sizes of the training and test sets for c5.0 experiments. The distribution of categories is the same in all training sets, 39% NEU, 35% POS, 26% NEG.

Dataset	# Sentences
T-1	247
T-2	482
T-3	1,631
TEST	281

3.3 Effective Coverage

Before moving to our evaluation that examines the performance of the family of lexicons on a sentence-level sentiment polarity classification task, we checked whether or not the expansion strategies actually succeeded in expanding the *effective coverage* of the data. Specifically, we examined whether the words added during expansion appear in our three training sets. This question is especially pertinent to our expansion strategies that used corpus-independent material (such as WordNet) or out-of-domain material, like the paraphrase generation tool. Table 3 shows the sizes of the lexicons as well as the number of different words from the lexicon that were in fact observed in the T-1, T-2, and T-3 datasets. Note that there is no guarantee that an observed word is a sentiment-bearing word; performance of each lexicon in the evaluation through sentiment classification as reported in the next section addresses this aspect of the expansion process.

Table 3. Sizes and effective sizes of the various subjectivity lexicons

Lexicon	# Words	#Words in T-1	#Words in T-2	#Words in T-3
Seed Lexicon	749	198	390	675
Raw WN	2,527	479	867	1,414
Raw Lin	1,907	292	563	981
Raw Para	2,994	585	1,028	1,679
Cleaned WN	1,495	280	541	936
Cleaned Lin	1,594	249	495	880
Cleaned Para	1,896	393	718	1,220
Raw WN + Raw Lin	967	232	457	788
Raw WN + Raw Para	1,158	326	601	973
Raw Lin + Raw Para	1,118	246	486	836
MPQA	6,450	244	504	1,014
GI	3,628	243	491	923

⁴ We added 160 POS sentences from the newly annotated 1,000 to T-3, in order to retain the distribution of categories the same as in T-1.

Table 3 shows that even the most conservative expansion (Raw WN + Raw Lin) is 29% bigger than the Seed Lexicon and has 17% more coverage than Seed Lexicon for the largest training set. We also note that both the manually- and the automatically-cleaned lexicons are on par with the state-of-art lexicons MPQA and GI in terms of effective coverage, even though they are at least 50% smaller in overall size.

3.4 Prediction of Sentiment Polarity

Table 4 below summarizes the accuracy of C5.0 classifier using counts of POS and NEG words as features, across various lexicons and the three cumulatively larger training sets with the same category distributions. All systems are evaluated on the TEST set. The majority baseline – 0.466 – corresponds to the proportion of NEU cases in TEST set. For the best system (Raw WN + Raw Para, T-3, accuracy of 0.548) the following are the Precision, Recall, and F-measures for the POS, NEG, and NEU categories: POS: P = 0.56, R = 0.44, F = 0.49; NEG: P = 0.45, R = 0.34, F = 0.39; NEU: P = 0.57, R = 0.72, F = 0.64.

Table 4. Accuracy of C5.0 classification of sentiment polarity. The best 4 runs for our lexicons are bold-faced and marked with an asterisk (*).

Lexicon	T-1	T-2	T-3
Majority Baseline	0.466	0.466	0.466
Seed Lexicon	0.520	0.512	0.512
Raw WN	0.448	0.473	0.456
Raw Lin	0.452	0.452	0.466
Raw Para	0.445	0.459	0.459
Cleaned WN	0.498	0.523	0.523
Cleaned Lin	0.537*	0.505	0.491
Cleaned Para	0.473	0.484	0.505
Raw WN + Raw Lin	0.544*	0.505	0.491
Raw WN + Raw Para	0.498	0.530	0.548*
Raw Lin + Raw Para	0.516	0.537*	0.484
MPQA	0.523	0.541	0.544
GI	0.512	0.530	0.491

4 Related Work

The research into subjectivity analysis can be clustered into three primary strands: (a) identification of words that are linked to subjectivity, (b) identification of subjectivity in sentences, clauses, phrases and words in a context – that is, subjective/objective, and positive/negative polarity classification, and (c) identification of subjectivity for applications, such as determining the sentiment orientation of reviews [10-12], of news headlines [13] and articles [1,14], or of blogs [15].

We are interested in exploiting subjectivity analysis for automated essay evaluation and scoring. Specifically, the target application would have the sentiment analysis

system working in tandem with a discourse analysis system [16], and allow, for example, identification of opinion orientation in the thesis statement of an issue essay, or determine the existence of both orientations in the summary statement of an essay written for the task of summarizing and evaluating contrasting opinions on a given subject. Our focus is, therefore, on sentence-level sentiment analysis, rather than phrase- or document-level, as in much of the current literature.

Identification of subjectivity in context has largely been left untouched in our work so far, and constitutes a major direction for future research. Approaches include the use of negation to alter the prior polarity of sentiment words, both grammatical negation (*not happy*) and content-word negations (*prevent further deterioration*), as well as of intensifiers (*extremely efficient* vs *somewhat efficient*), and identification of construction not representing a statement of belief, such as conditionals [17-20].

The step that concerned us most so far is the first step of compiling a comprehensive list of words with clear prior polarity ('prior' meaning before any contextual alterations occur, such as negation). Probably the broadest recent research initiative in this area is the work of Wiebe and colleagues. A concrete outcome of various annotation studies is the MPQA subjectivity lexicon, freely available for research: <http://www.cs.pitt.edu/mpqa/> [1,14,21]. This lexicon contains a list of words labeled with information about polarity (positive, neutral and negative), and the intensity of the polarity (strong or weak). More recent and detailed lexicons that include word sense information are also freely available (at the website above) and are extensions of this work [22,23]. Additional lexicons include the classic General Inquirer [2], the sense-based SentiWordNet [24], as well as custom-built lexicons with wide coverage such as in [19]. In the current work, our lexicons are compared to both MPQA and GI to provide a comparative performance and coverage evaluations.

The closest related work to our current project are studies dealing with expansions of subjectivity lexicons. The most popular source for expansion, also utilized in our work, is WordNet [24-31]. Additional resources include [32], symmetric syntactic patterns like conjunctions [33], as well as distributional information derived from a large corpus (12, 34, 35). In the latter setting, words are classified based on their distributional similarity to a small seed set of positive (negative) words; our approach in using Lin's distributional thesaurus for lexicon expansion is in a similar spirit.

Over the last decade, data-driven paraphrase generation has become an extremely active area in NLP. In particular, it has been used to improve several tasks such as query expansion in Information Retrieval [36-37], evaluation of NLP systems [38-40] and statistical machine translation [41,42]. A more comprehensive review of paraphrase generation and its applications can be found in [43]. To our knowledge, ours is the first reported attempt to use a paraphrase generation system for expanding a subjectivity lexicon.

