

Alexander Gelbukh (Ed.)

LNCS 7182

# Computational Linguistics and Intelligent Text Processing

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

2  
Part II



 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 II

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](mailto:gelbukh@gelbukh.com)

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-28600-1

e-ISBN 978-3-642-28601-8

DOI 10.1007/978-3-642-28601-8

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](http://www.springer.com))

# Preface

CICLing 2012 was the 13<sup>th</sup> 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 <sup>1</sup>			Country or region	Authors			Papers <sup>1</sup>			
	Subm.	Subm.	Accp.	Subm.	Subm.	Accp.		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

<sup>1</sup> 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](http://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 topic<sup>2</sup>

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

<sup>2</sup> 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](http://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):

- 1<sup>st</sup> Place: *Automated Detection of Local Coherence in Short Argumentative Essays Based on Centering Theory*, by Vasile Rus and Nobal Ni- raula, USA;
- 2<sup>nd</sup> Place: *Corpus-Driven Hyponym Acquisition for Turkish Language*, by Savaş Yıldırım and Tuğba Yıldız, Turkey;
- 3<sup>rd</sup> 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](http://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	German Rigau
Sun Maosong	Fabio Rinaldi
Yuji Matsumoto	Horacio Rodriguez
Diana McCarthy	Vasile Rus
Helen Meng	Horacio Saggion
Rada Mihalcea	Kepa Sarasola
Ruslan Mitkov	Serge Sharoff
Dunja Mladenic	Grigori Sidorov
Marie-Francine Moens	Thamar Solorio
Masaki Murata	John Sowa
Vivi Nastase	Ralf Steinberger
Roberto Navigli	Vera Lúcia Strube De Lima
Kjetil Nørvåg	Tomek Strzalkowski
Constantin Orăsan	Jun Suzuki
Patrick Saint-Dizier	Christoph Tillmann
Maria Teresa Pazienza	George Tsatsaronis
Ted Pedersen	Junichi Tsujii
Viktor Pekar	Dan Tufiș
Anselmo Peñas	Hans Uszkoreit
Stelios Piperidis	Felisa Verdejo
Irina Prodanof	Manuel Vilares Ferro
Aarne Ranta	Haifeng Wang
Victor Raskin	Bonnie Webber
Fuji Ren	

## Software Reviewing Committee

Ted Pedersen	Sergio Jiménez Vargas
Florian Holz	Miikka Silfverberg
Miloš Jakubíček	Ronald Winnemöller

## Award Committee

Alexander Gelbukh	Ted Pedersen
Eduard Hovy	Yorick Wiks
Rada Mihalcea	

## Additional Referees

Adrián Blanco González	Ana Garcia-Serrano
Ahmad Emami	Ananthakrishnan Ramanathan
Akinori Fujino	Andrej Gardon
Alexandra Balahur	Aniruddha Ghosh
Alvaro Rodrigo	Antoni Oliver
Amitava Das	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 Ashegh  
Oana Frunza  
Oier Lopez De Lacalle  
Olga Kolesnikova  
Omar Alonso  
Paolo Annesi  
Peter Ljunglöf  
Pinaki Bhaskar  
Prokopis Prokopidis

Rainer Winnenburg	Ting Liu
Ramona Enache	Tom De Smedt
Raquel Martínez	Tommaso Caselli
Richard Forsyth	Tong Wang
Robin Cooper	Toshiyuki Kanamaru
Rodrigo Agerri	Tsutomu Hirao
Roser Morante	Ulf Hermjakob
Ryo Otoguro	Upendra Sapkota
Samira Shaikh	Vanessa Murdock
Santanu Pal	Victor Darriba
Shamima Mithun	Víctor Peinado
Sharon Small	Vít Baisa
Simon Mille	Vojtech Kovar
Simone Paolo Ponzetto	Wilker Aziz
Siva Reddy	Yulia Ledeneva
Somnath Banerjee	Yvonne Skalban
Tadej Štajner	Zuzana Neverilova
Thierry Declerck	

## Website and Contact

The webpage of the CICLing conference series is [www.CICLing.org](http://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 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 Jacquin, 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, Sutantu 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>	

## Erratum

Neoclassical Compound Alignments from Comparable Corpora . . . . .	E1
<i>Rima Harastani, Béatrice Daille, and Emmanuel Morin</i>	
<b>Author Index . . . . .</b>	515

# 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-Riuadavets, 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 Ruvan Weerasinghe</i>	
Automatic Identification of Persian Light Verb Constructions .....	201
<i>Bahar Salehi, Narjes Askarian, and Afsaneh Fazly</i>	
<b>Word Sense Disambiguation and Named Entity Recognition</b>	
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 Disambiguation . . . . .	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 Marciničzuk and Maciej Janicki</i>	
Methods of Estimating the Number of Clusters for Person Cross Document Coreference Task . . . . .	270
<i>Octavian Popescu and Roberto Zanoli</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 Makhoulta, 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>	
<b>Semantics and Discourse</b>	
Space Projections as Distributional Models for Semantic Composition . . . . .	323
<i>Paolo Annese, Valerio Storch, and Roberto Basili</i>	
Distributional Models and Lexical Semantics in Convolution Kernels . . .	336
<i>Danilo 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 Nobile 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>	
<b>Sentiment Analysis, Opinion Mining, and Emotions</b>	
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 Heshaam 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>	
<b>Author Index .....</b>	<b>615</b>

# Exploring Extensive Linguistic Feature Sets in Near-Synonym Lexical Choice

Mari-Sanna Paukkeri<sup>1</sup>, Jaakko Väyrynen<sup>1</sup>, and Antti Arppe<sup>2</sup>

<sup>1</sup> Aalto University School of Science, P.O. Box 15400, FI-00076 Aalto, Finland

<sup>2</sup> University of Helsinki, Unioninkatu 40 A, FI-00014 University of Helsinki, Finland

**Abstract.** In the near-synonym lexical choice task, the best alternative out of a set of near-synonyms is selected to fill a lexical gap in a text. We experiment on an approach of an extensive set, over 650, linguistic features to represent the context of a word, and a range of machine learning approaches in the lexical choice task. We extend previous work by experimenting with unsupervised and semi-supervised methods, and use automatic feature selection to cope with the problems arising from the rich feature set. It is natural to think that linguistic analysis of the word context would yield almost perfect performance in the task but we show that too many features, even linguistic, introduce noise and make the task difficult for unsupervised and semi-supervised methods. We also show that purely syntactic features play the biggest role in the performance, but also certain semantic and morphological features are needed.

**Keywords:** Near-synonym lexical choice, linguistic features.

## 1 Introduction

In the *lexical choice* task, gaps in a text are filled with words that best fit the context. Lexical choice is needed in many natural language generation (NLG) applications: for example, in machine translation, question-answering, summarisation, text simplification, and adapting terminology so that it can be understood by a user. It can also help produce more readable language and expand the limits of bilingual dictionaries by taking the context better into account. Further, a second-language student or translator would benefit from an application which could help write text in a foreign language by suggesting appropriate alternatives to words. Lexical choice is a very difficult problem within a set of near-synonyms due to fine-grained differences between the words. Some methods have been proposed for the problem in the literature [7,23].

In this paper, we use extensive linguistic analysis of word context in the near-synonym lexical choice task. We apply the *amph* data set [2] which contains occurrences of four *think* lexemes in Finnish with over 650 morphological, semantic, syntactic, and extra-linguistic features. It has been shown that a rich manually selected feature set improves supervised classification based on polytomous logistic regression in the near-synonym lexical choice task [2]. In this work we verify the earlier results and take a step forward by using unsupervised and semi-supervised methods in the task. This direction is important for those NLP tasks in which there is not much labelled training data available. In some tasks unsupervised methods perform as well as supervised methods, or even

better (e.g., [13,24]), because of their wide coverage and ability to generalise to new data. Furthermore, unsupervised methods are good in explorative research of previously unseen data and in visualising the structure of complex data. In addition, we experiment with automatic feature selection in order to find the best-representative features for the task and to find a feature set that enhances the unsupervised results.

On a larger scale, this work aims towards understanding semantics of synonymous words: We take an explorative view, use an extensive set of linguistic features, and study how different machine learning approaches are able to find the similarities and differences between near-synonyms. We also study how syntactic, semantic, and morphological features affect the results. We examine how the number and quality of the features affect the classification accuracy in the near-synonym lexical choice task. Although our experiments are conducted for a data set of only one set of words in the Finnish language, the experimental setting is general and can be conducted for other words, data sets, and languages. The linguistic analysis of the data set is partially manual, but similar analysis can be performed with existing resources.

## 1.1 Related Work

The problem of lexical choice has been studied in some earlier works. [8] created a lexical choice system by considering the branches of an ontology as clusters of synonyms. The clustering was performed based on manually defined dimensions of denotational, stylistic, expressive, and structural variations. [11] proposed extraction patterns to get near-synonym differences from a synonym dictionary. [7] proposed a lexical choice method that uses co-occurrence networks. The data set contained seven English near-synonym sets, such as *difficult*, *hard*, *tough* and *give*, *provide*, *offer*. Rather recently, [23] experimented with the same data set. They used latent semantic analysis with lexical-level co-occurrence in a supervised manner by applying support vector machines. Our work concerns a similar set of near-synonyms but extends the work into Finnish, a very large set of linguistic features and a variety of machine learning approaches.

Lexical choice is closely related to other tasks common in the natural language processing (NLP) community. *Lexical substitution* [15] is a task in which a word in a context is to be replaced with a synonymous word that is also suitable for the context. However, there is no predefined list of possible answers available. Lexical substitution has gained some popularity e.g., in SemEval tasks [16,18]. In the information retrieval community, a similar task is *query expansion* [22]. A more common task is *word sense disambiguation* (WSD) [20], in which the meaning of a polysemous word is selected from a set of alternatives. Due to the similarities between lexical choice and WSD, the approaches may use the same categorisation or clustering methods. Machine translation (MT) is also a large application area [1,4]. In MT, the task is often referred as *lexical selection*, where the target word is selected from a set of possible translations. Many vector space models have been evaluated in lexical choice tasks, such as the synonym part of the TOEFL language test [14,19].

The *amph* data set has previously been analysed based on statistical measures, manual feature selection and classification based on polytomous logistic regression according to the one-vs-rest, multinomial and other heuristics [2]. Arppe observed that a supervised approach, polytomous logistic regression seems to reach an accuracy rate

of 60–66% of the instances. The results did not notably improve with the addition of further granularity in semantic and structural subclassification of the syntactic roles. Subsequently, [3] compared polytomous logistic regression and other supervised approaches. They concluded that there is no large difference on the accuracy rates of the tested supervised machine learning classifiers on the *amph* data set.

## 2 Data

The *amph* data set used in this work represents Finnish, which is part of the Uralic language family and is known for its highly rich agglutinative morphology. The *amph* data set is a collection of the four most frequent Finnish *think* lexemes: *ajatella* (*think* in English), *harkita* (*consider*), *miettiä* (*reflect*), and *pohtia* (*ponder*). It consists of 3404 occurrences that are collected from newsgroup postings and newspaper articles. The distribution of the four lexemes is given in Table 1. The most frequent lexeme is present in about 44% of all data instances and the least frequent lexeme comprises approximately 11% of the data. The data set is publicly available<sup>1</sup>.

**Table 1.** *Think* lexemes and their frequencies and percentages in the *amph* data set

Lexeme	Frequency	%
1. <i>ajatella</i> ( <i>think</i> )	1492	43.8
2. <i>harkita</i> ( <i>consider</i> )	387	11.4
3. <i>miettiä</i> ( <i>reflect</i> )	812	23.9
4. <i>pohtia</i> ( <i>ponder</i> )	713	20.9
<b>Total</b>	<b>3404</b>	<b>100.0</b>

The *amph* data set has been morphologically and syntactically analysed with a computational implementation of functional dependency grammar for Finnish [21], with manual validation and correction. In addition, the analysis has been supplemented with semantic and structural subclassifications of syntactic arguments and the verb-chain. For further details, see [2, Sec. 2.2]. The data set consists of 216 binary atomic features and 435 binary feature combinations. Each feature has at least 24 occurrences in the data set. The atomic features consist of morphological features, syntactic argument features, features associated with words in any syntactic position, and extra-linguistic features, such as the data source and the author of the text. The combined features consist of syntactic & semantic, syntactic & phrase-structure, syntactic argument & base-form lexeme and syntactic & morphological feature combinations. Semantic features do not exist as atomic features, but are always combined with syntactic features.

In this paper, we use two original feature sets: FULL, all 651 features, and ATOMIC, atomic features only (216 features), and compare their performance to the feature set M6, which has been manually selected from the FULL feature set to be linguistically interesting. It was presented in [2, page 194, referred as Model VI]. The set contains 46

<sup>1</sup> <http://www.csc.fi/english/research/software/amph>

features, consisting of 10 verb-chain general morphological features, and their semantic classifications (6 combined features), 10 syntactic argument types, and their selected or collapsed subtypes (20 features). In addition to the features present in the FULL feature set, Arppe’s M6 contains some feature combinations of the original features that are available in a supplementary data table THINK.data.extra. For more details about the features and the compilation of the data sets, see [2, Sec. 2.4, 3.1].

### 3 Methods

In this paper, the task is to select the most suitable lexeme out of a set of near-synonym alternatives for each context. The task is referred to *fill-in-the-blank* (FITB) [7,23]: in a corpus of sentences containing one of the near-synonyms, the original lexeme is removed from the sentences and the goal is to guess which of the near-synonyms is the missing word. Thus, the task reduces to a standard classification problem. In practice, we trained methods from different machine learning approaches to conduct the lexical choice and then used a labelled test data set to evaluate the classification accuracies. In addition to the two original feature sets, automatic feature selection was performed for the FULL feature set to obtain a small subset of features that contain relevant information for the task and obtain better classification accuracy.

#### 3.1 Feature Selection

The data set used in this work contains an extensive feature set, which also includes noise, i.e., linguistic information not crucial to the task. In a previous work, [2] experimented with different manually selected feature sets. In our work, we aim to select automatically a set of features that best distinguish between the lexemes of the data set. The technique of selecting a subset of relevant features is known as *feature* or *variable selection* [9], which can help alleviate the curse of dimensionality, enhance generalisation capability, speed up the learning process and improve model interpretability. For computational reasons, it is typically not feasible to compute an exhaustive search of all possible feature subsets. A very simple heuristic algorithm, the *forward feature selection* algorithm, starts from an empty set and adds one feature at a time, choosing the feature which most improves an evaluation criterion.

#### 3.2 Unsupervised Learning

Unsupervised learning methods do not use any labelled data about the correct clustering or categorisation but analyse the structure of the data. We discuss three unsupervised learning methods: K-means, self-organising map, and independent component analysis.

*K-means* is one of the best known clustering algorithms due to its efficiency and simplicity. It clusters data items into  $K$  clusters starting from a random initialisation of cluster centroids. The algorithm alternates between two steps: each data item is first assigned to its nearest cluster centroid, and then the centroids are updated as the means of the data items assigned to the clusters. Different distance measures can be used while the Euclidean distance metric is a common choice.

The *self-organising map* (SOM) [12] is an artificial neural network that is trained with unsupervised learning. The SOM fits an approximated manifold of prototype vectors to a data distribution. During training, the prototype vectors will start to approximate the data distribution, and the prototype vectors will self-organise so that neighbouring prototypes will model mutually similar data points. SOM can be used especially for explorative data analysis and data visualisation.

Both SOM and K-means are vector quantisation methods that cluster high-dimensional data in an unsupervised manner and represent the original data with few prototype vectors. The methods can also be used as simple classifiers. On the other hand, *independent component analysis* (ICA) [5] is an unsupervised feature extraction method that finds a representation of data in a new space. ICA assumes that each data item is generated as an instantaneous linear mixture of statistically independent components. There are several algorithms which can learn both the static mixing matrix and the component activities based on the observed data and the assumption of statistical independence.

### 3.3 Semi-supervised Learning

A semi-supervised approach used in this work is a semi-supervised version of the k-nearest-neighbours (kNN) method (see the following section). The selected learning approach is called *self-training*, in which previously classified data points are used as additional labelled data for further classifications. We used a straight-forward extension from the 1NN classifier introduced by [25].

### 3.4 Supervised Learning

Since unsupervised methods may not find the correct clustering accurately, we also experiment with some supervised methods. In supervised learning, labelled data are provided and the task is to predict correct labels for previously unseen data without labels. We consider three different methods: k-nearest-neighbours (kNN), feed-forward artificial neural network (ANN), and multinomial logistic regression (MNR), one form of polytomous logistic regression. Out of these three methods, kNN and MNR have been previously applied to the *amph* data set [2,3].

The *k-nearest-neighbours* method (kNN) [6] is a non-parametric learning method that classifies new data items according to those labelled data items that are most similar to the new one. The kNN method has no parameters to be learned, but the number of neighbours  $k$  and the distance measure have to be selected.

*Feed-forward artificial neural network* (ANN) is a parametric method that learns a nonlinear mapping from the input features to the given output labels from training data with scaled conjugate gradient (see, e.g., [10]). The network structure has an input layer, at least one hidden layer with nonlinear activation functions and a linear output layer. We use the network for classification and define a single output for each label.

*Multinomial logistic regression* (MNR) [17] is a linear parametric method. It learns a mapping from continuous and categorical dependent variables, usually assuming one outcome category as a default case against which the other outcomes are contrasted. The model learns weights (log-odds) for each dependent variable.

### 3.5 Evaluation

The performance of the methods in this work is evaluated with *accuracy*: the ratio of correctly classified data items to all items. The results of the methods depend on the selected data set and initialisation, and thus we run an  $n$ -fold cross-validation by dividing the data into  $n$  sets, taking each set separately to be a test set, and training the data with the other  $n - 1$  sets. The reported average accuracies are calculated as the mean of the fold accuracies. Statistical significances are measured with the 1-sided Wilcoxon signed rank test.