In recent years, online crowdsourcing services utilizing a scalable, anonymous workforce have emerged as cost-effective means for collecting linguistic annotations. In particular, Amazon Mechanical Turk has become a popular resource for non-expert annotation of linguistic data for use in diverse NLP applications [44-47], including sentiment analysis [48-50]. While our test data was annotated by research assistants,

we elected to employ AMT at various stages of lexicon development and for generating additional training data.

5 Discussion and Conclusion

Based on Tables 2-4, the following points deserve mention.

- (a) The top four runs of the expanded lexicons (see Table 4) all outperform the best run for Seed Lexicon (0.520 vs 0.548, 0.544, 0.537). These results support cautious optimism regarding the chosen lexicon expansion strategies.
- (b) In the top 4 runs, one represents a lexicon that underwent manual AMT-based cleaning (Clean Lin), while the other three, including the top performer, were produced by automatically combining different expansions (Raw WN and Raw Lin, Raw WN and Raw Para, Raw Lin and Raw Para). There is therefore some evidence that effective use of the complementarity of automated expansion methods can produce results of comparable quality to human clean-up.
- (c) The best accuracy is obtained on a run with Raw WN and Raw Para, suggesting that a paraphrase generation system developed on out-of-domain data is effective for expansion of a subjectivity lexicon. To our knowledge, this is the first attempt to use a paraphrase generation system for this task, and it is shown to hold promise.
- (d) The top 4 performers are on par with the best run for the state-of-art MPQA lexicon (0.544) and perform better than the General Inquirer lexicon (0.530). We therefore showed it to be feasible to build a competitive subjectivity lexicon from scratch using steps described in this paper.
- (e) The better performance of combinations of raw lexicons (Raw Lin + Raw WN, etc) relative to individual raw lexicons suggests that the different expansion strategies are complementary to a certain degree. In future work, we will investigate possibilities for combining multiple lexicons.
- (f) The improvement with the size of the training data is not consistent. We suspect that the reason lies in the variability in our data. The data is sampled from a large pool of materials as diverse as argumentative essays on general topics and technical essays from accounting exams. Apparently, the fit in the category distribution is not sufficient for effective utilization of the training data, as additional training items might differ substantially from the T-1 and TEST data. We might need to fit a different model to every essay type, or use a much larger sample of sentences that represents all the different subtypes. The particulars of the sampling procedure might also matter; for example, the NEU sentences added in T-2 might be more difficult than those in T-1 and TEST, since they contain distracters – sentiment words from Lexicon 0. Consider “Could there be a reason for the participants of the study to say they are eating healthier?” that uses a generally positive term *healthier* versus “To end the inning you have to get three outs.”

References

1. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39(2-3), 165–210 (2005)
2. Stone, P., Dunphy, D., Smith, M., Ogilvie, D.: *The General Inquirer: A Computer Approach to Content Analysis*. MIT (1966)
3. Miller, G.: WordNet: A lexical database. *Communications of the ACM* 38(11), 39–41 (1995)
4. Lin, D.: Automatic retrieval and clustering of similar word. In: *Proceedings of the ACL, Montreal, Canada* (1998)
5. Madnani, N., Dorr, B.: Generating Targeted Paraphrases for Improved Translation. *ACM Transactions on Intelligent Systems and Technology* 3(2) (to appear)
6. Burstein, J., Pedersen, T.: Towards Improving Synonym Options in a Text Modification Application. University of Minnesota Supercomputing Institute Research Report Series, UMSI 2010/165 (2010)
7. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: *Proceedings of the ACL, Ann Arbor, MI*, pp. 597–604 (2005)
8. Koehn, P.: EUROPARL: A Parallel corpus for Statistical Machine Translation. In: *Proceedings of the Machine Translation Summit* (2005)
9. Quinlan, J.R.: *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers (1993)
10. Popescu, A., Etzioni, O.: Extracting product features and opinions from reviews. In: *Proceedings of HLT/EMNLP*, pp. 339–346 (2005)
11. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the ACL*, pp. 271–278 (2004)
12. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the ACL, Philadelphia*, pp. 417–424 (2002)
13. Strapparava, C., Mihalcea, R.: SemEval-2007 Task 14: Affective Text. In: *Proceedings of SemEval, Prague, Czech Republic* (2007)
14. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase Level Sentiment Analysis. In: *Proceedings of HLT-EMNLP* (2005)
15. Godbole, N., Srinivasaiah, M., Skiena, S.: Large Scale Sentiment Analysis for News and Blogs. In: *Proceedings of ICWSM* (2007)
16. Burstein, J., Marcu, D., Knight, K.: Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. In: Harabagiu, S., Ciravegna, F. (eds.) *Special Issue on Advances in NLP, IEEE Intelligent Systems*, vol. 18(1), pp. 32–39 (2003)
17. Choi, Y., Cardie, C.: Learning with Compositional Semantics as Structure Inference for Subsentential Sentiment Analysis. In: *Proceedings of EMNLP*, pp. 793–801 (2008)
18. Moilanen, K., Pulman, S.: Sentiment Composition. In: *Proceedings of Recent Advances in NLP (RANLP), Borovets, Bulgaria*, pp. 378–382 (September 2007)
19. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-Based Method for Sentiment Analysis. *Computational Linguistics* 37(2), 267–307 (2011)
20. Polanyi, L., Zaenen, A.: Contextual valence shifters. In: Wiebe (ed.) *Computing Attitude and Affect in Text: Theory and Applications*, pp. 1–10. Springer, Dordrecht (2006)
21. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: *Proceedings of EMNLP*, pp. 105–112 (2003)
22. Wiebe, J., Mihalcea, R.: Word Sense and Subjectivity. In: *Proceedings of ACL* (2006)
23. Gyamfi, Y., Wiebe, J., Mihalcea, R., Akkaya, C.: Integrating Knowledge for Subjectivity Sense Labeling. In: *Proceedings of NAACL* (2009)