The evaluation of the unsupervised clustering methods K-means and SOM require that a label is assigned to each cluster. The label of each cluster is set as the majority label among the data items in the cluster. There might be more clusters than possible labels. If accuracy were calculated for the training data, it would approach 100% when the number of clusters approaches the number of data points. However, we use separate train and test sets in cross-validation. Thus, while the number of clusters increases the accuracy gets close to the supervised 1NN classification accuracy.

## 4 Experiments and Results

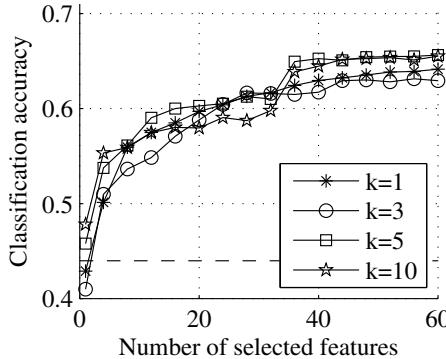
All the reported results have been produced with 20-fold cross-validation: each test set consists of 5% of the data, i.e., 170 instances. As a baseline method we classify all test data items to the largest category, lexeme 1. The average accuracy of the baseline is 0.44, the fraction of the largest lexeme class.

### 4.1 Feature Selection

We applied the forward feature selection method using the kNN classifiers with  $k = \{1, 3, 5, 10\}$  as the evaluation criteria for the FULL feature set. The kNN classifier was chosen because it was significantly faster to compute than an artificial neural network or multinomial logistic regression. Both the feature selection and the following classification were computed with the same data set because of the limited size of the *amph* data set. To alleviate this limitation, we used cross-validation in the evaluation criteria.

After feature selection, a kNN classifier with the corresponding number of neighbours  $k$  was applied to the reduced feature sets. The accuracy of the classification improved with the number of included features as shown in Fig. 1. The feature sets were evaluated with 20-fold cross-validation. 5NN quickly reached a plateau around 0.65–0.66 at about 40 features and we chose to use it for the automatically selected feature set Fs40. It has been included in the classification experiments.

The automatically selected set Fs40 contains six morphological features, two extra-linguistic features representing information about the text source, three features that mark that one of the lexemes appear earlier in the same text, and 29 syntactic features: 12 purely syntactic features, 12 syntactic features with semantic subtypes, and 5 syntactic features with a specific word and its part-of-speech. The linguistic categorisation of the first 10 features of the Fs40 set is given in Table 2. As examples, the first selected feature 1, related to indirect questions, appears with lexeme 3 (*miettiä*), when thinking



**Fig. 1.** Supervised classification accuracy of kNN for feature selection. The features are added incrementally with forward feature selection from the FULL feature set using kNN also in feature evaluation. The dashed horizontal line shows classification accuracy with a random classifier.

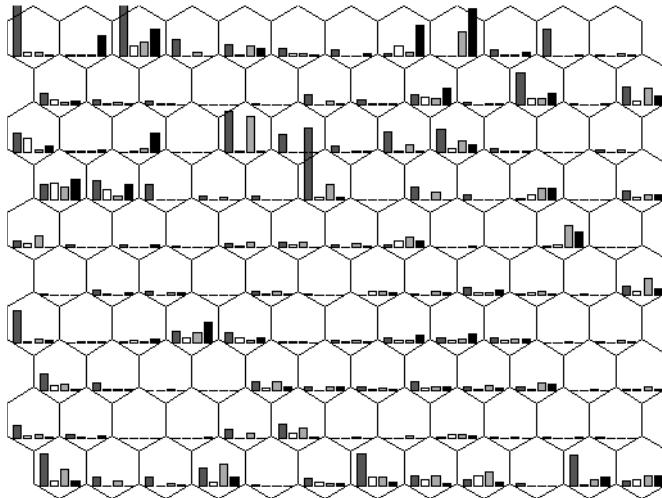
**Table 2.** The first ten features of the automatically selected Fs40 feature set and their existence in the Arppe's M6 feature set [2]

Feature	1	2	3	4	5	6	7	8	9	10
Morphological										x
Syntactic	x	x	x	x	x	x	x	x		
Semantic	x					x				
PoS										x
Also in M6 [2]	x	x	x	x	x	x				

is time-limited. Feature 2 is a significant determiner of the lexeme 2 (*harkita*). Feature 3 appears with lexeme 4 (*pohtia*), and is also associated with an expression of duration for the thinking process. The first automatically selected features match with the manual analysis of features that are good at predicting and depicting the *amph* verbs [2]; 6 out of the first 10 features exist also in Arppe's M6, which is also indicated in the table. Overall, only 8 out of 40 Fs40 features exist in Arppe's M6.

## 4.2 Unsupervised

To get an overview of the data, we first show a SOM clustering and visualisation of the FULL feature set in Fig. 2. A  $10 \times 12$  SOM lattice of prototype vectors was initialised with eigenvectors corresponding to the two largest eigenvalues. The SOM was trained with the whole data set and after training the best matching cells were calculated for each data item. The labels of the data items are shown in the figure as gray-scale bars: the height of a bar corresponds to the number of data items located in each hexagon cell. As the figure shows, the lexeme selection task with the FULL feature set is very difficult for an unsupervised clustering method: the data set contains also other structure than the four lexemes, and thus SOM cannot form nicely separated clusters of the four



**Fig. 2.** Unsupervised SOM clustering and visualisation using FULL feature set. Each hexagon corresponds to one prototype vector. The grey-scale bars show the distribution of the four lexemes assigned to each cell.

lexemes. Lexeme 1 (dark grey), that occurs in about 44% of the data set, is located in the upper and left hand side part of the map. Instances of lexeme 2 (white) are in the top left corner and in the middle of the map from top to bottom. The largest occurrences of lexeme 3 (light grey) are located in the right bottom part of the map. Lexeme 3 seems to be complementary to lexeme 1. Lexeme 4 (black) is located on the top and right-hand side of the map. In the top left corner is an area of all lexemes, whereas cells with a pair of strong lexemes can be seen on many areas of the map.

Similarly to SOM, independent component analysis of the FULL feature set does not seem to extract components that match well with the *think* lexemes. The resulting components clearly find an underlying structure in the data set, but the learned structure does not reflect the wanted classification. Thus, the results are not shown here or analysed further.

The K-means classification accuracy for 20-fold cross-validation of FULL, ATOMIC, and Fs40 feature sets are compared with the results of Arppe's M6 feature set in Table 3. The accuracies are calculated for the number of clusters varying between 4 and 100. The correlation distance measure was applied. Fs40 needed the addition of normally distributed noise to be able to distinguish between the vectors. The automatically selected feature set Fs40 performs significantly better than any of the other tested feature sets even though it contains the smallest number of features. Nevertheless, clustering into four categories does not exceed the baseline accuracy of 0.44.

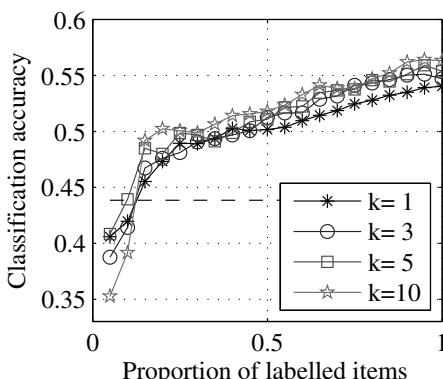
**Table 3.** Unsupervised classification accuracy of K-means using the four feature sets. Fs40 performs significantly better for all numbers of clusters  $K$  (in bold) against all other feature sets.

$K$	FULL ATOMIC Fs40			M6 [2]
	Avg	Avg	Avg	Avg
4	0.44	0.44	<b>0.45</b>	0.44
6	0.44	0.44	<b>0.47</b>	0.45
8	0.44	0.44	<b>0.50</b>	0.46
10	0.44	0.45	<b>0.51</b>	0.47
20	0.46	0.48	<b>0.55</b>	0.49
30	0.49	0.48	<b>0.56</b>	0.50
50	0.52	0.50	<b>0.57</b>	0.54
100	0.54	0.51	<b>0.59</b>	0.56

### 4.3 Semi-supervised

Since unsupervised methods do not perform very well for the tested feature sets, we next experiment with the semi-supervised method with both labelled and unlabelled data. In the semi-supervised kNN clustering with  $k = \{1, 3, 5, 10\}$  the percentages 5–100% of labelled training data were experimented. The averages of classification accuracies with the ATOMIC feature set, using 20-fold cross-validation, are shown in Fig. 3. With labelled data of 15% or more the semi-supervised 10NN performs best. With all values of  $k$  the accuracy is over the baseline when at least 15% of data is labelled. Statistically significant differences exist between 1NN and the other methods if 50% or more of the data was labelled. We got similar results also with the other feature sets (not shown).

We also tested with a fixed number of labelled data items, varying the amount of unlabelled data, and found that unlabelled data disturbs the classifier. This supports the



**Fig. 3.** Semi-supervised classification accuracy of semi-supervised kNN using ATOMIC feature set, varying the proportion of labelled data items between 0.05–1. The dashed line shows the baseline.

findings with SOM and ICA that the data set contains also some other structure than which separates the four lexemes.

#### 4.4 Supervised

Unsupervised and semi-supervised methods were not able to find very well the structure that differentiates the four lexemes. Thus we experiment with fully labelled data. The experiments with ANN were conducted with one hidden layer of 20 neurons. The FULL and ATOMIC feature sets were too large for MNR computation, and thus the dimensionality was reduced with principal component analysis (PCA) into 150 dimensions, which removed only a small fraction of the signal. The kNN method was run with the Euclidean distance.

Table 4 shows classification accuracy of the supervised ANN and MNR methods, and kNN with a varying number of neighbours. The feature sets are the original sets FULL, ATOMIC, as well as the automatically selected smaller feature set Fs40. Also results with Arppe's M6 feature set [2] are shown. The averages are calculated with 20-fold cross-validation. The highest supervised accuracy, 0.66, is obtained with MNR and the FULL feature set. The ANN classifier performs best with the automatically selected Fs40 and the FULL set. The best results with kNN are obtained with middle values of  $k$  for all feature sets. The best result for kNN, 0.65, was obtained with the automatically selected Fs40 feature set for  $k = 5$ . The result is natural because the feature set was optimized for 5NN.

All the results are clearly better than the baseline 0.44. The results of FULL and Fs40 are significantly better than ATOMIC and Arppe's manually selected M6 with the ANN classifier. For kNN, Fs40 performed significantly better than all other methods, except for the smallest value of  $k$ . In contrast, for MNR, only the FULL feature set performs

**Table 4.** Supervised classification accuracy of ANN, MNR, and kNN with different number of neighbours  $k$  using the four feature sets. The result for the significantly best feature set is printed in bold for each method (row). For kNN, the best values of  $k$  for each feature set is underlined.

	<b>FULL</b> <u>Avg</u>	<b>ATOMIC</b> <u>Avg</u>	<b>Fs40</b> <u>Avg</u>	<b>M6</b> [2] <u>Avg</u>
ANN	<b>0.62</b>	0.59	<b>0.64</b>	0.59
MNR	<b>0.66</b> <sup>1</sup>	0.61 <sup>1</sup>	0.60	0.63
kNN $k = 1$	<b>0.60</b>	0.54	0.47	0.53
3	<u>0.60</u>	0.55	<b>0.64</b>	0.58
5	<u>0.60</u>	<u>0.56</u>	<b>0.65</b>	0.58
10	<u>0.61</u>	<u>0.57</u>	<b>0.63</b>	<u>0.59</u>
20	<u>0.60</u>	<u>0.56</u>	<b>0.64</b>	<u>0.59</u>
30	0.59	<u>0.56</u>	<b>0.63</b>	0.58
50	0.57	0.54	<b>0.62</b>	0.57
100	0.54	0.54	<b>0.61</b>	0.56

<sup>1</sup> Computed for the first 150 principal components.

better than Arppe's M6, possibly because the feature set was selected using MNR results [2]. The results show that supervised feature selection can reduce the complexity of a parametric supervised method (ANN) without lowering quality and even improve a non-parametric supervised methods (kNN) by selecting features relevant to the task.

## 5 Discussion and Conclusions

In this paper we have studied the use of an extensive set of linguistic features from the *amph* data set in the near-synonym lexical choice task. We used a number of machine learning methods and experimented on an automatically selected feature set. While the automatically selected feature set uses a significantly smaller number of features, the results are comparable to the original feature sets.

The best classification accuracy obtained in the task was 0.66 with multinomial logistic regression (MNR) for the FULL feature set of 651 linguistic features, by first reducing the original dimensionality with principal component analysis to 150. The automatically selected feature set FS40 of only 40 features performed overall very well: it improved over the manually selected Arppe's M6 feature set [2] with ANN and gave a comparable result to the FULL feature set. It also gave better results than any of the other feature sets with K-means and kNN. The automatically selected feature set FS40 consists mostly of syntactic features, but also some semantic and morphological features were selected as the most important ones. The FULL set generally improved over the ATOMIC set, suggesting that combining or extracting features can help classification. An analysis of the effect of different manually selected syntactic, semantic, and morphological feature sets can be found in [2, p. 207].

All tested supervised methods reached approximately the same level of performance, the best classification accuracies of each method were between 0.60 and 0.66. This supports the findings in [3] which says that this is the maximum accuracy that can be obtained with supervised methods for this data set. Unsupervised methods did not perform as well as supervised methods, which is natural behaviour with a complex data set like *amph*. However, supervised feature selection can improve unsupervised classification accuracy with an additional advantage of a significantly smaller set of features. Unsupervised feature selection based on information theoretic measures instead of supervised feature selection would be a step towards a completely unsupervised method. Feature selection based on the simple kNN classifier does not improve the results of the other supervised classifiers, when comparing to the FULL feature set. However, the smaller models contribute to faster model training as well as smaller memory and computational complexity.

**Acknowledgments.** We kindly thank Oskar Kohonen, Krista Lagus and Timo Honkela for their valuable comments while writing this paper. This work has been financially supported by The Finnish Graduate School in Language Studies (Langnet), Academy of Finland directly and through AIRC Centre of Excellence, and META-NET Network of Excellence.

## References

1. Apidianaki, M.: Data-driven semantic analysis for multilingual WSD and lexical selection in translation. In: *Proceedings of EACL 2009*, pp. 77–85. ACL (2009)
2. Arppe, A.: Univariate, bivariate, and multivariate methods in corpus-based lexicography—a study of synonymy. Ph.D. thesis, University of Helsinki, Finland (2008)
3. Baayen, R.H., Arppe, A.: Statistical classification and principles of human learning. In: *Proceedings of QITL*, vol. 4 (2011)
4. Carpuat, M., Wu, D.: Improving statistical machine translation using word sense disambiguation. In: *Proceedings of EMNLP-CoNLL 2007*, pp. 61–72 (2007)
5. Comon, P.: Independent component analysis, a new concept? *Signal processing* 36(3), 287–314 (1994)
6. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1), 21–27 (1967)
7. Edmonds, P.: Choosing the word most typical in context using a lexical co-occurrence network. In: *Proceedings of EACL 1997*, pp. 507–509. ACL (1997)
8. Edmonds, P., Hirst, G.: Near-synonymy and lexical choice. *Computational Linguistics* 28(2), 105–144 (2002)
9. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
10. Haykin, S.: *Neural networks: a comprehensive foundation*. Prentice-Hall, Englewood Cliffs (1994)
11. Inkpen, D., Graeme, H.: Building and using a lexical knowledge base of near-synonym differences. *Computational Linguistics* 32(2), 223–262 (2006)
12. Kohonen, T.: *Self-Organizing Maps*. Springer Series in Information Sciences, vol. 30. Springer, New York (2001)
13. Kurimo, M., Creutz, M., Turunen, V.: Overview of morpho challenge in CLEF 2007. In: *Working Notes of the CLEF 2007 Workshop*, pp. 19–21 (2007)
14. 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* 104(2), 211–240 (1997)
15. McCarthy, D.: Lexical substitution as a task for WSD evaluation. In: *Proceedings of SIGLEX/SENSEVAL 2002*, pp. 109–115. ACL (2002)
16. McCarthy, D., Navigli, R.: SemEval-2007 task 10: English lexical substitution task. In: *Proceedings of SemEval 2007*, pp. 48–53. ACL (2007)
17. McCullagh, P., Nelder, J.A.: *Generalized Linear Models*. Chapman & Hall, New York (1990)
18. Mihalcea, R., Sinha, R., McCarthy, D.: SemEval-2010 Task 2: Cross-lingual lexical substitution. In: *Proceedings of SemEval 2010*, pp. 9–14. ACL (2010)
19. Sahlgren, M.: *The Word-Space Model*. Ph.D. thesis, Department of Linguistics, Stockholm University, Stockholm, Sweden (2006)
20. Schütze, H.: Dimensions of meaning. In: *Proceedings of SC 1992*, pp. 787–796. IEEE (1992)
21. Tapanainen, P., Järvinen, T.: A non-projective dependency parser. In: *Proceedings of Applied Natural Language Processing*, pp. 64–71. ACL (1997)
22. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: *Proceedings of ACM SIGIR 1994*, pp. 61–69. Springer, Heidelberg (1994)
23. Wang, T., Hirst, G.: Near-synonym lexical choice in latent semantic space. In: *Proceedings of Coling 2010*, pp. 1182–1190. ACL (2010)
24. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: *Proceedings of ACL 1995*, pp. 189–196. ACL (1995)
25. Zhu, X., Goldberg, A.B.: *Introduction to semi-supervised learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2009)

# Abduction in Games for a Flexible Approach to Discourse Planning

Ralf Klabunde, Sebastian Reuße, and Björn Schlünder

Ruhr-Universität Bochum, Department of Linguistics  
`{ralf.klabunde,sebastian.reusse,bjoern.schlunder}@rub.de`

**Abstract.** We propose a new approach to document planning for natural language generation that considers this process as strategic interaction between the generation component and a user model. The core task of the user model is abductive reasoning about the usefulness of rhetorical relations for the document plan with respect to the user's information requirements. Since the different preferences of the generation component and the user model are defined by parametrised utility functions, we achieve a highly flexible approach to the generation of document plans for different users. We apply this approach to the generation of reports on performance data. The questionnaire-based evaluation we accomplished so far corroborates the assumptions made in the model.