24. Esuli, A., Sabastiani, F.: Determining term subjectivity and term orientation for opinion mining. In: Proceedings of the EACL (2006)
25. Kim, S., Hovy, E.: Determining the sentiment of opinions. In: Proceedings of COLING 2004 (2004)
26. Andreevskaia, A., Bergler, S.: Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In: Proceedings of the EACL (2006)
27. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
28. Kanayama, H., Nasukawa, T.: Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In: Proceedings of EMNLP (2006)
29. Strapparava, C., Valitutti, A.: WordNet-affect: and affective extension of Word-Net. In: Proceedings of LREC, Lisbon, Portugal (2004)
30. Kamps, J., Marx, M., Mokken, R., de Rijke, M.: Using WordNet to measure semantic orientation of adjectives. In: Proceedings of LREC (2004)
31. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientation of words using spin model. In: Proceedings of the ACL, pp. 133–140 (2005)
32. Mohammad, S., Dunne, C., Dorr, B.: Generating High-Coverage Semantic Orientation Lexicons from Overtly Marked Words and a Thesaurus. In: Proceedings of EMNLP (2009)
33. Hatzivassiloglou, V., McKeown, K.: Predicting the semantic orientation of adjectives. In: Proceedings of the ACL, pp. 174–181 (1997)
34. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems* 21(4), 315–346 (2003)
35. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of EMNLP, Morristown, NJ, pp. 129–136 (2003)
36. Metzler, D., Dumais, S., Meek, C.: Similarity Measures for Short Segments of Text. In: Amati, G., Carpineto, C., Romano, G. (eds.) *ECIR 2007*. LNCS, vol. 4425, pp. 16–27. Springer, Heidelberg (2007)
37. Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., Liu, Y.: Statistical machine translation for query expansion in answer retrieval. In: Proceedings of ACL (2007)
38. Zhou, L., Lin, C., Muntenu, D., Hovy, E.: ParaEval: Using paraphrases to evaluate summaries automatically. In: Proceedings of HLT-NAACL, pp. 447–454 (2006)
39. Owczarzak, K., Groves, D., van Genabith, J., Way, A.: Contextual bitext-derived paraphrases in automatic MT evaluation. In: Proceedings on the Workshop on Statistical Machine Translation, New York, pp. 86–93 (2006)
40. Kauchak, D., Barzilay, R.: Paraphrasing for automatic evaluation. In: Proceedings of HLT-NAACL, New York, pp. 455–462 (2006)
41. Callison-Burch, C., Koehn, P., Osborne, M.: Improved statistical machine translation using paraphrases. In: Proceedings of NAACL, New York, pp. 17–24 (2006)
42. Madnani, N., Resnik, P., Dorr, B., Schwartz, R.: Are multiple reference translations necessary? Investigating the value of paraphrased reference translations in parameter optimization. In: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas (AMTA), Waikiki, HI, pp. 143–152 (2008)
43. Madnani, N., Dorr, B.: Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics* 36(3), 341–387 (2010)
44. Snow, R., O'Connor, B., Jurafsky, D., Ng, Y.: Cheap and fast – but is it good?: Evaluating non-expert annotations for natural language tasks. In: Proceedings of EMNLP, pp. 254–263. Association for Computational Linguistics, Stroudsburg (2008)

45. Sheng, V., Provost, F., Ipeirotis, P.: Get another label? Improving data quality and data mining using multiple, noisy labelers. In: *Proceeding of KDD*, pp. 614–622 (2008)
46. Callison-Burch, C.: Fast, cheap, and creative: evaluating translation quality using Amazon’s Mechanical Turk. In: *Proceedings of EMNLP*, pp. 286–295 (2010)
47. Callison-Burch, C., Dredze, M.: Creating speech and language data with Amazon’s Mechanical Turk. In: *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pp. 1–12 (2010)
48. Akkaya, C., Conrad, A., Wiebe, J., Mihalcea, R.: Amazon Mechanical Turk for subjectivity word sense disambiguation. In: *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pp. 195–203 (2010)
49. Mellebeek, B., Benavent, F., Grivolla, J., Codina, J., Costa-juss, M., Banchs, R.: Opinion mining of Spanish customer comments with non-expert annotations on Mechanical Turk. In: *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pp. 114–121 (2010)
50. Mohammad, S., Turney, P.: Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In: *Proceedings of NAACL Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 26–34 (2010)

Emotion Ontology Construction from Chinese Knowledge

Peilin Jiang^{1,2}, Fei Wang¹, Fuji Ren², and Nanning Zheng¹

¹ Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
{pljiang, fwang, nnzheng}@aiar.xjtu.edu.cn

² Graduate School of Advanced Technology and Science, The University of Tokushima,
Tokushima, Japan
{jiang, ren}@is.tokushima-u.ac.jp

Abstract. To understand emotion and make machine emotion is one of the goals of affective computing. In order to recognize one's intention from the communication, both the meaning and the emotion are necessary to be interpreted correctly. But until now the study of fine-grained theory of emotion to describe inter-relationship of mental states is still full of challenges. In this paper, an emotion ontology from Chinese dictionary is semi-automatically created for human machine interaction. The proposed method of construction of emotion ontology includes affective word annotation and emotion predicate hierarchy extraction. Firstly, over 7,000 common affective words have been manually labeled as affective with their detailed explanations and been collected for an affective lexicon, then the consistent relationships in the affective lexicon are automatically parsed and a serial of emotion hierarchical structures are built up. More than 50 affective categories are extracted and about 5,000 nouns and adjectives, 2,000 verbs are categorized into the predicate hierarchy.

Keywords: Affective word annotation; affective clustering; complex emotion; emotion ontology; Chinese.

1 Introduction

Understanding the emotion behind the communication is essential to make a full comprehension of human's language. In human machine interaction, to make the machine understanding not only the meaning but also the emotion of the human is one of the goals of the affective computing. Recently a lot of researches were done for emotion detection, emotion classification, sentiment analysis and emotion generation of the subject through speech, dialog and text in different language [5][6][9][16][21]. However, an emotion ontology can be used to better understand the mental states and human emotions, especially the emotion carried by semantic roles in sentence. Currently, there are a lot of works on resources such as emotion dictionaries, corpora, and ontologies [1][15][17][18]etc.

In order to estimate and detect the emotion and manner in text, each single word in text has to been tagged with emotion tag. Most of annotation works have been

undertaken by skilled experts manually according to some accepted standardized rules or theories of emotion. Up to the present, Several theories of basic emotion for human being have been used respectively from psychoevolutionary theory, psychological experiments [7][8][18]. Practically, they are inadequate to represent the various kinds of subtle emotion in text. e.g., in the case of six basic emotions [7]: sadness, anger, happiness, surprise, fear and disgust, the affective words *sensitive*, *grandiose*, *fear* and *timid* cannot be well described by these basic emotions. Also the traditional sentiment analysis can only classify them into positive or negative but cannot distinguish the positive words *sensitive* and *grandiose* as well. There is not a generally accepted definition psychologically and practically for these cases. i.e., the fine-grained emotion theory is necessary to analyze the emotion in texts, also in applications of such as embodied computer agents, games and effective information retrieval.

To figure out what emotion represents in human brain may be a continuous challenge for cognitive science. But it is possible to extract what element is related to the emotion in communications, especially in text. This is because the human natural language is the carrier of knowledge of emotion and affective event by various kinds of affective words and expressions. It is simple for a person to tell if the word contains emotion or suggest him/her to image any affective event. To better understand the emotion in text, it is necessary to not only tag the emotion out but also the relation and density of emotions and mental states. Namely, a structure of emotion category that considers the lexical knowledge and emotion tag will be close to the nature of emotion. In this paper, we propose 'complex emotion' as the element to build the emotion structure that is obtained by an automatic clustering method from the affective words. AW (affective word) used in this paper denotes the word that is supposed to describe emotion state or affective event, e.g., AW denotes the word '*happy*' but not '*table*'. An emotion ontology including more than 7,000 nouns, adjectives, and verbs are constructed semi-automatically considering the lexical knowledge in Chinese text.