## 1 Introduction

The basic goal of natural language generation (NLG) systems is the transformation of unwieldy information – for example, data streams or ontological knowledge – to linguistic representations that are more accessible to humans. Accessibility, in the broader sense, largely depends on the individual addressee: Different addressees possess different amounts of background knowledge which they may bring to bear upon the act of interpreting a text, and their different interests shape specific perspectives upon a discourse topic. For example, a report on meteorological data which elaborates the significance of reported climatic configurations may be indispensable to the layman, but wholly inadequate to the more meteorologically versed user.

If an NLG system has to cover a wide range of addressees, it has to incorporate a means of selecting and structuring content in a way that takes into account the communicative relevance of the generated text. In this paper, we present a general, user-oriented approach to discourse structuring that allows a flexible adaptation of document structures to individual users. The basic idea is to introduce a representation of a user's interests and prior beliefs, reflected as assumption costs on certain hypotheses, and to anticipate the relevance of different candidate messages during the generation process by means of abductive inference.

In what follows, we will first review some existing approaches to achieving text customized for different users in order to show that language generation should be considered as a joint activity between two independent agents (or players, in

game-theoretic terms). Based on this, we present our model of user-customized document generation which makes explicit reference to a user's potential interests and allows user types to be defined dynamically with reference to those interests. We will give an example of how this model is realized in a specific domain, and present the first evaluation result how well the resulting documents have been accepted by readers.

## 2 User-Oriented Document Planning

The insight that generated texts should be tailored to the specific users in order to enhance the acceptability of the text, is not really new. However, since user models are primarily used for pragmatic decisions during the generation process, it is remarkable that the principles behind formal approaches to pragmatics – describing the joint activities of the interlocutors – did not receive much attention in NLG systems.

For example, Hovy (1988) defines rhetorical relations as planning operators with associated pre- and post-conditions that describe the interests and knowledge states of the agents involved. While this approach results in sophisticated user-oriented documents, there are no means of determining whether one of several alternative document structures might be more relevant than others with regard to a user's expectations. More recently, Mairesse and Walker (2010) describe a generation system which explicitly seeks to match output to a user based on various cognitive decisions, but does so only on the level of linguistic realization.

Dale and Reiter (2000) mention the possibility to address user variability by pre-specifying document schemas for a finite number of user types. Each user type corresponds to one alternative document structure definition. Such an approach might become inconvenient when multiple factors are involved in the mapping, such that the number of types increases exponentially with the number of factors. Furthermore, there is no explicit representation of the reasons why a document schema is defined the way it is and why it includes the given information. Ultimately one has to rely on expert judgement.

In their overview, Zukerman and Litman (2001) describe the various approaches to user modeling in NLG, but this survey demonstrates nicely that generally accepted principles of user modeling do not exist so far. Instead a whole bundle of different approaches have been proposed, mostly linked to pragmatic tasks, but without recourse to formal pragmatics. Reiter et al. (2003) also mention the fact that little is known about the acquisition of user models.

Our approach centers around the well-established idea that language production should be considered as one component of a joint activity of two agents. Speakers adapt their utterances to the linguistic and cognitive abilities of the addressee and vice versa. The speaker's adaptation is based on stereotypes and associated defaults, the established common ground, and individual features of the respective addressee (See, e.g., Brennan et al. (2010)).

If one accepts this view of communication as highly flexible, joint activity, there are no clearly defined classes of users, but rather there is a set of different

needs and expectations an individual user might have. Prototypical users should be considered as mere points in a multidimensional space of user types, where points are located by assigning different priorities to a user's possible needs. Text plans are then to be derived with respect to this coordinate system and can be generated dynamically for every possible point in the user type space.

### 3 The Formal Model

The formal model for document planning we propose in this paper takes into account the aforementioned view of generation as a collaborative process.

#### 3.1 Data Source

Our model relies on some data source  $d$  from which we derive an initial pool of individual, atomic messages. The messages' content depends on how we devise the parsing mechanism mapping the data source to the document planners initial message repository.

#### 3.2 Rhetorical Relations

Our use of rhetorical relations as structuring means for document plans is more or less identical to the standard uses in NLG systems, except that we use a logical notation. We define how we may induce possible document structures over the unstructured pool of information derived from  $d$ , by giving logical definitions of how rhetorical relations may coherently be applied to complex or atomic messages. Messages are atomic if they are immediately derived from some data source, or complex if the message is made up of constituent messages, themselves either complex or atomic. The application of some type of rhetorical relation to its constituent messages results in a new, complex message of the same type and can be viewed as a partial document plan.

The definitions of the rhetorical relations are given with reference to the triggering messages' content or type, thus defining preconditions which, once satisfied, trigger the establishment of a relation. For example, a relation between two messages of specific type, with a constraint on some property of the second message, could be given as:

$$\begin{aligned} & \text{prerequisiteType}_1(X) \wedge \text{prerequisiteType}_2(Y) \\ & \wedge \text{prerequisiteProperty}(Y) \rightarrow \text{compositeMessage}(X, Y) \end{aligned}$$

We explicitly allow for a whole number of competing relations being triggered in some state during the generation of document plans, because we aim at singling out that relation that is most relevant to the specific user, according to the current state of the user model.

### 3.3 Relevance-Relations in a User Model

To represent how the different dimensions of potential user-interests constrain document structures, we define, where appropriate, whether some interest is indicative resp. counter-indicative of a specific message type. In other words, we provide a knowledge base consisting of theorems of the form ‘*hypothesis*  $\rightarrow$   $[\neg]messageType$ ’, where ‘ $\neg$ ’ marks counter-indicative rules. ‘*hypothesis*’ models a possible dimension of interest of a user, and ‘*messageType*’ corresponds to the type of a node in a document structure that is made relevant in the face of that dimension of interest. For example, we might use a rule ‘*preferDetail*  $\rightarrow$  *elaboration*’, which introduces a dimension called *preferDetail* into the user-type space, such that the generation of composite messages created by applying *elaboration*-relations will be encouraged for users whose model ranks prominently in the *preferDetail*-dimension. Since there could be multiple hypotheses indicating a (potentially composite) message’s relevance, and since not all hypotheses are appropriate for all user types, assumption costs are introduced.

The user model assigns user coordinates in the space created by the different dimensions introduced by the relevance-relations. We interpret these coordinates as the aforementioned assumption costs of hypotheses about the user, where each dimension corresponds to one hypothesis. For example, a relatively high numerical value for *preferDetail* would indicate some reluctance to assume that the user prefers detailed information and as such would limit the effect of *preferDetail*, encouraging the generation of certain messages for this user.

Be  $T$  a knowledge base,  $H$  the set of hypotheses given in  $T$ ,  $\psi$  a document plan, and  $cost$  a function that assigns assumption costs to hypotheses. We call the process of selecting the most felicitous hypothesis *naïve abduction*. If  $\langle T, \psi, costs \rangle$  is a cost-based abduction problem, we try to find  $h^* \in H$  such that:

$$\forall h \in H : match(h, \psi) - cost(h) \leq match(h^*, \psi) - cost(h^*)$$

The function  $match(h, \psi)$  measures how coherently some hypothetical user-interest  $h$  integrates with a partial or complete document plan  $\psi$ . Let  $M_h = \{m | h \rightarrow m\}$  and  $M_h^- = \{m | h \rightarrow \neg m\}$ . We define:

$$match(h, \psi) := \gamma_1 \times |M_h \in \psi| - \gamma_2 \times |M_h^- \in \psi|$$

I.e., we count the number of messages related to some hypothetical interest according to the relevance definitions, increasing an initial score of zero for each expected message, and decreasing it for every counter-indicated message. This approach might be considered as a lean version of weighted abduction (Hobbs et al. (1993), Ovchinnikova et al. (2011)): To determine the compatibility between a document plan and a hypothesis, the system first assumes that the hypothesis  $h$  is true and subsequently tests if the document plan under consideration is relevant given the relevance-definitions, by counting messages related to the interests modelled by  $h$ . The parameters  $\gamma_{1,2}$  allow us to assign different weights to expected vs. deprecated messages.

### 3.4 A Game-Theoretic Approach

Our algorithm for user-tailored document planning uses a normal form game (cf. Leyton-Brown and Shoham (2008); Parikh (2010))  $\langle \{S, L\}, \{A_S, A_L\}, \{U_S, U_L\} \rangle$  which is iteratively played, with each iteration effectively establishing a single rhetorical relation over a subset of messages taken from the message pool.

The game is defined for two agents, the generation-system  $S$  and a user model  $L$ . During a single iteration, the system considers different alternative actions  $A_S$ , where each action corresponds to one rhetorical relation which might possibly be applied to a set of component messages taken from the repository. The set of possible rhetorical relations to be established is constrained by the prerequisites on component messages defined for the different relations.  $A_L$  consists of possible hypotheses an addressee might assume, in order to explain why some composite message generated by the system might be relevant to him. Finally, the utility-functions  $U_S$  and  $U_L$  determine the felicity of the resulting combination of generated composite message and underlying assumptions.

The definitions of the utility functions draw on three basic notions which model the ‘cognitive burden’ in establishing the resulting document structure:

1. The aforementioned function  $match(h, \psi)$ .
2. A metric  $complexity(\psi)$  that indicates the structural complexity of a (partial or complete) document plan  $\psi$  by simply counting the number of message nodes contained in it.
3. The function  $cost(h)$  gives the assumption cost of some hypothetical interest  $h$  according to the user model. The higher the cost, the less likely we are to assume that the interest represented by  $h$  does apply to the user.

Given these notions, we define the utility functions for  $S$  and  $L$  as follows:

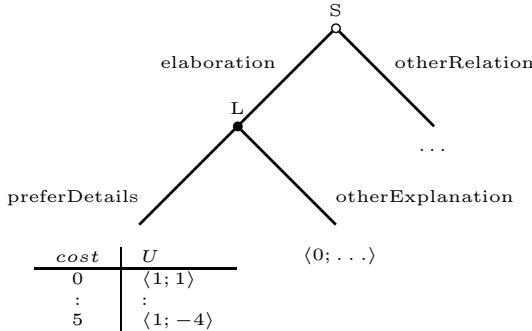
$$U_L(m, h) := \alpha_1 \times match(h, \{m\}) - \alpha_2 \times cost(h)$$

$$U_S(m, h) := \frac{\beta_1 \times match(h, \psi \cup \{m\})}{\beta_2 \times complexity(m)}$$

Both agents,  $S$  and  $L$ , prefer documents which are coherent with regard to a user’s potential interests. Furthermore, the generation system seeks to generate structurally plain documents. Both formulas are parameterized in order to control the impact of each contributing factor ( $\alpha_{1,2}, \beta_{1,2}$ ).

Figure 1 shows a partial game-tree with possible payoffs. Here, the generation system considers two possible relations it might establish. According to the relevance-relation definition, *preferDetails* would indicate the presence of an *elaboration*-message, and as such accounts for a utility of one in the agents’ payoffs, while *otherExplanation*, given that there is no relation to *elaboration*-messages in the assumed relevance-model, yields a utility of zero. To determine  $L$ ’s utility, we also discount the assumption cost of *preferDetails* from his payoffs, so that the total payoff varies with the specific type of addressee we assume.

The game-theoretic mechanism as defined above is reiterated according to the algorithm shown in Table 1. We iteratively apply the most felicitous type



**Fig. 1.** An example of a single message-generation-game in extensive form. Payoffs are shown as function of  $cost$ .

**Table 1.** A game-theoretic document-generation algorithm. In each iteration, an inventory of possible new messages is generated by applying rhetorical relations to subsets of the message pool. Out of the inventory of possible new messages, the most felicitous one is selected for subsequent generation by adding it back into the message pool while removing all of its constituents from it.

```

1:  $POOL \leftarrow$  all messages derived from  $d$ 
2:  $A_L \leftarrow \{\text{all interest types used in the relation-relevance definition}\}$ 
3: while  $(A_S \leftarrow \{\text{rhetorical relations which may link elements in } POOL\}) \neq \emptyset$ :
4:    $\langle as, aL \rangle \leftarrow$  pareto-optimal pure strategy equilibirum of
    :    $\langle \{S, L\}, \{A_S, A_L\}, \{U_S, U_L\} \rangle$ 
5:    $POOL \leftarrow (POOL \setminus \{\text{constituents of } as\}) \cup as$ 
6: return  $POOL$ 

```

of rhetorical relation until no more relations are applicable. As a result, the message pool contains one or several tree structures, depending on whether or not at some point a conjoining relation existed and was optimal in terms of both agent's goals.

### 3.5 An Example Game

Table 2 provides a schematic example of how our abstract model of document generation can be instantiated into a fully-fledged generation system. It lists the four input factors determining the construction of a document plan. The **message pool** contains a set of atomic messages which the system will seek to combine by means of rhetorical relations. In this case, we assume that the data source provides some basic information (*someMsg*), accompanied by related messages (*bgInfo*, *advice*) which either relate additional information regarding *someMsg*, or dispense relevant advice. The **text grammar** then defines

**Table 2.** A schematic instantiation of the document generation model

<b>Message Pool:</b> <i>someMsg, bgInfo, advice</i>	<b>Text Grammar:</b> $someMsg \wedge bgInfo \rightarrow elaboration$ $someMsg \wedge advice \rightarrow interpretation$
<b>Relevance Theory:</b> $isNovice \rightarrow interpretation$ $preferDetail \rightarrow elaboration$	<b>User Model:</b> $cost(isNovice) := 0$ $cost(preferDetail) := 5$

**Table 3.** Complete first iteration of the schematic example system

<i>S</i> generates <i>elaboration</i>	
(a) <i>L</i> assumes <i>isNovice</i>	(b) <i>L</i> assumes <i>preferDetail</i>
– <i>S</i> 's utility: $0/2 = 0$	– <i>S</i> 's utility: $1/2 = 0.5$
– <i>L</i> 's utility: $0 - 0 = 0$	– <i>L</i> 's utility: $1 - 5 = -4$
<i>S</i> generates <i>interpretation</i>	
(a) <i>L</i> assumes <i>isNovice</i>	(b) <i>L</i> assumes <i>preferDetail</i>
– <i>S</i> 's utility: $1/2 = 0.5$	– <i>S</i> 's utility: $0/2 = 0$
– <i>L</i> 's utility: $1 - 0 = 1$	– <i>L</i> 's utility: $0 - 5 = -5$

how these messages may be combined in a coherent way to form new complex messages. As assumed above, combining *someMsg* with background information will form an *elaboration*, while providing advice alternatively creates an *interpretation* message.

Once the means of applying rhetorical relations are defined, the **relevance theory** indicates when each of the alternative ways of forming complex messages might be relevant, by listing hypothetical aspects of a user's interests and the message types related to those interests. Here, we assume that a novice in our schematic domain will prefer *interpretation* messages, while a preference for detail is expected to give rise to *elaboration*. Finally, the **user model** specifies the actual interests of the user currently served, by listing assumption costs for each of the predicates used in modeling user interests. In this example we assume the user is likely a novice, since assuming *isNovice* incurs no cost, while also assuming that she has no particular interest in detail, since *preferDetail* comes with a relatively high cost.

Once these factors are set, the system begins constructing a document plan by iterating through all relations applicable in a single turn, and determining the payoffs of both agents relative to the message in consideration and each hypothetical dimension of the listener's interests. Table 3 shows the complete first turn of the generation system set out in Table 2 and each agent's payoffs

**Table 4.** The outcome achieved after the first iteration

	<i>elaboration</i>	<i>interpretation</i>
<i>isNovice</i>	$\langle 0, 0 \rangle$	$\langle 0.5, 0 \rangle$ ✓
<i>preferDetail</i>	$\langle 0.5, -4 \rangle$	$\langle 0, -5 \rangle$

according to the relevant utility function. Table 4 shows the turn in normal form and marks the game’s equilibrium. In this case, both agent’s goals (coherence, relevance and concision) are best served by generating an *interpretation* message from the message pool. At the end of the turn, the *interpretation* will accordingly be added into the document pool, while its constituent messages are removed from it.

## 4 Implementation and Application

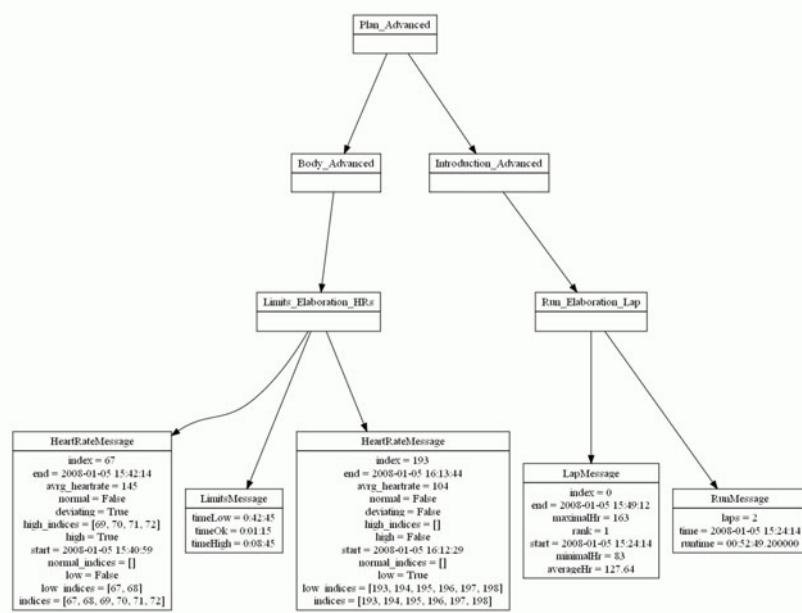
We applied this model of document planning to the generation of runner’s performance data. These data, generated by a heart-rate monitor device while jogging, are transformed into different texts according to the user’s needs. Relevant dimensions, defining a user’s background, are the frequency of exercise, degree of experience and prior training factors such as the degree of strain caused by exercising. Our main focus regarding user variability concerned the abundance of numerical data and the presence of explanatory content, but we also incorporated the different goals of amateur vs. experienced exercisers by generating appropriate advisory messages as to how these goals might be realized in the future.

We implemented the following rhetorical relations, following their standard descriptions in Rhetorical Structure Theory: *Preparation*, *Conjunction*, *Elaboration*, *Background*, *Sequence* and *Contrast*.

Although our model is capable of generating documents for all combinations of possible users according to the user-space spanned by the relevant dimensions, we restricted ourselves to three prototypical users, i.e. amateurs, advanced and semi-professional runners represented by salient points in the space of possible users. Figure 2 shows a sample document plan generated by the system.

The system is realized as a Python module. The core consists of a parser for the data files generated by the heart-rate monitors, and an abductive reasoning module used to trigger rhetorical relations. In order to solve the game played by the generation system and the user model – and thus determine the most felicitous message to be generated – we employed the freely available Gambit tool (McKelvey et al. (2010)).

The linguistic realization of the generated document plans was performed by a schema-based approach for message types, consisting of canned text interspersed with schematic references to a message content. For example, a message of type ‘RunMessage’, relating general information about the user’s exercise, is as follows:



**Fig. 2.** An example document within the exercise data domain. Box headers indicate message types. The contents list all data contained within a message. Edges visualize component relationships between messages.

RunMessage: You have been running a total of <laps> laps which has taken you <runtime>.

With less specific, complex message types, canned text can be used to introduce signal words into the realization. Interestingly this rather simple approach, when applied to the generated discourse structures, results in texts with apparently sufficient complexity for the users.

Contrast: <component[1]>. However, <component[2]>.

The first paragraph of a generated German text for occasional runners and its English translation are given in Table 5.

**Table 5.** Beginning of a generated text for occasional runners

German original:

English translation:

Am 13.10. waren Sie Joggen. Für Sie als Gelegenheitsläufer ist ein ausgewogener Lauf mit relativ niedriger Herzfrequenz wichtig. Während des Laufs lag Ihre Herzfrequenz 0:00:00 Minuten im für Sie optimalen Bereich. 0:07:45 Minuten lag sie darunter, 0:00:15 Minuten lag Sie oberhalb der optimalen Frequenz... On october 13th you went on a run. You as an occasional runner need to keep your run balanced and your heart rate relatively low and steady. During your run your heart rate was in the optimal interval for 0:00:00 Minutes. 0:07:45 it was lower, 0:00:15 it was higher than your optimal frequency...

## 5 Evaluation Results

Our evaluation concerned the acceptability of the texts for different users. For this, 41 questionnaires were completed by students of the department of linguistics.

### 5.1 Method of Evaluation

The questionnaires at first presented three short texts generated by the system. After reading, the test persons were asked to assess the stereotype of runner they belong to, and assess themselves in a set of attributes which correlate with attributes used in the generating system. The ratings were to be assessed on a scale from 1 (False) to 5 (True). Propositions to be assessed were:

- I train regularly.
  - I find training easy.
  - Sometimes I train too intensively.
  - I am in good physical condition.

After their self-assessment the test persons were asked to rate each of the texts previously read. For every text in question a set of three propositions was given, each to be rated on the same scale as before:

- The data presented in the text is explained sufficiently for my concerns.
  - The amount of data and numbers are after my fancy.
  - The information given is useful for my further training.

The test persons then were asked to choose one of the given texts as the one they would prefer.

## 5.2 First Results

From a subjective point of view, the overall ratings seem encouraging, since in every instance there is at least one text that is rated as acceptable, and the

**Table 6.** Average score of the document types in propositions checked over all instances

Document type	Explanation sufficient?	suf- okay?	Amount of Data	Information helpful?
Beginner	3.7	3.7	3.5	
Advanced	3.3	3.4	3.0	
Professional	3.7	3.0	3.0	

**Table 7.** Chosen text per runner type over all instances

Runner type	No. of instances	in- ner	Chose Begin- vanced	Chose Ad-	Chose Pro
Beginner	29	15	9	5	
Advanced	9	5	2	2	
Professional	3	1	1	1	

average rating of all questions regarding all three texts over all 41 instances is at least 3.0. Therefore we may conclude that the texts generated using the game-theoretic planning algorithm seem to be of sufficient quality.

The results, however, must be interpreted with caution, since only a small number of advanced runners and professionals were available. Since the test persons are only able to evaluate the resulting text and not the underlying abstract discourse structure, the results give us just a tentative hint that the resulting discourse structures are really tailored to different listener types.

Anyhow, the data gathered in the survey clearly indicate that the texts generated by the system are of sufficient quality and are accepted by most test persons. Therefore, the assumptions represented in the knowledge base seem to be mostly correct in principle. However, a more sophisticated evaluation is certainly needed as future work.

## 6 Summary and Outlook

We presented an account of document planning based on the rigorous definition of a user's needs and expectations and their relation to a document's content and structure. As matters stand, it will be necessary to gather further experimental evidence that takes into account the continuous representation of users in our model. So far, we worked with three prototypical user types only. However, we believe that the evaluation of our system already demonstrates a methodological gain, compared with some established approaches to user modeling in NLG.

## References

- Brennan, S.E., Galati, A., Kuhlen, A.K.: Two Minds, One Dialog: Coordinating Speaking and Understanding. In: Ross, B.H. (ed.) *The Psychology of Learning and Motivation*, vol. 53, pp. 301–344. Academic Press, Burlington (2010)
- Reiter, E., Dale, R.: *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge (2000)
- Hobbs, J.R., Stickel, M., Appelt, D., Martin, P.: Interpretation as Abduction. *Artificial Intelligence* 63, 69–142 (1993)
- Hovy, E.: Generating Natural Language under Pragmatic Constraints. Lawrence Erlbaum, Hillsdale (1988)
- Leyton-Brown, K., Shoham, Y.: *Essentials of Game Theory*. Morgan & Claypool Publishers (2008)
- McKelvey, R.D., McLennan, A.M., Turocy, T.L.: *Gambit: Software Tools for Game Theory*, Version 0.2010.09.01 (2010), <http://www.gambit-project.org>
- Mairesse, F., Walker, M.A.: Towards Personality-Based User Adaptation: Psychologically Informed Stylistic Language Generation. *User Modeling and User-Adapted Interaction* 20(3), 227–278 (2010)
- Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J., McCord, M.C., Mulkar-Mehta, R.: Abductive Reasoning with a Large Knowledge Base for Discourse Processing. In: *Proceedings of IWCS 2011*, Oxford, pp. 225–234 (2011)
- Parikh, P.: *Language and Equilibrium*. The MIT Press, Cambridge (2010)
- Reiter, E., Sripada, S., Williams, S.: Acquiring and Using Limited User Models in NLG. In: *Proceedings of ENLGW*, Budapest, pp. 87–94 (2003)
- Zukerman, I., Litman, D.: Natural Language Processing and User Modeling: Synergies and Limitations. *User Modeling and User-Adapted Interaction* 11, 129–158 (2001)

# Document-Specific Statistical Machine Translation for Improving Human Translation Productivity

Salim Roukos, Abraham Ittycheriah, and Jian-Ming Xu

IBM T.J. Watson Research Center,  
1101 Kitchawan Road, Rt. 134  
Yorktown Heights NY 10598, USA  
{roukos, abei, jianxu}@us.ibm.com

**Abstract.** We present two long term studies of the productivity of human translators by augmenting an existing Translation Memory system with Document-Specific Statistical Machine Translation. While the MT Post-Editing approach represents a significant change to the current practice of human translation, the two studies demonstrate a significant increase in the productivity of human translators, on the order of about 50% in the first study and of 68% in the second study conducted a year later. Both studies used a pool of 15 translators and concentrated on English-Spanish translation of IBM content in a production Translation Services Center.

**Keywords:** Statistical Machine Translation, Translation Memory, Post-Editing.

## 1 Introduction

Human language translation is a significant business activity reported to reach \$12 billion in 2010 [1]. There are a number of tools to increase the productivity of human translators from terminology management and lookup systems to translation memories (TM). TM tools store previous translations from a previous version of the content to be translated which are optionally used as a starting proposal for the translator to edit for creating the final translation for the content. A small improvement in productivity of a few percentage points can justify the technology tool investment for improving the translation process.

**Translation Memories.** Typically, the TM tool provides proposals for translating a segment from previous translations available in the TM. These proposals come in two varieties:

1. Exact Matches (EM) for a segment from previously translated segments – when there are multiple EMs for a segment, the human needs to check them and select the appropriate one due to context variations outside the segment. The EM segments are obviously done very quickly 10 to 100 times faster than translating the content from scratch.

2. Fuzzy Matches (FM) where a similarity score<sup>1</sup> is used to identify proposals from the TM that are close to the input source of a segment (usually a sentence) basis. A typical FM similarity score of greater than 70% is used to decide when one or more proposals are displayed to the translator.

Those segments that do not have a close FM proposal or an EM proposal are denoted as No Proposal (NP) segments. The FM segments require a few Post-Edits and are done faster than the NP segments. For the NP segment the translator starts from scratch in creating the target translation.

Depending on the content, the fraction of NP segments may be around 50% or higher (as in news where TM tools have not traditionally been used), hence reducing the impact on productivity by the previous translations via a TM tool. Another source of TM tool ineffectiveness is a change in the definition of segmentation since the matching in TM is done on the whole segment level rendering previous TMs nearly useless.

The TM tools have proven to increase human translation productivity significantly particularly in product documentation which tends to be slowly changing from one version to the next. We report on the productivity increase of TM tools in the experimental results section.

**MT Post-Editing.** The goal of this work is to explore the use of statistical machine translation to exploit previous translation memories (TMs), in addition to the EM and FM proposals produced by a TM tool, by creating what we will call MT proposals.

We explore the impact of MT proposals for both the NP case (where the MT proposal is the only proposal available to the translator) and the FM case (where the MT proposal is added to the other FM proposals).

There have been earlier efforts on using MT (mostly rule-based MT and some SMT) for Post-Editing with various successes though the studies were somewhat small scale. In our own experience at IBM, the benefit of rule-based MT was rather small if any. The MT customization effort of a rule-based engine is significant which requires a rather large project (several hundred thousand words) to justify the cost of customization.

With SMT, the customization process is automated (minimal cost) by using a previous TM, a bilingual parallel corpus, to adapt the SMT models for the new document. Hence, the customization cost is rather small enabling much wider applicability of using MT (as long as the translation project has some kind of relevant TM). In this work, we approached the customization process on a document specific basis.

We performed two long term studies over a two year period on the impact of Document-Specific SMT on human translation productivity in the context of IBM's production of translated content for a variety of products and software packages.

---

<sup>1</sup> Typically a Word Error Rate metric is used to measure the difference between the input source segment and a source segment in the TM; though some more general similarity metrics using additional features have been proposed.

We present the current IBM process for producing translated content in over 40 languages in Section 2. Section 3 presents the MT engine and the Document-Specific SMT adaptation. Section 4 presents the experimental results for two studies of using customized MT for English-Spanish translation of IBM content. Finally, we present our conclusions in Section 5.

## 2 IBM Worldwide Translation Operations

IBM translates on the order of 0.4 billion words per year in over 60 language-pairs managed from about 24 Translation Services Centers distributed across the globe and working with about 115 translation suppliers (who actually hire and manage the translation workforce). The content spans a wide variety of genres from publications (e.g. manuals) referred to as nonPII, to Product Integrated Information (PII) such as menus in software packages, to marketing material, and legal/safety contracts, etc.

In this study, we will run both nonPII and PII content. We anticipated that nonPII content would be the most impacted by machine translation since this represents more typical well structured language content. Sentences are expected to be well formed and declarative in nonPII content in contrast to menu entries or program comments which may not be as well structured.

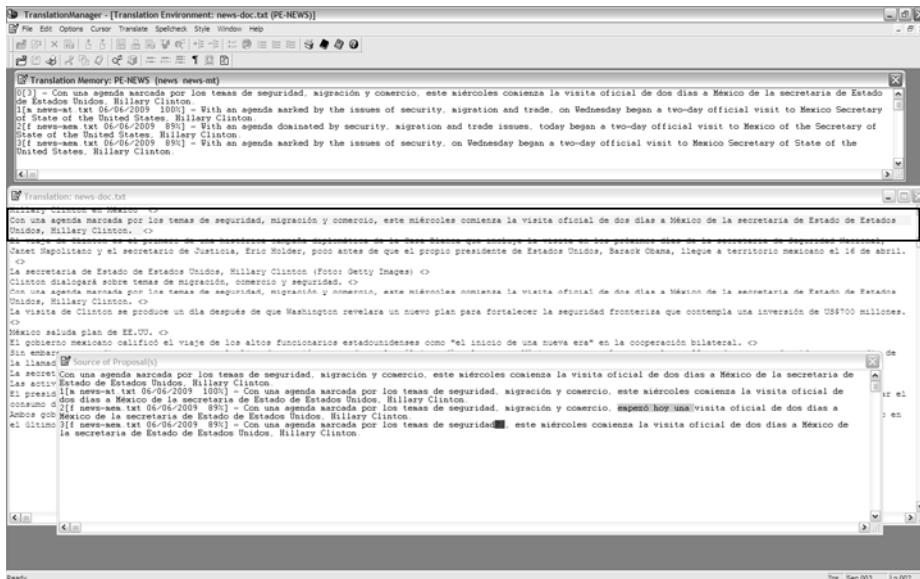


Fig. 1. TranslationManager 2 User Interface for Spanish-to-English translation

**TranslationManager 2.** All IBM translation suppliers use the TranslationManager 2 (TM2)<sup>2</sup> an IBM created tool to handle TMs and Post-Editing by a translator to create the target content.

Figure 1 shows a typical screen shot of TM2<sup>3</sup>. The middle pane shows the currently active segment (highlighted in grey and rectangle) for Spanish-to-English translation. This is the editing window. The translator can select one the 3 proposals shown in the top window or start from scratch to create the target English translation. The 3 proposals are: first, the MT proposal indicated by the letter “m” with 100% source match score, second, a fuzzy match indicated by “f” with a fuzzy match score of 89%, and a third fuzzy match with also a 89% FM score.

The third bottom pane displays the source sentences that correspond to each of the three English proposals. It also highlights in the FM proposals where the source sentence differs from the source for an FM proposal to help the translator identify which region to fix when they chose to post edit an FM proposal. This indicator is inapplicable to the MT proposal which requires the translator to focus on the produced MT to identify errors in the MT proposal. This represents a significant change to the Post-Editing activity that the translator has to adjust to.

TM2 has additional tools such as dictionary management for terminology and various reporting functions on a completed translation job.

TM2 projects are based on the concept of a folder which contains the actual input to be translated and optionally any TMs that can be useful for the specific project. These folders are shipped between IBM project managers and the translation vendors which effectively are using TM2 as a stand alone application. In order to integrate machine translation to support the current operational environment, we had to create offline the MT translations of the input and ship them as an additional machine-produced TM (hence the “m” indicator in the proposal window.).

Given the five interested parties in this effort, namely, the WTO project managers, the translation vendors, the translators, the SMT technology providers, and the customers (the group that wants the final translation product), a number of concerns had to be addressed to kickoff the in-vivo pilot. These concerns were:

- Preserving the quality of the produced translation. The concern being that translators may miss some of the MT errors in the Post-Editing process.
- Acceptance of a new mode of work by the translators – MT Post-Editing is a new skill that the translators have to acquire and may not be acceptable to some translators.
- Minimizing the risk to the productivity of the translators – the introduction of MT proposals may slow down the translators who are working in extremely competitive environment for producing translated content. We needed to have a financial mechanism to mitigate any downside risk to the translators.

---

<sup>2</sup> Note the similarity with the TM acronym for the generic Translation Memory.

<sup>3</sup> IBM recently released TM2 as open source software called OpenTM2. See [www.opentm2.org](http://www.opentm2.org)

- Impact on the work flow of project managers to manage their translation projects under the sometimes short deadlines required. MT introduces more latency in the current workflow.

We developed the following steps to address the above considerations:

- We created a training module to introduce the vendor and the selected translators to the new process of MT Post-Editing with hand-on experience. The training course was about 4 hours for a group of 4-6 translators at a time.
- We had one-on-one feedback sessions with a few selected translators to identify issues with the SMT output to improve the usefulness of MT proposals. These were conducted a couple weeks into a new project; the SMT technology team tried to address many of the issues. We discuss below some typical issues that popped up.
- Initially, we used a full payment approach meaning that IBM paid 100% per word cost even for FM segments (which typically are discounted since they are translated at a faster rate). This payment factor was seen as an incentive to protect against potential downside slowdown due to MT.
- We selected an easier language pair English-to-Spanish for the first pilot to improve the odds of positive impact due to SMT.
- We made an informal agreement with the vendors that any of the benefits due to customized SMT technology will be split somewhat evenly between the 3 players: the translation vendors (and indirectly the translators), the SMT technology provider, and the customer (as reduced per word charges).
- We also conducted quality checks to ensure that the produced content is no worse than what is usually obtained without MT.