2 Background

At present many psychologists have claimed certain defined emotions as basic emotions by different reasons. e.g., psychological theory of primary emotions of [8] has 8 kinds of emotion related to adaptive biological processes, including acceptance, anger, anticipation, disgust, joy, fear, sadness and surprise. The most popular computational model for emotion synthesis, OCC model [19] defines 22 kinds of emotions and at last it was reduced into 12 emotions [20]: joy, hope, relief, pride, gratitude, love, distress, fear, disappointment, remorse, anger, and hate. Along with the increase of research on affective computing, these definitions have been introduced as the baseline into the engineering and furthermore, researchers extended them with more additional emotions. Some of current definitions of basic emotion [7] are shown in Table. 1.

Table 1. Samples of basic emotions. The emotions are defined in English and translated into Chinese.

Author	N	Basic Emotions (English Chinese)
Ekman	6	anger 愤怒, disgust 厌恶, fear 害怕, joy 高兴, sadness 悲哀, surprise 惊奇
Frijda	6	desire 渴望, happiness 快乐, interest 兴趣, surprise 惊奇, wonder 惊讶, sorrow 悲伤
Plutchik	8	acceptance 接受, anger 被怒, anticipation 希望, disgust 厌恶, joy 高兴, fear 害怕, sadness 悲哀, surprise 惊奇
Tomkins	9	anger 愤怒, interest 兴趣, contempt 轻视, disgust 厌恶, distress 悲痛, fear 害怕, joy 高兴, shame 害羞, surprise 惊奇
OCC model [20]	12	joy 高兴, hope 希望, relief 安慰, pride 自豪, gratitude 感谢, love 爱, distress 悲痛, fear 害怕, disappointment 失望, remorse 懊悔, anger 愤怒, hate 憎恨
Matsumoto[6]	22	joy 高兴, anticipation 希望, anger 愤怒, disgust 厌恶, sadness 悲哀, surprise 惊奇, fear 害怕, acceptance 接受, shy 害羞, pride 自豪, appreciate 感谢, calmness 平静, admire 钦佩, contempt 轻视, love 爱, happiness 快乐, exciting 兴奋, regret 后悔, ease 轻松, discomfort 不适, respect 尊重, like 喜爱

In Table.1 only anger, joy, fear, surprise, sadness, disgust can be found as basic emotion in 5 or all of 6 emotion theories. Therefore, it is hard to guarantee to the agreement of the affective word or idiom annotations. For example, people may have his own idea on what does emotion 'love' mean. Namely, a categorization based on the empirical language knowledge is supposed to objective in affective tagging over the manual annotation.

Many affective lexical resources make it possible to access the emotion, affective event or mental state conveniently. In English, WordNet-Affect [17] is an extension of WordNet-domains [18] that label synsets with representing affective concepts. In WordNet-Affect an additional hierarchy of affect domain label is used to annotate affective concepts. Lexical information including part-of-speech, synonyms, antonyms and definitions are manually added. An ontology of emotion cue [1] is introduced to especially modeling emotion cues. In Chinese, the most used lexical resource is HowNet[11]. HowNet is a Chinese network of lexical resource that contains conceptual and attribute relations among words. It only contains 6 sub-files for sentiment analysis such as: plus feeling, minus feeling, plus sentiment, minus sentiment, opinion and degree. Also HowNet has affective predicate hierarchy and in [14] a Chinese emotion ontology created based on extraction and manually annotation of affective predicate hierarchy from HowNet. All of these works need a time consuming manually annotation of emotion category and it is hardly to ensure the annotation agreement. In [22] an automatic method was proposed to extract paired affect classes in English. The intensity and centrality are used to measure the affect words.

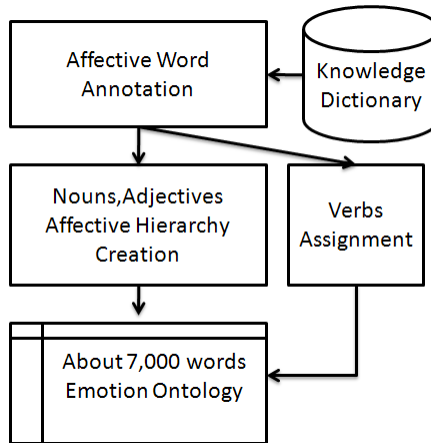


Fig. 1. The flow of emotion ontology construction from a Chinese dictionary resource

3 Emotion Ontology Construction

In this section, the construction of Chinese emotion ontology is introduced. Our construction flows of the emotion ontology creating method are illustrated in Fig. 1.

3.1 Affective Word Annotation

Normally each basic emotion can be represented by one affective word. But the boundary between them is hard to determine therefore the affective word classification is always done manually. On the contrary, a hierarchy of consistent affective words can preferably express the emotion state. We called the emotion hierarchy 'complex emotion' which can be explicated by more than one affective word and directly discerned in the text than the basic emotion represented by a single word. In order to label the affective word from a coarse resource, we used two skilled annotators to label out the affective word separately and consider the results together. For corpus, we select a kind of generally accepted resource, Chinese dictionary [2][4][10], Table.2 indicates some selected items from the resource. For some Chinese word there is no single English word that can translate the meanings but a phrase (e.g. second item in Table.2).

As know, generally dictionary does not indicate whether any word is affective or not. Therefore, for an affective word annotation, firstly we select the controlled source: including about 44,900 common words as the corpus. Then two skilled annotators are asked to assign the word whether it is related to the emotion state or not. The tag set uses only two tags, 'Emotional' and 'Null'. This make the annotation processing much easier than annotating what emotion it contains. The annotators are experienced researchers and tag the word depending on its word explanation independently. As such, totally 9,024 words have been tagged 'Emotional' by both annotators. Next some unsimplified Chinese words and dialects in them are removed

in advanced, therefore above 7,000 words are remained as the annotated affective corpus. The Table.2 indicates the samples of the common word items with POS (part-of-speech) tagged in their explanations and translation in English. For example, item '1,2,3,4' are assigned 'Emotional' and item '5,6' are assigned 'Null' respectively.

Subsequently we have also collected detailed explanations of these affective words. According to the dictionary of contemporary Chinese [2], we have firstly extracted 9,024 explanations (including definitions and examples). Then morphological analysis is processed to parse the explanations. Table.2 indicates the items of the common words with extracted affective words and POS in their explanations in English and Chinese.

Table 2. Samples of morphological analysis of source

N	Word (English Chinese pos)	Explanation and Definition
1	joy 高兴 a	愉快 a 而 c 兴奋 a 。 w (happy and exciting.)
2	(quiet and comfortable) 安逸 a	安静 a 而 c 舒适 a 。 w (quiet and comfortable.)
3	mourn 哀伤 a	悲伤 a 。 w(sorrow.)
4	a plaintive whine 哀鸣 v	悲哀 a 地 u 呼叫 v 。 w (shout sadly.)
5	corner 街角 n	街巷 n 等 n 的 d 拐角处 n 。 w (corner on the street and alleys road.)
6	paper 论文 n	讨论 v 某个 d 问题 n 或者 c 研究 v 某个 d 问题 n 的 d 文章 n 。 w (an article that discusses some issues.)