The pilot was started in July 2009. We performed a full year analysis in Nov 2010. We then performed a 2<sup>nd</sup> year analysis covering the work during the July-November 2011 period.

TM2 can measure on a segment by segment basis the time the segment is active, the starting proposal type (whether an FM or MT proposal is used as a starting point or the translator decided to start from scratch), the number of character edits and word edits. TM2 also tracks the time if the translator comes back to a segment multiple times and the corresponding statistics for each additional visit. The detailed logging enabled us to track translator productivity for various types of segments.

The usual cycle for a translation project includes the following steps:

- 1) A project is identified by the customer. Typically a project has multiple shipments associated with it.
- 2) The Project Manager identifies relevant TMs and other related TMs; the PM ftp's the identified TMs to the MT service including optionally a terminology dictionary.
- 3) The MT service creates a customized MT engine using the identified TMs. The MT service also creates an MT TM that represents the machine translation of all non-exact matches in the input folder.
- 4) The PM ships the folder including the MT TM to the vendor.
- 5) The vendor ships the completed shipment with log files back to IBM.

For those shipments, where previous shipments have been completed, those shipments are used in customizing the engine for additional shipments from the same project. Step 3 is usually completed in less than 24 hours. Step 5 may take one to a few weeks depending on the size of the shipment. For all work in this project a minimum size of about 3,000 new words is used to send folders to the MT service.

### 3 Document - Specific Customization

In this section, we give a brief overview of the statistical machine translation system we use, called the Direct Translation Model [2], the customization method, and discuss some issues that needed to be addressed to improve the effectiveness of the MT for Post-Editing productivity.

#### 3.1 TM Customization

The Direct Translation Model framework [2] utilizes a Maximum Entropy model to guide the search over all possible target strings given a source sentence. The model  $p(T_c, j | S, T-, j-)$ , where  $T_c$  is the target language new substring to be produced and is concatenated with  $T-$  which is the target string produced up to this instant,  $j$  is the jump from the previous source position,  $j-$  are the previous jumps and  $S$  is the source sequence is formulated as an exponential model incorporating millions of features.

In the experiments below, for each shipment folder a training set is created by sub-sampling available parallel data including project specific memories to obtain a smaller training set that is relevant to the test. The parallel corpus size is thus reduced from several million sentence pairs to 300-500K sentence pairs for each shipment folder. Despite this reduction in training set size, the number of features utilized for each sub-sampled system is 10-20 million. This Document-Specific training set creates a customized model for the folder.

The decoder uses a multi-stack beam search to find the best target sentence where each extension is scored by a set of costs (including the MaxEnt score mentioned above) that are commonly found in phrase-based decoders. The system uses a large (trained on 5 billion words) modified Kneyser-Ney 5-gram language model together with a smaller in-domain model. This decoder has consistently placed well in NIST evaluations [3].

For the purposes of Post-Editing and based on translator feedback, we have modified the decoder to respect tagged regions which arise from utilizing xml to mark regions in text that is being translated. Generally, words inside these tagged regions can not be re-ordered to outside the region and vice-versa. In the Post-Editing task, for high fuzzy match sentences, we observe long phrases and to maximize performance on these sentences, we extract phrases up-to length 10.

To give a sense of the impact of TM customization we report the BLEU score for one folder using the NP subset to minimize any bias in BLEU estimation. The adapted DTM system achieves a BLEU score of 0.61 whereas a general baseline system available on the web (Google) achieves a BLEU score of 0.40 on the same test set. These BLEU score use one reference and 4-grams. The general system score of 0.40

indicates that the material is relatively easy. In addition, the customization yields a significant 50% relative improvement in MT quality as measured by BLEU.

### 3.2 MT Issues for MT Post-Editing

Based on the one-on-one feedback sessions with the translators, a number of issues were identified for the MT output. Here is a brief synopsis:

1. Trailing white space at end of sentences.
2. Preservation of presence or non-presence of white space around tags
3. Incorrect casing particularly in titles and headings
4. Identification of tagged content that should not be translated
5. Preserving the nesting of content relative to tags
6. Incorrect translations of some key terminology
7. Number and gender agreement
8. Incorrect grammatical structure

We show in Figure 2 an example of typical English content and its corresponding Spanish translation showing the mixed nature of content and tagged elements including white space management around tags. Notice the “.” after the tag in the 1<sup>st</sup> pair without a leading space. The second pair shows keywords place holders that will be evaluated later in the process. The third pair illustrates the usual text formatting markup of some words. The last pair shows a command example.

```

<Source>This edition replaces <ph
otherprops="tpcusersguideonly">SC27-2338-05</ph><ph
otherprops="messagesguideonly">SC27-2340-04</ph>.</Source>

<Target>Esta edición sustituye a <ph
otherprops="tpcusersguideonly">SC27-2338-05</ph><ph
otherprops="messagesguideonly">SC27-2340-04</ph>.</Target>

<Source>Additional storage information collected by a
subordinate server and used within such <keyword
conref="fqz0_entities.dita#fqz0_entities\ktpc_short"></keyword>
functions as the topology viewer, data path explorer, volume
provisioning, volume performance, <keyword
conref="fqz0_entities.dita#fqz0_entities\ksanp"></keyword> ,
etc. is available for that subordinate server only. </Source>

<Target>La información de almacenamiento adicional recopilada
por un servidor subordinado y que se utiliza dentro de las
funciones de dicho <keyword
conref="fqz0_entities.dita#fqz0_entities\ktpc_short"></keyword>
como visor de topología, explorador de vías de acceso de datos,
suministro de volumen, rendimiento de volumen, <keyword
conref="fqz0_entities.dita#fqz0_entities\ksanp"></keyword>, etc.
está disponible sólo para dicho servidor subordinado. </Target>

```

**Fig. 2.** Sample of 4 segment pairs with English source and its target translation in Spanish

```

<Source>In the <uicontrol>Triggered Actions</uicontrol> area,
select <uicontrol>Archive/Backup</uicontrol> and click
<uicontrol>Define</uicontrol>. </Source>

<Target>En el área <uicontrol>Acciones
desencadenadas</uicontrol>, seleccione
<uicontrol>Archivado/Copia de seguridad</uicontrol> y haga clic
en <uicontrol>Definir</uicontrol>. </Target>

<Source>tpctool&gt; -user me -pwd mypass -url myhost:myport
</Source>

<Target>tpctool&gt; -user me -pwd mypass -url myhost:myport
</Target>

```

**Fig. 2. (continued)**

We targeted fixing the issues around bullets 1 to 5 and made significant progress towards solving them. For bullet 6, we integrated a terminology dictionary in the translation engine including regular updates of the dictionary. Bullets 7 and 8 are the more generic core MT issues that only improve as core algorithms improve, unfortunately at a much slower rate of progress.

## 4 Experimental Results

We conducted two studies for MT Post-Editing (MTPE) effectiveness. The first, the “2010 Study” was over about 10 months in 2010. The second, the “2011 Study”, was an analysis of the workflow in production over the period of 5 months from July to November, 2011.

We first discuss the 2010 Study which consisted of 144 shipments that were processed, each having its own Document-Specific customization, over the 10 months period. A pool of 15 translators participated in the first MTPE study for English-to-Spanish.

### 4.1 2010 MTPE Study

Table 1 gives the word count for the 144 folders in the study categorized into the 3 main categories of EM, FM, and NP for a total of about 3.7 Mwords. All the content in the 2010 Study was of the nonPII type (publications) which was considered to be the easier type for MT processing.

**Table 1.** Gross word count and percentages for the English-to-Spanish 2010 Study

EnEs	Num Words	%
EM	1,499,822	40%
FM	1,365,372	36%
NP	883,505	24%
Total	3,748,699	100%

As we indicated earlier, TM2 creates a log that shows the time duration when a segment is active (including multiple visits by the translator as it sometimes happens). The data are censored by removing those segments that are not part of the normal workflow; for example, the translator may take a break, a phone call, etc. Also, there are segments that are completed at a much faster rate than expected as happens when the input is copied verbatim into the target output such as would be the case for a code snippet. To minimize the impact of these spurious effects on the data, we removed two types of segments:

- 1) the segments with the editing time more than 10 minutes, and
- 2) the segments where the total number of characters typed is less than two times the number of source words (these typically correspond to source code snippets or commands that are handled by keeping the input unmodified. The chosen censoring removes about 29% of the words.

We also remove the EM segments since they do not have MT and are not analyzed further. The remaining 1.6M words are further divided into 4 categories to enable controlled measurement of productivity.

To analyze the impact of MT we needed to control for the presence or lack thereof of MT proposals. So we created a “control” set where we randomly remove the MT proposal for a small fraction of the segments in a folder. We do this both for the FM and NP cases and we denote them by FM0 and NP0; we denote by FM1 and NP1 the cases where the MT proposal is present. The control set represents the original case of using TM2 without MT proposals. The control case identifies the base numbers that

**Table 2.** Word count for the control and MT cases. We also show the percentage of the control sets for the FM and NP segments.

EnEs	Num Words	%
FM0	189,722	19%
FM1	818,065	81%
NP0	110,099	18%
NP1	485,274	82%

we compare to the case with MT proposals. Table 2 shows the censored word counts for the 4 cases of interest. The control sets are about 19% and 18% of the words for the FM and NP cases, respectively.

In addition to the four aforementioned categories, we classified the segments further by the action of the translator. For an FM0 segment, the translator can either start with the FM proposal or start from scratch in producing the target translation, which we indicate by FM0\_FM and FM0\_NP, respectively. For FM1 segments the translator has 3 options denoted FM1\_FM, FM1\_MT, and FM1\_NP, the middle one indicating the case where the translator starts with the MT proposal (instead of the FM proposal for instance). Table 3 shows the word count and the percentage within each category for each of the refined classifications. As can be seen, translators chose to create a target from scratch in the FM0 case for 55% of the words. In their judgment, they decided that it would be faster to translate from scratch than Post-Editing the FM proposal.

Significantly, we note that the translators chose to start with the MT proposal for 43% of the words for the FM1 case indicating that they estimate that for those segments it would be faster to start with the MT proposal than either the FM proposal or starting from scratch (NP). We also note that for the NP1 case, they started with the MT proposal for 71% of the words. The translators are voting for MT with their fingers!

We define translator productivity as the number of words per unit time. Using a normalized unit of 1 for the productivity of a translator for the NP0 (starting from scratch) sentences without MT present, the productivity for EM segments is 22. The EM speedup compared to the NP is not too surprising since it is mostly reading and checking.

**Table 3.** Word count and percentages for all the English-to-Spanish folders in Pilot 1 categorized by the translator action

EnES nonPII	-	% in Category
	Num Words	
FM0_FM	85152	45%
FM0_NP	104570	55%
FM1_FM	228085	28%
FM1_MT	353464	43%
FM1_NP	236516	29%
NP0_NP	110099	100%
NP1_MT	344050	71%
NP1_NP	141224	29%

**Table 4.** Productivity and its increase due to MT for the English-Spanish nonPII content

EnEs	Num Words	%	Productivity	MT Improvement
FM0	189,722	19%	1.36	
FM1	818,065	81%	1.86	36%
NP0	110,099	18%	1.00	
NP1	485,274	82%	1.50	50%

Table 4 shows the actual productivity for each of the four cases case normalized by the productivity of NP0 (which is the case of creating a translation from scratch without any technology help). The FM0 productivity increases to 1.36 (a relative increase of 36%) over that of NP0, a significant effect that justifies the use of TM technology tools and the corresponding requirement of the management of TMs across all translation projects for an enterprise. Table 4 also shows the increase in productivity due to the presence of MT proposals. As can be seen, MT leads to another 36% productivity increase for FM segments and 50% for NP segments.

The improvement for FM segments due to MTPE is a significant surprise given that in this case a burden is added for the translators to look at one more proposal of a very different nature than the FM proposals. As we saw in Table 3, the presence of the MT proposal reduced the rates of translators starting with either the FM or NP to produce the target. The rate for FM decreased from 45% to 28% and that for starting with NP from 55% to 29%. The balance is obviously starting with the MT proposal.

**Quality.** Given the concern about the impact of quality in MTPE, a number of folders were selected in the early part of the study, where few hundred sentences were randomly scanned for translation quality. It was determined that the quality was similar to or better than the previous TM based output. So if anything, expected quality is improved when MT is present.

The results of the 2010 Study formed the basis of a new pricing agreement with translation vendors for a production deployment of Document-Specific SMT adaptation. The production solution went live in June 2011.

## 4.2 2011 MTPE Study

We conducted a second study of translator productivity for a 5-month period from July to November 2011 for the deployed production solution. The results of this study are presented next. We analyzed both PII and nonPII content in the 2011 Study.

**Table 5.** NonPII gross word count and percentages for the English-Spanish MTPE 2011 study

EnEs	Num Words	%
EM	473,694	18%
FM	1,029,091	40%
NP	1,062,882	41%
Total	2,565,667	100%

**NonPII Content.** Table 5 shows the gross word count for the nonPII content for the three main categories. Table 6 shows the productivity for the four segment types of FM0, FM1, NP0, and NP1 using the same censoring approach described in Section 4.1. It also shows that for nonPII content MT increases productivity by 38% for FM segments and a whopping 68% for NP segments. Note that the FM0 segments are 36% faster than NP0, as was obtained in the 2010 Study.

**Table 6.** Productivity and its increase due to MT for the English-Spanish nonPII content

EnEs	Num Words	%	Productivity	MT
				Improvement
FM0	272145	31%	1.36	
FM1	609065	69%	1.88	38%
NP0	116346	18%	1.00	
NP1	843947	88%	1.68	68%

We were surprised by the relative increase in productivity for the NP1 case between the 2010 Study at 50% and the 2011 Study at 68%. It is hard to know the reason since we do not have controlled conditions to determine if the material is easier for MT or not. But we conducted a more detailed analysis of the production data where we collected for every segment in FM1 and NP1 the TER<sup>4</sup> (Translation Error Rate) between the MT and the final target output. In the case of those segments where the translator used the MT proposal, this would correspond to the HTER (Human-targeted Translation Error Rate [4]).

We also looked at the set of segments that belong to FM1\_MT and NP1\_MT which correspond to those segments where the translators decided to start from the MT proposal for their Post-Editing work. We measured the correction effort using HTER to be 26% and 28% for FM1\_MT and NP1\_MT, respectively. We also compared the relative productivity for these segments and found the translators to produce about 17% fewer words per unit time for the NP1\_MT case versus the FM1\_MT case suggesting a bit more effort is required for the NP1\_MT sentences even though the final HTER is only slightly higher (8% relative) for these sentences.

<sup>4</sup> Translation Edit/Error Rate; see [4] for more details.

**Table 7.** TER between the final target sentence and the MT proposal

EnEs	TER/HTER
FM1_FM	28%
FM1_MT	26%
FM1_NP	35%
FM1	28%
NP1_MT	28%
NP1_NP	41%
NP1	31%

We show in Table 7, the TER between the final target sentence and the corresponding MT proposal. Note that for FM1\_MT and NP1\_MT the TER would be HTER [4] in this case. The FM1 and NP1 rows show the overall average TER for each category.

**Table 8.** PII gross word count and percentages for the English-Spanish MTPE study

EnEs	Num Words	%
EM	248,300	38%
FM	276,902	42%
NP	136,809	21%
Total	662,011	100%

**PII Content.** We also processed PII content as part of the production deployment. Table 8 gives the gross word count for PII content. These numbers are still relatively small, but they give an initial indication on whether MT is useful for the more challenging PII content. Table 9 gives the productivity for the four segment types of FM0, FM1, NP0, NP1 again normalized by the NP0 productivity for PII content (which is 0.58 times that of nonPII indicating the more time consuming nature of PII content).

As can be seen in Table 9, FM0 productivity is 2.52 a 152% faster than NP0 productivity; MT adds another 49% speedup.

For NP1 MT increases the productivity by an unexpected 176%! While we still think that these are small data sets, it appears that MT is even more useful for PII content than nonPII.

**Table 9.** Productivity and its increase due to MT for the English-Spanish PII content

EnEs	Num Words	%	Productivity	MT Improvement
FM0	150124	72%	2.52	
FM1	57475	28%	3.76	49%
NP0	48334	47%	1.00	
NP1	53899	53%	2.76	176%

## 5 Conclusion

We presented the results of two studies that are based on a few thousands of hours of human translation time to create few millions of words of translated content. Given the large scale of these studies we believe that the productivity results are quite reliable for the kind of content typical of an IT company needs. Document Specific DTM customization has proven to yield significant increase in human translation productivity about 50% for the first study and 68% for the second. It is hard to characterize the reason for the increased efficiency in the second study; whether it is due to improved core MT algorithms, increased relevant TMs, or translators becoming more efficient at Post-Editing. We guess all 3 factors are at play.

Another interesting finding is that even for the more terse PII content we find the MT Post-Editing yields big increases in productivity on the order of 176% though we caution that these are preliminary results given the relatively smaller set size over which this estimate is based.

Contrary to other practitioners in the field of using MT for MT Post-Editing, we have found that adding the MT proposal as an additional proposal to the more predictable (to the translator) Fuzzy Match proposals to yield a net gain in productivity in spite the added burden for having one more proposal to assess. The productivity increased by 38% for nonPII content and a larger figure of 49% for PII content.

The above results are significant and need to be validated for other languages<sup>5</sup>. They indicate a significant niche for the use of automation tools such as MT for the human translation business. We expect that major evolutionary steps will happen in the coming few years due to the dramatic change in productivity. In addition, given the recent rate of progress in core statistical machine translation, we anticipate that MT improvements to lead to additional improvements in productivity of Post-Editing.

Finally, we expect that due to the almost completely automated approach to customization in SMT, hence a rather small associated cost, the pervasiveness of

<sup>5</sup> We are currently running MT Post-Editing studies for French, Italian, Brazilian Portuguese, Simplified Chinese, Japanese, and German. We see significant improvements for most languages (except German) though the volumes are still relatively small to draw firm conclusions.

SMT for Post-Editing will increase significantly spanning not only larger projects as has been the case so far but cover a very broad spectrum of translation projects including very small projects.

**Acknowledgments.** We would like to thank our IBM colleagues for their cooperation and support to run these MT Post-Editing studies; specifically Alex Martínez Corriá, IBM Translation Services Center Manager – Iberia, Santi Pont Nesta, Project Manager at IBM TSC – Iberia, Frank Rojas, IBM Globalization Team, and other members of the IBM Globalization team. We also want to thank the translation vendor Sajan and their translators for their open mindedness and enthusiasm for this project.

## References

1. Common Sense Advisory Releases In-depth Analysis of the Translation and Localization Industry, <http://www.commonsenseadvisory.com/Default.aspx?Contenttype=ArticleDet&tabID=64&moduleId=392&Aid=1158&PR=PR>
2. Ittycheriah, A., Roukos, S.: Direct translation model 2. In: HLT-NAACL, pp. 57–64 (2007)
3. NIST 2008 Open Machine Translation Evaluation - (MT08), Arabic to English (primary system) Results, Constrained and Unconstrained Training Tracks, [http://www.itl.nist.gov/iad/mig/tests/mt/2008/doc/mt08\\_official\\_results\\_v0.html](http://www.itl.nist.gov/iad/mig/tests/mt/2008/doc/mt08_official_results_v0.html)
4. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223–231 (2006)

# Minimum Bayes Risk Decoding with Enlarged Hypothesis Space in System Combination

Tsuyoshi Okita and Josef van Genabith

Dublin City University, School of Computing, Glasnevin, Dublin 9, Ireland

**Abstract.** This paper describes a new system combination strategy in Statistical Machine Translation. Tromble et al. (2008) introduced the evidence space into Minimum Bayes Risk decoding in order to quantify the relative performance within lattice or n-best output with regard to the 1-best output. In contrast, our approach is to enlarge the hypothesis space in order to incorporate the combinatorial nature of MBR decoding. In this setting, we perform experiments on three language pairs ES-EN, FR-EN and JP-EN. For ES-EN JRC-Acquis our approach shows 0.50 BLEU points absolute and 1.9% relative improvement over the standard confusion network-based system combination without hypothesis expansion, and 2.16 BLEU points absolute and 9.2% relative improvement compared to the single best system. For JP-EN NTCIR-8 the improvement is 0.94 points absolute and 3.4% relative, and for FR-EN WMT09 0.30 points absolute and 1.3% relative compared to the single best system, respectively.

## 1 Introduction

In a sequence prediction task, a max-product algorithm (or Viterbi decoding [29]) is a standard technique to find an approximate solution  $x$  which maximizes the joint distribution  $p(x)$  (while a sum-product algorithm [23] attempts to find an exact solution  $x$ ). Max-product is an inference algorithm for a single model in a tree or a chain structure [13]. Suppose that we consider a combination of multiple systems whose model parameters are different. The first problem is that we are required to calibrate the quantities coming from the different models since these quantities are not immediately comparable in general. The second problem is that it is often the case that an increase in the number of participating systems increases the overall computation in a non-linear way; fortunately, however, it turns out that often a lot of calculations are redundant over systems at the same time. In our particular situation, the number of nodes increases exponentially since the corresponding nodes are searched in a combinatorial manner (even though the overall number of system is small); however, there are a lot of redundancies.

In order to address these problems, this paper imposes practical assumptions limiting our scope but in such a way that our immediate application of Minimum Bayes Risk decoding [14] does not suffer.<sup>1</sup> Our assumptions are that

---

<sup>1</sup> Note that it is not clear what kind of other applications exist.

(i) the model structures are almost identical and that (ii) the probabilities which we compare are indexed and thus can be calibrated locally. Under this assumption, it turned out that we can employ a standard MAP assignment algorithm [13] to calibrate the probabilities arising from different systems, even though the original aim of normalization of MAP assignment is different in that the unnormalized probabilities arise by themselves since MAP assignment partitions variables into  $E$ (evidence),  $Q$ (query), and  $H$ (hidden) variables. Clique tree [24] is a technique to consider only some factors locally, which can be applied here.

With these preparations, we develop a new system combination strategy using Minimum Bayes Risk (MBR) decoding [14] which exploits a larger hypothesis space. A system combination strategy [2,16,6] is a state-of-the-art technique to improve the overall BLEU score. Recently, Tromble et al. [28] attempted to exploit a larger evidence space by using a lattice structure. DeNero et al. [4,5] introduced n-gram expectation, while Arun et al. [1] compared MBR decoding with MAP decoding for general translation tasks in a MERT setting [17].

The remainder of this paper is organized as follows. Section 2 reviews the decoding algorithm in SMT. Section 3 describes our algorithm. In Section 4, our experimental results are presented. We conclude in Section 5.

## 2 Decoding Algorithm in SMT

There are two popular decoding algorithms in phrase-based SMT: MAP decoding and MBR decoding [10]. MAP decoding is the main approach in phrase-based SMT [12], while MBR decoding is mainly used for system combination [2,16,6,28,4]. The MAP decoding algorithm seeks the most likely output sequence, while the MBR decoding seeks the output sequence whose loss is the smallest.

Let  $E$  be the target language,  $F$  be the source language,  $A$  be an alignment which represents the mapping from source to target phrases, and  $M(\cdot)$  be an MT system which maps some sequence in the source language  $F$  into some sequence in the target language  $E$ . MAP decoding can be written as in (1):

$$\hat{E}_{best}^{MAP} = \arg \max_E \sum_A P(E, A|F) \quad (1)$$

Let  $\mathcal{E}$  be the translation outputs of all the MT systems. For a given reference translation  $E$ , the decoder performance can be measured by the loss function  $L(E, M(F))$ . Given such a loss function  $L(E, E')$  between an automatic translation  $E'$  and the reference  $E$ , a set of translation outputs  $\mathcal{E}$ , and an underlying probability model  $P(E|F)$ , a MBR decoder is defined as in (2) [14]:

$$\begin{aligned} \hat{E}_{best}^{MBR} &= \arg \min_{E' \in \mathcal{E}} R(E') \\ &= \arg \min_{E' \in \mathcal{E}_H} \sum_{E' \in \mathcal{E}_E} L(E, E') P(E|F) \end{aligned} \quad (2)$$

$$= \arg \max_{E' \in \mathcal{E}_H} \sum_{E' \in \mathcal{E}_E} BLEU_E(E') P(E|F) \quad (3)$$

where  $R(E')$  denotes the Bayes risk of candidate translation  $E'$  under the loss function  $L$ ,  $\text{BLEU}_E(E')$  [22] is a function to evaluate a hypothesis  $E'$  according to  $E$ ,  $\mathcal{E}_H$  refers to the hypothesis space from which translations are chosen,  $\mathcal{E}_E$  refers to the evidence space used for calculating risk. Note that a hypothesis space  $\mathcal{E}_H$  and an evidence space  $\mathcal{E}_E$  appeared in [9,28,4,1].

The confusion network-based approach [2,16,6] enables us to combine several fragments from different MT outputs. In the first step, we select the sentence-based best single system via a MBR decoder (or single system outputs are often used as the backbone of the confusion network). Note that the backbone determines the general word order of the confusion network. In the second step, based on the backbone which is selected in the first step, we build the confusion network by aligning the hypotheses with the backbone. In this process, we used the TER distance [25] between the backbone and the hypotheses. We do this for all the hypotheses sentence by sentence. Note that in this process, deleted words are substituted as NULL words (or  $\epsilon$ -arcs). In the third step, the consensus translation is extracted as the best path in the confusion network. The most primitive approach [16] is to select the best word  $\hat{e}_k$  by the word posterior probability via voting at each position  $k$  in the confusion network, as in (4):

$$\hat{E}_k = \arg \max_{e \in \mathcal{E}} p_k(e|F) \quad (4)$$

Note that this word posterior probability can be used as a measure how confident the model is about this particular word translation [10], as defined in (5):

$$p_i(e|F) = \sum_j \delta(e, e_{j,i}) p(e_j|F) \quad (5)$$

where  $e_{j,i}$  denotes the  $i$ -th word and  $\delta(e, e_{j,i})$  denotes the indicator function which is 1 if the  $i$ -th word is  $e$ , otherwise 0. However, in practice as is shown by [6,15], the incorporation of a language model in this voting process will improve the quality further. Hence, we use the following features in this voting process: word posterior probability, 4-gram and 5-gram target language model, word length penalty, and NULL word length penalty. Note that Minimum Error-Rate Training (MERT) is used to tune the weights of the confusion network. In the final step, we remove  $\epsilon$ -arcs, if they exist.

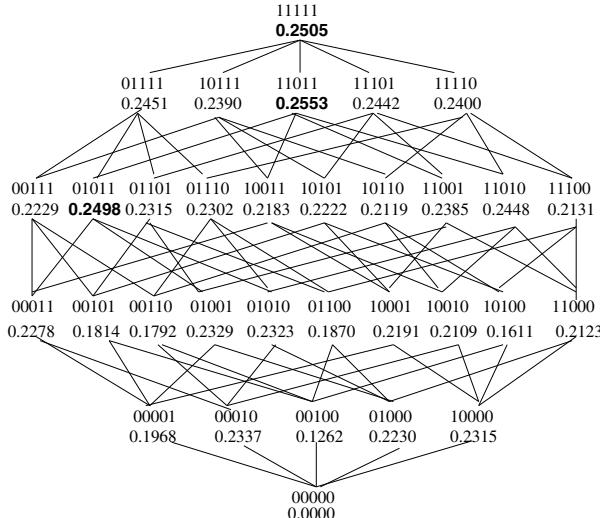
### 3 Our Algorithm

Tromble et al. [28] introduced a lattice in the evidence space into Minimum Bayes Risk decoding in order to quantify the relative performance within lattice or n-best output with regard to the 1-best output. In contrast, our approach is to enlarge the hypothesis space via different kinds of lattices in order to incorporate the combinatorial nature of MBR decoding.

We first present the motivation for using the enlarged hypothesis space and searching for the optimal subset  $\mathcal{E}_0$  among this enlarged hypothesis space  $\mathcal{E}$  (where  $\mathcal{E}$  is the translation outputs of all the MT systems participating in the

**Table 1.** Motivating examples. MBR decoding can be schematically described as maximizing the n-gram expectations between the MT output sequence and some sequence, as is described in this table. The left table shows the MT output sequences consisting of 5 systems, while the right table shows the MT output sequences consisting of 4 systems. In this case, the 1-gram expectation of “bbcd” for 4 systems (right table) are better than those for 5 systems (left table). This suggests that it may be better to remove extremely bad MT output from the inputs of system combination.

A	MT outputs	prob	1-gram expectation	B	MT outputs	prob	1-gram expectation
1	a a a c	0.30	$E_A(aaac)=1.2$	1	a a a c	0.33	$E_B(aaac)=1.32$
2	b b c d	0.20	$E_A(bbcd)=2.1$	2	b b c d	0.22	$E_B(bbcd)=\mathbf{2.20}$
3	b b b d	0.20	$E_A(bbbd)=2.0$	3	b b b d	0.22	$E_B(bbbd)=1.98$
4	b b c f	0.20	$E_A(bbcf)=1.8$	4	b b c f	0.22	$E_B(bbcf)=1.98$
5	f f b d	0.10	$E_A(ffbd)=1.0$	5	- - - -	0.00	



**Fig. 1.** Figure shows the lattice of five MT output sequences encoded as binary sequences (‘11111’, ‘01111’, etc) and BLEU scores (‘0.2505’, ‘0.2451’, etc) for ES-EN JRC-Acquis (Refer Table 2). The top row shows the results using five MT output sequences; the second row uses four MT output sequences; . . . ; the fourth row uses the individual BLEU scores; the bottom row does not use any MT output sequence (Hence, BLEU score is zero). The observation from this lattice is that the resulting BLEU score is not always between two BLEU scores of adjacent nodes; sometimes the resulting BLEU score is lower than both of them (e.g. ‘00010’ and ‘10000’ resulted in 0.2109.) and it is higher than both of them (e.g. ‘00011’, ‘01001’ and ‘01010’ resulted in 0.2498.) The maximal value in the lattice is 0.2553 in the second row in this case.

system combination). The focus is on  $E$  of  $P(E|F)$  in Eq (2) where  $E$  is a set of MT outputs participating in the system combination. That is, if we combine four systems the number of systems, that is  $|E|$ , is four. A toy example is shown in Table 1. In this example, five MT output sequences “aaac”, “bbcd”, “bbbd”, “bbcf”, and “ffbd” are given. Suppose that we calculate the 1-gram expectation of “bbcd”, which constitute the negative quantity in Bayes risk. If we use all the given MT outputs consisting of 5 systems, the expected matches sum to 2.1. If we discard the system producing “ffbd” and only use 4 systems, the 1-gram expectation improves to 2.20. As a conclusion, it is not always the best solution to use the full set of given MT outputs, but removing some bad MT output can be a good strategy. This suggests to consider all possible subsets of the full set of MT outputs, as is shown in (7):

$$\hat{E} = \arg \min_{\mathcal{E}_i \subseteq \mathcal{E}} \sum_{E' \in \mathcal{E}_i} L(E, E') P(E|F) \quad (6)$$

$$= \arg \min_{E' \in \mathcal{E}_{H_i}, \mathcal{E}_{H_i} \subseteq \mathcal{E}} \sum_{E' \in \mathcal{E}_{E_i}} L(E, E') P(E|F) \quad (7)$$

where  $\mathcal{E}_{H_i} \subseteq \mathcal{E}$  indicates that we choose  $\mathcal{E}_{H_i}$  from all the possible subsets of  $\mathcal{E}$  (or a power set of  $\mathcal{E}$ ),  $\mathcal{E}_{H_i}$  denotes a  $i$ -th hypothesis space, and  $\mathcal{E}_{E_i}$  denotes a  $i$ -th evidence space corresponding to  $\mathcal{E}_{E_i}$ .<sup>2</sup>

Now we explain how to formulate an algorithm. As is explained in the latter half of Section 2, a confusion network-based system combination approach takes three steps<sup>3</sup> as follows.

1. Choosing a backbone by a MBR decoder from MT outputs  $S$ .
2. Measure the distance between the backbone and each output.
3. Run the decoding algorithm to choose the best path in the confusion network.

Let  $|S| = n$ . If we consider all the combinations of  $|S|$ , the simplest algorithm which enumerates all the possibilities requires to repeat these three steps  $2^n - 1$  times. However, if we observe this computation we can immediately recognize that there are a considerable number of redundant operations. Hence, our approach is to reduce such redundant operations. First we observe what is changed in these three steps by considering a combinatorial exploration.

- Due to the combinatorial exploration of MT outputs of  $|S|$  cases, all the MT outputs can be selected as a backbone for some combination of  $S$  in theory. However, if we exclude the combination of using only one or two MT outputs, two cases remain important which have high chances to result in the backbone in most of the cases: the output with the highest BLEU score and that the MBR decoding selects the MT output with highest density (when many MT outputs include the segment).

<sup>2</sup> A power set of  $\mathcal{E} = \{1, 2\}$  is  $\{\{1, 2\}, \{1\}, \{2\}, \emptyset\}$ .

<sup>3</sup> In Section 2, we described the final step. However, this step is just to remove deletion marks and is omitted here.

- Under the combinatorial exploration strategy, what we need to care about is the unnormalized probabilities in the word posterior probabilities. Note that the word posterior probabilities  $P(e_j|F)$  in Eq (5) will not vary even if we take the scheme of combinatorial exploration.
- Other quantities, such as language model, word length penalty, and NULL word length penalty will not be changed.

Following on from the second point above, we transform the parallel trees of several MT outputs into a so-called clique tree [13], as is shown in Figure 2. In this clique tree, each clique tree contains the corresponding word pairs in confusion networks. By this transformation, we can reduce the message cost considerably in the third step of decoding to choose the best path in the confusion network, where a message is to connect a node and neighboring node.

Hence, the primitive version which computes all the combinations one by one, takes  $O(|S| \times n|T|)$  execution time in the third step where  $|T|$  denotes the number of message passing events which is equivalent to the  $n$  times the length of the clique tree. Compared to this, the version which uses a clique tree can reduce this message costs from  $n|T|$  to  $|T|$ , hence the overall cost becomes  $O(|S| \times |T|)$ . If we apply the max-product algorithm, the computation in the clique, which is  $O(|S|)$ , may be reduced further.