Recursively we gathered about 7,200 available AWs and their explanations in our lexicon because some unsimplified Chinese words and dialects are removed in advanced. Then, we defined two types of relationship in AWs explanation, consistent and inconsistent, respectively. Consistent relation means to be similar polarity in affective property and inconsistent relation shows opposite attitude in affective property correspondingly. e.g., in Table.2, 'joy' is consistent with the 'happy', and negative word often represents inconsistent relation, e.g., 'safela: not|n danger|a'. It is clear that 85% explanations are composed by consistent relation. i.e., most of the affective words are coherent with the affective words in their explanations.

3.2 Affective Hierarchy Extraction

To obtain the affective hierarchy we used a hierarchical clustering algorithm depending on lexical knowledge to gather the distribution of affective relationship. The algorithm can be presented as follows:

1. Assign each affective word to cluster C_i , $1 \leq i \leq N$, N is the total number of affective words,

2. Calculate the distance $d_{i,j}$ between C_i and C_j , $i, j \in (1, \dots, N)$
3. Assume $d_{i,j} = 1$ if consistent,
4. Assign consistent C_i, C_j into C_k , where $k = \min(i, j)$.
5. Repeat the previous process until all consistent relations are calculated.

The 'consistent' indicates that the emotion of the word is consistent with the emotion of the semantic roles of the affective word in the explanation. In order to gather all 'same' emotion into one complex emotion, a lexical affective clustering method is proposed. A set of POS rules is defined as consistent rules. Table 3 indicates part of the rules.

Table 3. Samples of POS rules

POS Rule	Sample Definitions
a < aca	joyl高兴la (child node):愉快la 而lc 兴奋la 。lw (happy and exciting.)
v < aduv	a plaintive whine l哀鸣lv (child node): 悲哀lad地lu呼叫lv。lw(shout sadly.)
a < a	mourn l哀伤la (child node): 悲伤la 。lw(sorrow)

However, in this step, more than 5,000 nouns and adjectives are clustered, and the word (e.g. *joy* in the Table 3.) is the child node of the affective words (e.g. *happy* and *exciting*) in the right side and this makes a hierarchical structure for all the affective words categories. Fig 2 indicates a part of affective words in one emotion category 'suffering'.

suffering (痛苦): { experiencer (suffering), degree (1), cause (null)}
grief (难过): { parent (suffering), experiencer (suffering), degree (2), cause (null)}
sorrow (悲伤): { parent (grief), experiencer (suffering), degree (3), cause (null)}
woeful (悲郁): { parent (sorrow, gloom), experiencer (suffering), degree (3), cause (null)}
mourn (哀伤): { parent (sorrow), experiencer (suffering), degree (3), cause (null)}
plaintive (哀怨): { parent (sorrow), experiencer (suffering), degree (3), cause (null)}

Fig. 2. Samples of nouns and adjectives in one emotion hierarchy. The degree role is assigned with the depth in the hierarchy. The child node inherits the emotion tag of the semantic role of subject from the parent node. In this example, the experience role inherits the emotion tag from their parent node as 'suffering'.

Fig 3 shows the hierarchical structure of complex emotion. And the distance is basically decided by the shortest path between the words and the depth of hierarchical structure [12][13]. In case of circle structure, the distance between the words on the circle is 0. Finally, we have clustered 52 complex emotion categories to classify the

affective hierarchy for the next assignment step. Meanwhile, the semantic roles of the hierarchy are also assigned emotions referring to clustering results to create emotion category.

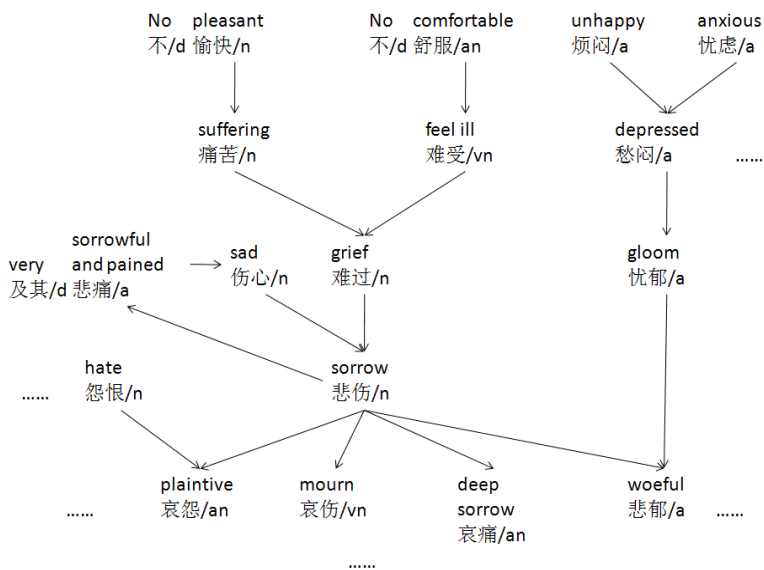


Fig. 3. Sample of part of 'suffering' complex emotion structure

welcome (欢迎/v) to meet someone very **pleasantly**. (很/d 高兴/ad 地/u 迎接/v。/w)

bring benefit to (造福/v) bring someone **benefit** (给/p 人/n 带来/v 幸福/a。/w)

bemoan (哀叹/v) **sorrowfully** sigh (悲哀/a 地/u 叹息/v。/w)

mourn (哀悼/v) **sorrowfully** remember (悲痛/a 地/u 追念/v。/w)

scowl at (怒视/v) look **angrily** at (愤怒/a 地/u 注视/v。/w)

wonder at (惊叹/v) surprisingly **praise** (惊讶/a 赞叹/vn。/w)

love (喜爱/v) **be fond/interested** of someone or something. (对/p 人/n 或/c 事物/n 有/v 好感/n 或/c 感到/v 兴趣/n。/w)

praise (称赞/v) to express **favorite** manner to someone or something's **merit**. (用言语/n 表达/v 对/p 人/n 或/c 事物/n 的/u 优点/n 的/u 喜爱/vn。/w)

Fig. 4. Samples of affective verbs. The bold word indicates the subject of emotion predicate.

3.3 Affective Verb Assignment

To create emotion ontology, in the last step about 2,000 affective verbs are filled into the emotion predicate hierarchy. The detailed explanation of each affective verb is analyzed and the word is assigned to the emotion predicate which is appeared in the verb's explanation. The subject semantic role of the verb inherits the emotion tag from the subject of the emotion predicate in the explanation. For example, Fig 4 indicates the affective verbs that have emotion predicate in their explanations.