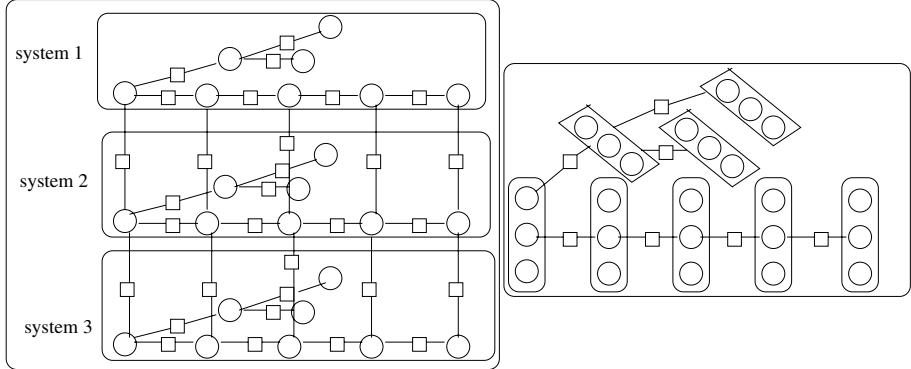
Message passing is done in the clique one by one propagating from the root to the leaf. Let  $C_i$  and  $C_j$  be the neighboring clique in a clique tree. The value of the message sent from  $C_i$  to  $C_j$  does not depend on the specific choice of root clique. This argument applies in both directions (p.355 of [13]). Hence, the message from  $C_i$  to another clique  $C_j$ , denoted as  $\delta_{i \rightarrow j}$ , can be written as (8):

$$\delta_{i \rightarrow j} = \max_{C_i - S_{i,j}} \phi_i \prod_{k \in (Nb_i - \{j\})} \delta_{k \rightarrow i} \quad (8)$$

where  $\phi_i$  denotes a factor in clique  $i$ , and  $Nb_i$  denotes the set of indices of cliques that are neighbors of  $C_i$ . This message passing process proceeds up the tree. When the root clique has received all messages, it multiplies them with its own initial potential.

## 4 Heuristic Algorithm

The second algorithm is intended to provide one of the baselines. Suppose we are given 5 translation outputs (the top node marked with ‘11111’ in Fig. 1) and we traverse from this node to the bottom node in a breadth first manner where we only measure the BLEU score on trajectory nodes. Suppose also that we know in advanced each single BLEU score of each translation output (‘00001’ to ‘10000’). The first task is to predict which children of ‘11111’ attains the best BLEU score among its siblings (‘01111’ to ‘11110’). We choose the combination (‘11011’) removing a worst single translation output (‘00100’) will attain the best BLEU score. Then, we measure and compare the actual BLEU score of the parent node and only this child node. (We do not measure the BLEU score of other siblings). If there is an



**Fig. 2.** Figures show a max-product algorithm on multiple systems under two assumptions described in Introduction. In the figure, a circle denote a variable node, a square denote a factor node, and a big rectangle denote a system (in the left figure) and a clique (in the right figure).

---

#### Algorithm 1. Heuristic Algorithm

---

**Given:** A set of MT devset output  $S = \{s_1, \dots, s_n\}$  and MT testset output  $T = \{t_1, \dots, t_n\}$ .

**Step 1:** Rank devset outputs  $S$  according to the performance measure (BLEU, TER, etc) as  $S' = \{s'_1, \dots, s'_n\}$  where  $s'_i \prec s'_{i+1}$  (the rank of  $s'_i$  is higher than (or the same as)  $s'_{i+1}$ ).

**Step 2:**  $i$  iteration: Discard the worst system  $i$  of  $S'$  to make  $S'_{(i)}$ .

**Step 3:** Measure the performance of  $S'_{(i)}$ .

**Step 4:** If  $M(S'_{(i)}) > M(S'_{(i-1)})$  then repeat Step 2.

**Step 5:** Reply the correspondent MT testset output  $T$  with regard to  $S'_{(i)}$ .

---

increase, we repeat this process until we reach the bottom node. If we observe decrease, we judge that the parent node attains the best BLEU score. This is shown in Algorithm 1. Although this starts from the full set (of MT systems in a combination) to the empty set (We refer this as Heuristic 1), it is also possible to take the reverse direction which starts from the singleton set to the full set (We refer this as Heuristic 2). There have been no quantitative predictions as far as we are aware.

## 5 Experiments

We used three different language pairs in our experiments. The first set is ES-EN based on JRC-Acquis [26]; we use the translation outputs of 5 MT systems provided by [7]. The second set is JP-EN provided by NTCIR-8 [8] where translation outputs are prepared by ourselves [20]. The third set is EN-FR

**Table 2.** Experiment between ES and EN for JRC-Acquis dataset. All the scores are on testset except those marked \* (which are on devset). On comparison, we did sampling of three combinations of the single systems, which shows that our results are equivalent to the combination 2. These experimental results validate our motivating results: it is often the case that some radically bad translation output may harm the final output by system combination. In this case, system t3 whose BLEU score is 12.62 has a negative effect on the results of system combination. The best performance was achieved by removing this system, i.e. the combination of systems t1, t2, t4, and t5. The baseline obtained the best score at ‘01000’, the heuristic algorithm obtained at ‘11011’, and our algorithm obtained at ‘11011’.

	NIST	BLEU	METEOR	WER	PER
system t1 (‘10000’)	6.3934	0.1968/0.1289*	0.5022487	62.3685	47.3074
system t2 (‘01000’)	6.3818	0.2337/0.1498*	<b>0.5732194</b>	64.7816	49.2348
system t3 (‘00100’)	4.5648	0.1262/0.0837*	0.4073446	77.6184	63.0546
system t4 (‘00010’)	6.2136	0.2230/0.1343*	0.5544878	64.9050	50.2139
system t5 (‘00001’)	6.7082	0.2315/0.1453*	0.5412563	60.6646	45.1949
baseline	6.3818	0.2337	0.5732194	64.7816	49.2348
heuristic 1	6.8419	0.2553	0.5683086	59.9591	44.5357
heuristic 2	6.3818	0.2337	0.5732194	64.7816	49.2348
<b>our algorithm</b> (‘11011’)	<b>6.8419</b>	<b>0.2553</b>	0.5683086	<b>59.9591</b>	<b>44.5357</b>

provided by WMT09 [3]. We use MERT [17] internally to tune the weights and language modeling by SRILM [27].

Tables 2, 3, and 4 include first the BLEU score of individual systems, and then show four results: baseline, heuristic 1 and 2 (Refer Section 4), and our algorithm (Refer Section 3). The baseline is the BLEU score of the best single system.

Table 2 shows our results from ES to EN. The improvement in BLEU was 2.16 points absolute and 9.2% relative compared to the performance of system t2, the single best performing system (we optimized according to BLEU). Except for METEOR, we achieved the best performance in NIST (0.14 points absolute and 2.1% relative), WER (0.71 points absolute and 1.1% relative) and PER (0.64 points absolute and 1.3% relative) as well. However, in this case, Heuristic 1 also achieved the same result. The heuristic algorithm 1 was processed from the point ‘11011’ (BLEU 0.2553) to ‘11001’ (0.2385). The result of heuristic algorithm 1 was 0.2553.

The left half of Table 3 shows our results from JP to EN. The improvement in BLEU was 0.94 points absolute and 3.4% relative compared to the single best performing system. Heuristic 2 and baseline shows the result of system t2. The baseline obtained the result at ‘010000000000’, the heuristic algorithm 1 at ‘11001111101’, the heuristic algorithm 2 at ‘010000000000’, and our algorithm at ‘11100010101’. The heuristic algorithm 1 was processed from the point ‘11011111111’ (BLEU 0.2202) to ‘11011111101’ (0.2750), ‘11001111101’ (0.2750), and ‘11001110101’ (0.2345). The result of heuristic algorithm 1 was 0.2750. The right half of Table 3 shows the results from EN to FR. The improvement in

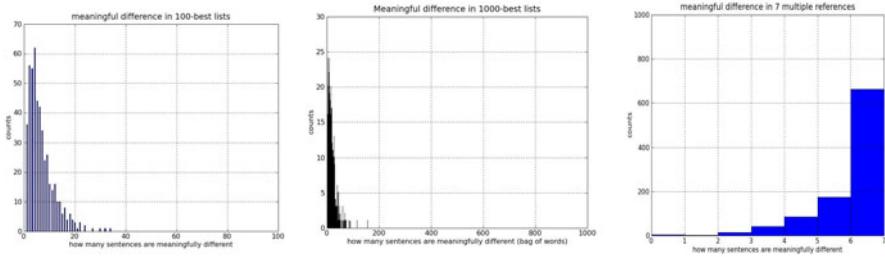
**Table 3.** (Left half) Experiment between JP and EN for NTCIR dataset. The baseline obtained the result at ‘01000000000’, heuristic algorithm 1 was at ‘1100111101’, heuristic algorithm 2 was at ‘01000000000’, and our algorithm obtained at ‘11100010101’. In this combination, system t3 of BLEU score 0.1243 is included which can be explained that . (Right half) Experiment between EN and FR for WMT 2009 devset. The baseline and the heuristic 2 were at ‘0000000100000000’ and the heuristic 1 was at ‘0100101110011011’.

JP-EN	NIST	BLEU	METEOR	EN-FR	NIST	BLEU	METEOR
system t1	7.0374	0.2532	0.6083487	system t1	5.6683	0.1652	0.5134530
system t2	7.2992	0.2775	0.6223682	system t2	6.3356	0.2235	0.5765081
system t3	5.1474	0.1243	0.4527874	system t3	5.2992	0.1402	0.4622777
system t4	6.6323	0.1913	0.5590906	system t4	6.0325	0.1945	0.5499950
system t5	6.6682	0.2165	0.5827379	system t5	6.3880	0.2217	0.5579302
system t6	6.8597	0.2428	0.5909936	system t6	5.6773	0.1664	0.5152482
system t7	7.2555	0.2755	0.6193990	system t7	6.2267	0.2170	0.5575926
system t8	6.1250	0.1946	0.6090198	system t8	6.4064	0.2262	0.5614477
system t9	7.2182	0.2529	0.6062563	system t9	6.2788	0.2148	0.5525901
system t10	5.6288	0.1727	0.5141809	system t10	6.0535	0.2034	0.5516885
system t11	7.2625	0.2529	0.6105696	system t11	5.5635	0.1624	0.5137018
				system t12	6.3131	0.2201	0.5574140
				system t13	6.1832	0.2112	0.5514069
				system t14	6.1462	0.2055	0.5582915
				system t15	6.2394	0.2059	0.5303054
				system t16	6.2529	0.2161	0.5567934
baseline	7.2992	0.2775	0.6223682	baseline	6.4064	0.2262	0.5614477
heuristic 1	7.4292	0.2750	0.6228906	heuristic 1	5.5584	0.1799	0.5820681
heuristic 2	7.2992	0.2775	0.6223682	heuristic 2	6.4064	0.2262	0.5614477
<b>our algorithm</b>	<b>7.5161</b>	<b>0.2869</b>	<b>0.6305818</b>	<b>our algorithm</b>	<b>6.5033</b>	<b>0.2292</b>	<b>0.5792332</b>

BLEU was 0.30 points absolute and 1.3% relative compared to the single best performing system. Heuristic 2 and baseline shows the result of system t8. Note that the number of items in the power set (corresponding to the set of all possible sets of MT systems participating in the combination) in ES-EN was 31, JP-EN was 4094, and EN-FR was 65534.

## 6 Conclusion and Further Studies

This paper investigates the enlarged hypothesis space in MBR decoding in SMT, employing MAP inference on clique tree. This mechanism can substitute the calibration of probabilities with the mechanism of max-product algorithm. First of all, MBR decoding has not been much investigated compared to MAP decoding in SMT, but is rather regarded as a practical tool which achieves state-of-the-art performance for evaluation campaigns. Traditionally, the full set of MT outputs or only to some MT outputs as selected by human beings are employed for MBR decoding. There has been no paper yet to describe the optimization process of this as far as we know (Hence, the search space for the best combination shown



**Fig. 3.** The left figure shows the count of exact matches among the translation outputs of Moses as a 100-best list after stop-word removal and sorting; We project each sentence in a 100-best list onto a vector space model and count the number of points. The middle figure shows the same quantity for a 1000-best list. The right figure shows the same quantity for a 7-multiple reference (human translation). We use the parallel data of IWSLT 07 JP-EN where we use devset5 (500 sentence pairs) as a development set and devset4 (489 sentence pairs) as a test set; 7-multiple references consist of devset4 and devset5 (989 sentence pairs). For example, the left figure shows that 7% of sentences produce only one really useful translation in a 100-best list and the other 99 sentences in the 100-best list are just reordered versions. In contrast, the right figure of human translation shows that more than 70% of sentences in 7 multiple references are meaningfully different.

in Figure 2 is rarely seen.) Secondly, our algorithm can be successfully applied to the case where the number of participating systems is more than 10, which is the case for the second and the third experiments. Between ES-EN, the improvement was 2.16 BLEU points absolute and 9.2% relative compared to the best single system. Between JP-EN, the improvement was 0.94 points absolute and 3.4% relative. Between FR-EN, the improvement was 0.30 points absolute and 1.3% relative.

There are several avenues for further study. Firstly, to date our experiments involved at most 16 systems. We would like to enlarge the size of the input such as the 1000-best list as in Tromble et al. [28] and DeNero et al. [4], and a general MT translation setting as in Arun et al. [1]. Their improvements are in general quite small compared to the confusion network-based approach. As is shown in Figure 3, the 100-best list and the 1000-best list produced by Moses [11] tend not to be sufficiently different and do not produce useful translation alternatives. As a result, their BLEU score tends to be low compared to the (nearly best) single systems. This means that in our strategy those MT inputs may be better removed rather than employed as a useful source in system combination.

Yet another avenue for further study is to provide prior knowledge into the system combination module. In [19,18,21], we showed that word alignment may include successfully prior knowledge about alignment links. It would be interesting to incorporate some prior knowledge about system combination, for example, (in)correct words or phrases in some particular translation output.

**Acknowledgments.** We thank Jinhua Du. This research is supported by the the 7th Framework Programme and the ICT Policy Support Programme of the European Commission through the T4ME project (Grant agreement No. 249119).

## References

1. Arun, A., Haddow, B., Koehn, P.: A unified approach to minimum risk training and decoding. In: Proceedings of Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp. 365–374 (2010)
2. Bangalore, S., Bordel, G., Riccardi, G.: Computing consensus translation from multiple machine translation systems. In: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 350–354 (2001)
3. Callison-Burch, C., Koehn, P., Monz, C., Schroeder, J.: Findings of the 2009 workshop on statistical machine translation. In: Proceedings of EACL Workshop on Statistical Machine Translation 2009, pp. 1–28 (2009)
4. DeNero, J., Chiang, D., Knight, K.: Fast consensus decoding over translation forests. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 567–575 (2009)
5. DeNero, J., Kumar, S., Chelba, C., Och, F.: Model combination for machine translation. In: Proceedings of NAACL, pp. 975–983 (2010)
6. Du, J., He, Y., Penkale, S., Way, A.: MaTrEx: the DCU MT System for WMT 2009. In: Proceedings of the Third EACL Workshop on Statistical Machine Translation, pp. 95–99 (2009)
7. Federmann, C.: Ml4hmt workshop challenge at mt summit xiii. In: Proceedings of ML4HMT Workshop, pp. 110–117 (2011)
8. Fujii, A., Utiyama, M., Yamamoto, M., Utsuro, T., Ehara, T., Echizen-ya, H., Shimohata, S.: Overview of the patent translation task at the NTCIR-8 workshop. In: Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access, pp. 293–302 (2010)
9. Goel, V., Byrne, W.: Task dependent loss functions in speech recognition: A-star search over recognition lattices. In: Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH), pp. 51–80 (1999)
10. Koehn, P.: Statistical machine translation. Cambridge University Press (2010)
11. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180 (2007)
12. Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT / NAACL 2003), pp. 115–124 (2003)
13. Koller, D., Friedman, N.: Probabilistic graphical models: Principles and techniques. MIT Press (2009)

14. Kumar, S., Byrne, W.: Minimum Bayes-Risk word alignment of bilingual texts. In: Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 140–147 (2002)
15. Leusch, G., Matusov, E., Ney, H.: The rwth system combination system for wmt 2009. In: Fourth EACL Workshop on Statistical Machine Translation (WMT 2009), pp. 56–60 (2009)
16. Matusov, E., Ueffing, N., Ney, H.: Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In: Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 33–40 (2006)
17. Och, F.: Minimum Error Rate Training in Statistical Machine Translation. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 160–167 (2003)
18. Okita, T.: Word alignment and smoothing method in statistical machine translation: Noise, prior knowledge and overfitting. PhD thesis. Dublin City University, pp. 1–130 (2011)
19. Okita, T., Guerra, A.M., Graham, Y., Way, A.: Multi-Word Expression sensitive word alignment. In: Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA 2010, collocated with COLING 2010), Beijing, China, pp. 1–8 (2010)
20. Okita, T., Jiang, J., Haque, R., Al-Maghout, H., Du, J., Naskar, S.K., Way, A.: MaTrEx: the DCU MT System for NTCIR-8. In: Proceedings of the MII Test Collection for IR Systems-8 Meeting (NTCIR-8), Tokyo, pp. 377–383 (2010)
21. Okita, T., Way, A.: Given bilingual terminology in statistical machine translation: Mwe-sensitive word alignment and hierarchical pitman-yor process-based translation model smoothing. In: Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference (FLAIRS-24), pp. 269–274 (2011)
22. 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 of the Association for Computational Linguistics (ACL 2002), pp. 311–318 (2002)
23. Pearl, J.: Reverend bayes on inference engines: A distributed hierarchical approach. In: Proceedings of the Second National Conference on Artificial Intelligence (AAAI 1982), pp. 133–136 (1983)
24. Shenoy, P., Shafer, G.: Axioms for probability and belief-function propagation. In: Proceedings of the 6th Conference of Uncertainty in Artificial Intelligence (UAI), pp. 169–198 (1990)
25. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223–231 (2006)
26. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pp. 2142–2147 (2006)
27. Stolcke, A.: SRILM – An extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, pp. 901–904 (2002)
28. Tromble, R., Kumar, S., Och, F., Macherey, W.: Lattice minimum bayes-risk decoding for statistical machine translation. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 620–629 (2008)
29. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13, 260–269 (1967)

# Phrasal Syntactic Category Sequence Model for Phrase-Based MT

Hailong Cao<sup>1</sup>, Eiichiro Sumita<sup>2</sup>, Tiejun Zhao<sup>1</sup>, and Sheng Li<sup>1</sup>

<sup>1</sup> Harbin Institute of Technology, China

<sup>2</sup> National Institute of Information and Communications Technology, Japan

{hailong, tjzhao, shengli}@mtlab.hit.edu.cn,  
eiichiro.sumita@nict.go.jp

**Abstract.** Incorporating target syntax into phrase-based machine translation (PBMT) can generate syntactically well-formed translations. We propose a novel phrasal syntactic category sequence (PSCS) model which allows a PBMT decoder to prefer more grammatical translations. We parse all the sentences on the target side of the bilingual training corpus. In the standard phrase pair extraction procedure, we assign a syntactic category to each phrase pair and build a PSCS model from the parallel training data. Then, we log linearly incorporate the PSCS model into a standard PBMT system. Our method is very simple and yields a 0.7 BLEU point improvement when compared to the baseline PBMT system.