4 Evaluation

In the step of manually annotation of affective words from controlled source, the internal agreement is evaluated by the kappa value [3] equals to 0.60. The kappa values demonstrate the promising inter-agreement of the affective word annotation. In Ren-CECps [15], there is an evaluation criteria to judge if the word contains emotions, the kappa is over 80%. This is because our word set (44,900 different words) contains much more words without duplication than them (26 documents, 805 sentences and 19,738 words).

Then to demonstrate the complex emotion category, we have required two experienced raters in linguistics to do the evaluation experiment. Two metrics are used:

(1) Emotion consistency test. Two human-center experiments are conducted as follows: (a) Similarity Test: 5 words randomly extracted from one complex emotion category are assigned as 'A' group, and then another 5 words composed 'B' group, in which one word are from the same category as the 'A' group and the others from the different categories randomly. raters are asked to select the most similar AW to 'A' from 'B' group; (b) Difference Test: 'C' group with 4 AWs extracted from the same category and 1 from the other category, then the raters are asked to pick out that most different one in 'C' group.

Participants were required to repeat each experiment 20 times and in this way the average accuracy rates are (a) 80.5% and (b) 83.5% respectively.

(2) Kappa coefficient agreement. Two raters are asked to mark out the word belonging to the category or not. For each category we can get one kappa value. We conducted every cluster and got the average kappa value for the clustered result kappa value equals to 0.63.

Finally for the proposed Chinese emotion ontology, also, 2 skilled human raters are asked sign 'yes' or 'no' for the previous result in the hierarchy. The average kappa value equals to 0.62.

5 Conclusion and Future Work

At the present stage, we have labeled and analyzed about 7,000 affective nouns, adjectives and verbs with their detailed explanations and we have automatically summarized 52 complex emotion categories that appear in Chinese text. Accordingly, we build up a Chinese emotion ontology that contains affective nouns, adjectives and

verbs with hierarchical structure considering the lexical knowledge features such as emotion tag, part-of-speech. Synonyms and antonyms are not included because the hierarchical structure extracted from the explanation ensures that the similar words will have small conceptual distance. The evaluation experiment for internal agreement was conducted and the results have been evaluated to be valid.

In the future, we intend to extend the emotion ontology from much more new language resources such as Information extracted from Internet by a web crawler, and verify our emotion ontology on emotion recognition in sentence in applications such as question answer (QA) system, virtual character embodied on multimedia etc.

Acknowledgments. The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (A), 22240021, and Grant-in-Aid for Challenging Exploratory Research, 21650030.

References

1. Obrenovic, Z., Garay, N., López, J.M., Fajardo, I., Cearreta, I.: An Ontology for Description of Emotional Cues. In: Tao, J., Tan, T., Picard, R.W. (eds.) ACII 2005. LNCS, vol. 3784, pp. 505–512. Springer, Heidelberg (2005)
2. Lv, S.: Dictionary of contemporary Chinese. The Commercial Press, Beijing (1996)
3. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
4. Guo, X., Chang, Y.: Changyong BaoBianYi XiangJie Dictionary. The Commercial Press, Beijing (1996)
5. Jaap, K., Marx, M., Mokken, R.J., de John, M.R.: Using WordNet to Measure Semantic Orientations of Adjectives. In: Proceeding of the 4th International Conference on Language Resources and Evaluation, pp. 1115–1118 (2004)
6. Matsumoto, K., Mishina, K., Ren, F., Kuroiwa, S.: Emotion Estimation Algorithm based on Emotion Occurrence Sentence Pattern. *Journal of Natural Language Processing* 14(3), 239–271 (2007)
7. Ortony, A., Turner, T.J.: What's basic about basic emotions? *Psychological Review* 97, 315–331 (1990)
8. Plutchik, R.: The Multifactor-Analytic Theory of Emotion. *The Journal of Psychology* 50, 153–171 (1960)
9. Turney, P.D.: Thumb up? Thumb down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417–424 (2002)
10. Yu, S., Zhu, X., et al.: The Grammatical Knowledgebase of Contemporary Chinese. Tsinghua University Press, Beijing (1998)
11. Dong, Z., Dong, Q.: Introduction to hownet. In: HowNet (2010), <http://www.keenage.com>
12. Liu, Q., Li, S.: Word Similarity Computing Based on How-net. *Computational Linguistics and Chinese Language Processing* 7(2), 59–76 (2002)

13. Agirre, E., Rigau, G.: A proposal for word sense disambiguation using conceptual distance. In: Proc. of International Conference Recent Advance in Natural Language Processing (RANLP), Tzigrav Chark, Bulgaria, pp. 258–264 (1995)
14. Yan, J., Bracewell, D.B., Ren, F., Kuroiwa, S.: The Creation of a Chinese Emotion Ontology Based on HowNet. *Engineering Letters* 16(1), 166–171 (2008)
15. Quan, C., Ren, F.: Construction of a Blog Emotion Corpus for Chinese Emotional Expression Analysis. In: EMNLP 2009, pp. 1446–1454 (2009)
16. Quan, C., Ren, F.: An Exploration of Features for Recognizing Word Emotion. In: COLING 2010, pp. 922–930 (2010)
17. Strapparava, C., Valitutti, A.: Wordnet-affect: An Affective Extension of WordNet. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, pp. 1083–1086 (2004)
18. Fellbaum, C. (ed.): *WordNet: An Electronic Lexical Database*. The MIT Press (1998)
19. Ortony, A., Clore, G., Collins, A.: *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge (1988)
20. Ortony, A.: On Making Believable Emotional Agents Believable. In: Trappl, R., et al. (eds.) *Emotions in Humans and Artifacts*, pp. 189–212. MIT Press (2003)
21. Kasap, Z., Moussa, M.B., Chaudhuri, P., Magnenat-Thalmann, N.: Making Them Remember—Emotional Virtual Characters with Memory. *IEEE Computer Graphics and Applications* 29(2), 20–29 (2009)
22. Grefenstette, G., Qu, Y., et al.: Validating the Coverage of Lexical Resources for Affect Analysis and Automatically Classifying New Words along Semantic Axes. In: *Exploring Attitude and Affect in Text: Theories and Applications*. AAAI-2004 Spring Symposium Series, pp. 71–78 (2004)

Appendix

Table 4. Complex emotion category in Chinese. The number, polarity and sample words of complex emotion category are indicated.

Complex Emotion		Complex Emotion	
1	- 怨恨 n, 忧伤 n, 凄凉 a, 愤怒 n	27	+ 振奋 a, 奋发 a
2	- 受窘 n, 窘迫 n, 苦寒 n	28	- 蛮横 a, 凶猛 a, 骁勇 a, 残忍 a
3	- 无拘无束 a, 毫无顾忌 a, 散漫 a	29	- 老辣 a, 恶毒 n, 强横 n, 凶狠 n
4	+ 爽朗 a, 欢畅 a, 宽畅 a	30	+ 舒心 n, 可心 n
5	+ 决断 a, 英勇 a, 果决 a, 勇敢 a	31	+ 关心 n, 信任 n
6	+ 自如 a, 镇定 a, 体面 a	32	+ 友爱 n, 亲近 a, 热爱 n
7	慷慨 a, 粗暴 a, 暴躁 a, 刚烈 a, 鲁莽 a	33	- 癖好 n, 离奇 a, 古怪 a
8	+ 敦厚 a, 缓和 a, 温和 a	34	+ 豁朗 a, 热忱 a
9	撒娇 v, 无赖 n, 风流 n	35	+ 刚直 a, 正直 a
10	+ 安宁 n, 平静 n, 幸福 n, 和平 n, 快乐 a	36	- 懈怠 a, 懒惰 a
11	荒谬 a, 希奇 a, 巧妙 a, 奇怪 a, 惊异 v	37	- 迷糊 a, 糊涂 a, 冷漠 a
12	+ 善于应变 a, 了解 v, 聪敏 a, 内行 a	38	匆忙 a, 紧急 a
13	- 心虚 v, 害羞 a, 谦恭 a, 不安 a	39	- 颓废 n, 灰心 n, 荒废 v
14	+ 灵敏 a, 聪明 a	40	- 沉痛 n, 窝火 n
15	+ 宽阔 a, 豪壮 a	41	- 傲然 a, 孤傲 a
16	- 懦弱 a, 怕事 a, 窝囊 a, 胆小 a	42	- 孤寂 n, 寂寞 n
17	- 顾虑 v, 留神 v	43	+ 天真 a, 高雅 a, 高贵 a
18	- 蠢笨 a, 愚蒙 a	44	- 阔绰 a, 浪费 v
19	+ 淳朴 a, 厚道 a	45	- 穷困 n, 尴尬 n
20	+ 良善 a, 和善 a	46	- 心焦 n, 焦虑 v
21	+ 爽直 a, 豪放 a	47	+ 颖慧 a, 聪慧 a
22	- 烦杂 v, 吵闹 v	48	- 卑劣 a, 卑怯 a, 齷齪 a
23	幽静 a, 清静 a	49	+ 欢乐 a, 喜悦 n, 可喜 n, 愉悦 a, 高兴 a
24	+ 精通 v, 懂事 a	50	- 厌恶 n, 痛恨 v, 恶心 v
25	- 守旧 a, 死板 a	51	+ 敬重 v, 珍视 v
26	+ 通畅 a, 恬淡 a	52	- 清苦 a, 刻苦 a

Table 5. Complex emotion category (reference translation in English of Table. 4) The number, polarity and sample words of complex emotion category are indicated

Complex Emotion		Complex Emotion	
1	- resentment, grief, misery, anger	27	+ inspired, work hard
2	- constraints, distress, embarrass	28	- rude, violent, tyrannical, cruel
3	- unrestricted, cynical, lack discipline	29	- crafty, vicious, malicious
4	+ hearty, delight, spacious	30	+ comfort, fancy
5	+ determination, valiant, resolute, courageous	31	+ concern, trust, acceptance
6	+ smooth, calm, decent	32	+ friendliness, close, loving
7	generous, rude, impatient, irascible, criticized	33	- habit, bizarre, weird
8	+ honest and sincere, relaxed, and moderate	34	+ illumination, enthusiasm
9	pout, rogue, dissolute	35	+ integrity, honestly
10	+ tranquility, serenity, happiness, peace, merry	36	- laxity, laziness
11	absurd, amazed, ingenuity, strange, surprising	37	- confused, muddled, apathy
12	+ adaptable, understand, sagacity, experienced	38	hurry, emergency
13	- diffidence, shy, humility, anxiety	39	- decadent, frustration, abandoned
14	+ sensitive, smart	40	- resentful, irritated
15	+ wide, grandiose	41	- haughtily, egocentric
16	- cowardly, soft, fear, timid	42	- lonesome, lonely
17	- scruple, pay attention	43	+ naive, elegant, noble
18	- clumsy, naive	44	- lavish, waste
19	+ simplicity, guileless	45	- poverty, embarrassment
20	+ kindness, good	46	- anxiety, panic
21	+ chattiness, uninhibited	47	+ bright, intelligence
22	- trivial, noisy	48	- despicable, dastardly, dirty
23	loneness, quiet	49	+ joy, good, happy
24	+ proficient, thoughtful	50	- obnoxious, hate, disgust
25	- conservative, rigid	51	+ respected, cherished
26	+ patency, light mood	52	- poor, hard

Author Index

- Abbas, Qaiser I-66
Abdallah, Sherief I-311
Agichtein, Eugene II-318
Ahmed, Umair Z. I-415
Alberti, Gábor I-349
Alfared, Ramadan I-104
Allende, Héctor II-438
Annesi, Paolo I-323
Arnulphy, Béatrice II-219
Arppe, Antti II-1
Artiles, Javier II-194
Ashraf, Md. Izhar I-142
Askarian, Narjes I-201
Athiappan, G. II-378
- Bajwa, Imran Sarwar I-178
Bali, Kalika I-415
Ballesteros, Miguel I-363
Bandypadhyay, Sivaji I-117, I-513, I-540
Bangalore, Srinivas I-1
Banu, W. Aisha II-274
Barrera, Araly II-366
Basili, Roberto I-323, I-336
Basu, Anupam I-211
Béchet, Denis I-104
Béchet, Nicolas I-154
Beigman Klebanov, Beata I-591
Benis, Nirupama I-54
Bhattacharyya, Pushpak I-92, I-475
Bhattarai, Archana I-568
Bordbar, Behzad I-178
Burkett, Candice II-502
Burstein, Jill I-591
- Cabré, M. Teresa I-462
Cao, Hailong II-52
Carreras-Riudavets, Francisco J. I-80
Carroll, John II-232
Cassidy, Taylor II-194
Castillo, Carlos II-181
Castillo, Mauro I-17
Cellier, Peggy I-154, I-166
- Chakraborti, Sutanu II-462
Charnois, Thierry I-154, I-166
Choi, Yoonjung I-500
Choudhury, Monojit I-415
Choudhury, Sanjay Kumar II-462
Climent, Salvador II-110
Compton, Paul II-414
Crémilleux, Bruno I-154
Croce, Danilo I-336
- da Cunha, Iria I-462
Daille, Béatrice II-72, II-169
Dalbello Bašić, Bojana I-428
Das, Amitava I-540
Das, Dipankar I-513
De Belder, Jan II-426
Descoins, Alan II-206
Deshpande, Shailesh II-378
Devi, Laishram Martina I-117
Díaz, Alberto I-363
Dinu, Liviu P. I-556
Domínguez García, Renato I-42
Dutta, Sudakshina I-211
- Ekbal, Asif I-513
- Faili, Hessaam I-526
Faulkner, Adam I-591
Fazly, Afsaneh I-201
Fei, Geli II-144
Filice, Simone I-336
Finch, Andrew II-122
Francisco, Virginia I-363
Fresno, Víctor II-157
Fu, Guohong I-580
- Galgani, Filippo II-414
Galicia-Haro, Sofia N. I-402
Gambäck, Björn I-540
Gelbukh, Alexander I-402
Gervás, Pablo I-363
González, Aitor I-17
Graesser, Arthur II-502
Gutiérrez, Yoan I-225

- Hao, Tianyong II-318
 Haralick, Robert M. II-247
 Harastani, Rima II-72
 Harkous, Hamza I-297
 Hazem, Amir II-83
 Herath, Dulip Lakmal I-188
 Hernández-Figueroa, Zenón I-80
 Herrera, Jesús I-363
 Hoffmann, Achim II-414
 Huang, Minhua II-247
 Hyvärinen, Mirka II-478
- Ittycheriah, Abraham II-25
 Iuga, Iulia I-556
- Jacquín, Christine II-169
 Janicki, Maciej I-258
 Ji, Heng II-194
 Jiang, Peilin I-603
 Jordão, Carlos C. II-297
- Kaliyaperumal, Rajaram I-54
 Károly, Márton I-349
 Kato, Tsuneo II-306
 Keisam, Napoleon I-117
 Keshtkar, Fazel II-502
 Klabunde, Ralf II-13
 Koeling, Rob II-232
 Kolya, Anup Kumar I-513
 Kuboň, Vladislav I-130
 Kumar, Arpit I-415
 Kumar, Niraj II-353, II-390
 Kundu, Bibekananda II-462
 Kvist, Maria II-450
- Lalitha Devi, Sobha I-285
 Lee, Mark I-178
 Legallois, Dominique I-166
 Li, Haiying II-502
 Li, Qi II-194
 Li, Sheng II-52
 Lin, King-Ip I-568
 Lindén, Krister II-478
 Liyanage, Chamila I-188
 Lopatková, Markéta I-130
- Madnani, Nitin I-591
 Mahendran, Anand II-490
 Makhoul, Jad I-297
 Malcuori, Marisa II-206
- Marcinićzuk, Michał I-258
 Martínez, Raquel II-157
 Materna, Jiří I-376
 Mathur, Prashant II-60
 Mavaluru, Dinesh II-274
 Mendoza, Marcelo II-438
 Meng, Yao I-249
 Moens, Marie-Francine II-426
 Moncecchi, Guillermo II-206
 Montoyo, Andrés I-225
 Morin, Emmanuel II-72, II-83
 Mukherjee, Subhabrata I-475
 Myaeng, Sung-Hyon I-500
- Nguyen, Kiem-Hieu I-238
 Nguyen, Le Minh I-438
 Niraula, Nobal I-450, I-568
 Nongmeikapam, Kishorjit I-117
- Oakes, Michael II-132
 Ock, Cheol-Young I-238
 Oh, Hyo-Jung I-500
 Okita, Tsuyoshi II-40
 Oliver, Antoni II-110
 Ovchinnikova, Ekaterina I-388
- Palshikar, Girish Keshav II-378
 Pascual, Fernando Llopis II-342
 Paukkeri, Mari-Sanna II-1
 Paul, Soma II-60
 Peñas, Anselmo I-388
 Peregrino, Fernando S. II-342
 Pérez García-Plaza, Alberto II-157
 Petran, Florian II-97
 Pham, Quang Nhat Minh I-438
 Plátek, Martin I-130
 Ponomareva, Natalia I-488
 Popescu, Octavian I-270
 Puri, Shivani II-232
 Pushpananda, Randil I-188
- Quiniou, Solen I-166
- Rajadesingan, Ashwin II-490
 Ram, R. Vijay Sundar I-285
 Ramos, José Guadalupe II-181
 Ren, Fuji I-603
 Rensing, Christoph I-42
 Reuße, Sebastian II-13
 Rigau, German I-17

- Rodríguez-del-Pino, Juan C. I-80
 Rodríguez-Rodríguez, Gustavo I-80
 Rosá, Aiala II-206
 Rosa, João Luís G. II-297
 Roukos, Salim II-25
 Rus, Vasile I-450, I-568
- Sadh, Ashish II-402
 Saffar, Mohammadtaghi I-526
 Sahu, Amit II-402
 Salehi, Bahar I-201
 SanJuan, Eric I-462
 Sanyal, Ratna II-402
 Sanyal, Sudip II-402
 Sanz, Luis II-438
 Schlünder, Björn II-13
 Schmidt, Sebastian I-42
 Shaalan, Khaled I-311
 Shakery, Azadeh I-526
 Shams, Mohammadreza I-526
 Sharma, Aribam Umananda I-117
 Shimazu, Akira I-438
 Shoaib, Muhammad I-311
 Shrestha, Prajol II-169
 Shriram, R. II-274
 Sierra, Gerardo I-462
 Silfverberg, Miikka II-478
 Šilić, Artur I-428
 Silva, Josep II-181
 Singh, Khangengbam Dilip I-117
 Sinha, Sitabhra I-142
 Srinathan, Kannan II-353, II-390
 Srinivasarao, Vundavalli II-286
 Srivastava, Devesh II-402
 Steinmetz, Ralf I-42
 Storch, Valerio I-323
 Sumita, Eiichiro II-52, II-122
- Tan, He I-54
 Tannier, Xavier II-219
- Tetreault, Joel I-591
 Thelwall, Mike I-488
 Tomás, David II-342
 Torres-Moreno, Juan-Manuel I-462
 Tripathi, Nandita II-132
- Valero, Héctor II-181
 van Genabith, Josef II-40
 Varma, Vasudeva II-286, II-353, II-390
 Vasudevan, N. I-92
 Väyrynen, Jaakko II-1
 Vázquez, Sonia I-225
 Velupillai, Sumithra II-450
 Verma, Rakesh II-366
 Vilnat, Anne II-219
- Walas, Marcin II-330
 Wang, Fei I-603, II-261
 Wang, Xin I-580
 Weerasinghe, Ruwan I-188
 Wermter, Stefan II-132
 Wonsever, Dina II-206
 Wu, Jianwei I-249
 Wu, Yunfang II-261
- Xia, Yingju I-249
 Xu, Jian-Ming II-25
 Xu, Xin II-306
- Yasuda, Keiji II-122
 Yıldırım, Savaş I-29
 Yıldız, Tuğba I-29
 Yu, Hao I-249
- Zanoli, Roberto I-270
 Zaraket, Fadi I-297
 Zhang, Chenghe II-144
 Zhang, Shu I-249
 Zhao, Tiejun II-52, II-144
 Zheng, Dequan I-249, II-144
 Zheng, Nanning I-603