**Keywords:** machine translation, natural language processing, phrase-based machine translation.

## 1 Introduction

Both PBMT models (Koehn et al., 2003; Chiang, 2005) and syntax-based machine translation models (Yamada et al., 2000; Quirk et al., 2005; Galley et al., 2006; Liu et al., 2006; Marcu et al., 2006; and numerous others) are state-of-the-art statistical machine translation (SMT) methods. Over the last several years, an increasing amount of work has been done to combine the advantages of the two approaches. DeNeefe et al. (2007) made a quantitative comparison of the phrase pairs that each model has to work with and found it is useful to improve the phrasal coverage of their string-to-tree model. Liu et al. (2007) proposed forest-to-string rules to capture the non-syntactic phrases in their tree-to-string model. Zhang et al. (2008) proposed a tree sequence based tree-to-tree model which can describe non-syntactic phrases with syntactic structure information.

The converse of the above methods is to incorporate syntactic information into the PBMT model. Zollmann and Venugopal (2006) started with a complete set of phrases as extracted by traditional PBMT heuristics, and then annotated the target side of each phrasal entry with the label of the constituent node in the target-side parse tree that

subsumes the span. Hassan et al. (2007) and Birch et al. (2007) improved a PBMT system by incorporating syntax in the form of supertags. Marton and Resnik (2008) and Cherry (2008) imposed syntactic constraints by making use of prior linguistic knowledge in the form of syntax analysis. Xiong et al. (2009) proposed a syntax-driven bracketing model to predict whether a phrase (a sequence of contiguous words) is bracketable or not using rich syntactic constraints.

This paper focuses on incorporating syntactic information into a PBMT model. Our motivation is that PBMT is good at generating translations inherent in the phrase pairs but inefficient at grammatically reordering the target phrases. To deal with this problem, we propose a novel phrasal syntactic category sequence (PSCS) model which allows a PBMT decoder to prefer more grammatical target phrase sequences and better translations.

## 2 Target Phrase Annotation

In this section, we briefly review the phrase pair extraction algorithm and describe how to assign a syntactic category to each phrase pair.

The basic translation unit of a PBMT model is a phrase pair consisting of a sequence of source words, a sequence of target words and a vector of feature values which represents this pair’s contribution to the translation model. In typical PBMT systems such as MOSES (Koehn, 2007), phrase pairs are extracted from word-aligned parallel corpora. All pairs of “source word sequence ||| target word sequence” that are consistent with word alignments are collected. Prior to the phrase pair extraction, we use the Berkeley parser<sup>1</sup> (Petrov et al., 2006) to generate the most likely parse tree for each English target sentence in the training corpus.

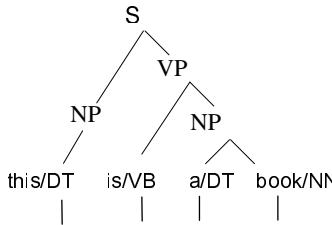
There are many ways to annotate a phrase pair using a parse tree. Here, we follow the method used in Zollmann and Venugopal (2006). In detail, if the target side of any of these phrase pairs corresponds to a syntactic category of the target side parse tree, we label the phrase pair with that syntactic category. Phrase pairs that do not correspond to a span in the parse tree are given a default category “X”. For each phrase pair, we also record the original position of its first and last target word in the target sentence. These position indices will be used in the next section.

For example, given a Chinese-English sentence pair, the English parse tree and the word alignments as shown in Figure 1, we can extract the phrase pairs and the syntactic categories shown in Table 1. Note that if there are any unary rules in the parse tree, we only keep the highest node. So “NP” over the word “this” is kept and “DT” is ignored.

We ran the above procedure on the entire parallel corpus. We may extract the same phrase pair from different parallel sentence pairs whose target side parsing trees are different. So a phrase pair may have multiple syntactic categories. We record all possible syntactic categories and their counts for each phrase pair.

---

<sup>1</sup> <http://code.google.com/p/berkeleyparser/>



**Fig. 1.** An example parse tree and word-based alignments

(Zollmann and Venugopal, 2006) used the syntactic category in a hierarchical PBMT model by treating category as non-terminal symbols. In the next section, we propose a novel model which uses the syntactic category in a conventional (i.e., non-hierarchical) PBMT system.

**Table 1.** Phrase pairs and the syntactic categories extracted from the example in Figure 1

Source phrase	Target phrase	Syntactic category	start	end
	this	NP	0	0
	is	VB	1	1
	a	DT	2	2
	book	NN	3	3
	this is	X	0	1
	is a	X	1	2
	a book	NP	2	3
	this is a	X	0	2
	is a book	VP	1	3
	this is a book	S	0	3

### 3 PSCS Model

In a PBMT system, the translation candidates are generated from left to right by using a sequence of phrase pairs. Together with the sequence of phrase pairs, a PSCS in the form of  $sc_1 sc_2, \dots, sc_n$  is also generated. The variable  $sc_i$  stands for the syntactic

category of the  $i$ -th phrase pair. For example, if we translate a sentence “ ” with the phrase pairs “ ||| this” and “ ||| is a book” then the corresponding PSCS is “NP VP”. By preferring more likely PSCS, one would expect that the output of the decoder will be more grammatical.

We use a bi-gram model to calculate the probability of a PSCS:

$$\begin{aligned} & P(sc_1 sc_2, \dots, sc_n) \\ &= P(sc_1 | < s >) \cdot P(sc_2 | sc_1) \cdot \dots \cdot \\ & P(sc_n | sc_{n-1}) \cdot P(< /s > | sc_n) \end{aligned} \quad (1)$$

The start mark  $< s >$  and the end mark  $< /s >$  are used to model how likely  $sc_1$  and  $sc_n$  is to occur at the beginning and the end position respectively. The probability is incorporated into the PBMT model log linearly as a new feature.

Our PSCS model is similar to the supertagged language model utilized in Hassan et al. (2007) and Birch et al. (2007). The difference is that they use word-level shallow syntax information in the form of supertags while we use phrase-level full parsing information in the form of syntactic category.

### 3.1 Dealing with Ambiguity

So far, in this section, we have assumed that each phrase pair used by the decoder has only one syntactic category. However, as we mentioned in section 2, there may be multiple syntactic categories corresponding to one phrase pair.

In order to deal with this ambiguity, one method is to consider each possible syntactic category separately in the decoder. Another is to consider the syntactic category as a hidden variable. In this paper, we use the latter approach simply because it is easy to implement. If the  $i$ -th phrase pair  $pp_i$  has  $m$  possible syntactic categories  $sc_{i,1}, sc_{i,2}, \dots, sc_{i,m}$ , and the  $(i-1)$ -th phrase pair  $pp_{i-1}$  has  $n$  possible syntactic categories  $sc_{i-1,1}, sc_{i-1,2}, \dots, sc_{i-1,n}$ , then we intuitively replace the log probability  $\log(P(sc_i | sc_{i-1}))$  in Equation (1) with a linear combination score:

$$\sum_{p=1}^m \sum_{q=1}^n P(sc_{i,p} | pp_i) \cdot P(sc_{i-1,q} | pp_{i-1}) \cdot \log(P(sc_{i,p} | sc_{i-1,q}))$$

This score is a weighted sum of all possible PSCS bi-gram log probabilities for two contiguous phrase pairs. The weight is empirically set as:

$$P(sc_{i,p} | pp_i) \cdot P(sc_{i-1,q} | pp_{i-1})$$

### 3.2 Training of the PSCS Model

Now we describe how to estimate the parameters of the PSCS model. The syntactic category is of phrase-level information, but there is no explicit phrase segmentation in the parallel corpus. This means that we do not have the syntactic category sequence

data that can be directly used to train our PSCS model. Following the phrase extraction method (Koehn, 2007), a heuristic method is used to solve this problem.

We begin with a set of phrase pairs extracted from each sentence pair in the parallel corpus. Each phrase pair is assigned a syntactic category by the method described in section 2. Then we look at the set of phrase pairs, and if any two of them are contiguous in the target side, we extract the syntactic category of these two phrase pairs as a bigram. Figure 2 shows the details of our algorithm. Then we split each bigram to get uni-gram samples, which are used to perform data smoothing. Given the collected uni-gram and bi-gram syntactic category samples and their counts, we use the SRI language modeling toolkit<sup>2</sup> to build a bigram PSCS model. The model is smoothed by Witten-Bell discounting.

```

Input:
  Source sentence  $s$ 
  Target sentences  $t$ 
  Source phrase  $sp$ 
  Target phrase  $tp$ 
Output: PSCS bigram
bigram = empty
For each  $(t, s)$  in the parallel corpus
  For each  $(sp_i, tp_i)$  extracted from  $(t, s)$ 
    For each  $(sp_j, tp_j)$  extracted from  $(t, s)$ 
      If  $(tp_i.end + 1 == tp_j.start)$ 
        bigram.add( $tp_i.sc, tp_j.sc$ )
      End
    If( $tp_i.start == 0$ )
      bigram.add( $< s >, tp_i.sc$ )
    If( $tp_i.end == t.length - 1$ )
      bigram.add( $tp_i.sc, < /s >$ )
    End
  End

```

**Fig. 2.** Training algorithm for PSCS model

## 4 Experiments

Our SMT system is based on a fairly typical phrase-based model (Finch and Sumita, 2008). We use a modified training toolkit adapted from the MOSES decoder to train our SMT model. Our decoder can operate on the same principles as the MOSES decoder. The decoder is modified to accommodate our PSCS model. Minimum error rate training (MERT) with respect to BLEU score is used to tune the decoder's parameters, and it is performed using the standard technique of Och (2003). Lexical reordering model is used in our experiments.

<sup>2</sup> <http://www-speech.sri.com/projects/srilm/manpages/ngram-count.1.html>

**Table 2.** Corpora statistics

Data	Sentences	Chinese words	English words
Training set	243,698	7,933,133	10,343,140
Development set	1664	38,779	46,387
Test set	1357	32377	42,444
GIGAWORD	19,049,757	-	306,221,306

The translation model was created from the FBIS corpus. We use a 5-gram language model trained with modified Knesser-Ney smoothing. The language model is trained on the target side of the FBIS corpus and the Xinhua news from the GIGAWORD corpus. The development and test sets are from the NIST MT08 evaluation campaign. Table 2 shows the statistics of the corpora used in our experiments.

#### 4.1 Experiments on PSCS Model

As we mentioned in section 2, we use the Berkeley parser to parse the target side of the parallel corpus. Each phrase pair is annotated with the method introduced in section 2. For sentences in which the parser fails to generate a parse tree, we use the default syntactic category X to annotate the phrase pairs. We extracted 21,862,759 phrase pairs in total. There are 14,890,317 phrase pairs whose target side syntactic category is X. The other 6,972,442, or 31%, of the phrase pairs are annotated with linguistic syntactic categories.

Then we build a PSCS model based on the method proposed in section 3. There were 72 kinds of uni-gram syntactic categories and 2478 kinds of bi-gram syntactic categories.

#### 4.2 Experiments on Chinese-English SMT

In order to confirm the effect of our PSCS model, we performed two translation experiments. The first one was a baseline PBMT experiment. In the second experiment, we incorporated our PSCS model into the PBMT system. The evaluation metric is case-sensitive BLEU-4. The results are given in Table 3.

**Table 3.** Comparison of translation quality

System	BLEU Score
PBMT	17.26
PBMT+PSCS	17.92

We were able to achieve an improvement of about 0.7 BLEU point over the baseline PBMT system. This improvement indicates that syntactic categories, even though only 31% of them maintain linguistic meanings, can help select better translation candidates.

## 5 Conclusion and Future Work

We propose a novel PSCS model to incorporate syntactic information into the conventional PBMT. The PSCS model allows a PBMT decoder to prefer more grammatical target phrase sequences and better translations. Our method is very simple and yields a 0.7 BLEU point improvement when compared to the baseline PBMT system.

We plan to annotate phrase pairs with additional richer syntactic information to obtain further improvements in future work.

**Acknowledgment.** The work of HIT in this paper is funded by the project of National Natural Science Foundation of China (No. 61173073).

## References

- Birch, A., Osborne, M., Koehn, P.: CCG Supertags in Factored Translation Models. In: SMT Workshop. ACL (2007)
- Cherry, C.: Cohesive phrase-Based decoding for statistical machine translation. In: ACL- HLT (2008)
- Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: ACL (2005)
- DeNeefe, S., Knight, K., Wang, W., Marcu, D.: What can syntax-based MT learn from phrase-based MT? In: EMNLP-CoNLL (2007)
- Finch, A., Sumita, E.: Dynamic model interpolation for statistical machine translation. In: SMT Workshop (2008)
- Galley, M., Graehl, J., Knight, K., Marcu, D., Deneefe, S., Wang, W., Thayer, I.: Scalable inference and training of context-rich syntactic translation models. In: ACL (2006)
- Hassan, H., Sima'an, K., Way, A.: Supertagged phrase-based statistical machine translation. In: ACL (2007)
- Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: HLT-NAACL (2003)
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: ACL demo and poster sessions (2007)
- Liu, Y., Liu, Q., Lin, S.: Tree-to-string alignment template for statistical machine translation. In: ACL-COLING (2006)
- Liu, Y., Huang, Y., Liu, Q., Lin, S.: Forest-to-string statistical translation rules. In: ACL (2007)
- Marcu, D., Wang, W., Echihabi, A., Knight, K.: SPMT: Statistical machine translation with syntactified target language phrases. In: EMNLP (2006)

- Marton, Y., Resnik, P.: Soft syntactic constraints for hierarchical phrasal-based translation. In: ACL-HLT (2008)
- Och, F.: Minimum error rate training in statistical machine translation. In: ACL (2003)
- Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: COLING-ACL (2006)
- Quirk, C., Menezes, A., Cherry, C.: Dependency treelet translation: Syntactically informed phrasal SMT. In: ACL (2005)
- Xiong, D., Zhang, M., Aw, A., Li, H.: A syntax-driven bracketing model for phrase-based translation. In: ACL-IJCNLP (2009)
- Yamada, K., Knight, K.: A syntax-based statistical translation model. In: ACL (2000)
- Zhang, M., Jiang, H., Aw, A., Tan, C.L., Li, S.: A tree sequence alignment-based tree-to-tree translation model. In: ACL- HLT (2008)
- Zollmann, A., Venugopal, A.: Syntax augmented machine translation via chart parsing. In: SMT Workshop, HLT-NAACL (2006)

# Integration of a Noun Compound Translator Tool with Moses for English-Hindi Machine Translation and Evaluation

Prashant Mathur and Soma Paul

Language Technology Research Center  
International Institute of Information Technology  
Hyderabad (A.P)  
`mathur@research.iiit.ac.in, soma@iiit.ac.in`

**Abstract.** Noun Compounds are a frequently occurring multiword expression in English written texts. English noun compounds are translated into varied syntactic constructs in Hindi. The performance of existing translation system makes the point clear that there exists no satisfactorily efficient Noun Compound translation tool from English to Hindi although the need of one is unprecedented in the context of machine translation. In this paper we integrate Noun Compound Translator [13], a statistical tool for Noun Compound translation, with the state-of-the-art machine translation tool, Moses [10]. We evaluate the integrated system on test data of 300 source language sentences which contain Noun Compounds and are translated manually into Hindi. A gain of 29% on BLEU score and 27% on Human evaluation has been observed on the test data.

## 1 Introduction

Noun Compounds (henceforth NC) are frequently occurring multi word expression in English written texts<sup>1</sup>. There involves the following issues which make automatic translation of English NCs a challenging NLP tasks:

1. NCs in English are too varied to be pre-compiled in an exhaustive list of translated NN compounds. Any translation system must therefore be able to handle novel NN compounds, such as computer keyboard, blog posts, home loan, sitting-room window, on the fly.
2. English NCs can be translated into various syntactic constructs in Hindi making selection of right construct type in the target language for a given source language NC a hard task.

In this paper we describe an integrated MT system which is developed by combining two systems, a Noun Compound Translator (henceforth NCT) developed

<sup>1</sup> [15] stated that BNC Corpus (84M words) [6] has 2.6% of NCs and Reuters has 3.9% of bigram NCs.

by [13] and [10]. NCT is a tool that uses sentential context to translate an English Noun Compound into Hindi. In order to examine the usefulness of the tool in the context of a full-fledged translation system, we have made an effort to integrate NCT with one state-of-the art translation system. We aim to verify whether the tool once integrated with a fully developed MT system results in improvement of the performance of the current MT system. In order to carry out the task, we have selected Moses, a SMT toolkit that performs statistical machine translation in the following steps:

1. It automatically trains translation and reordering models from a given pair of parallel texts and
2. Moses decoder decodes the source sentence (containing the Noun Compounds in this case) into target sentence using the translation/reordering models and the language model (built on the target language corpus).

There are two types of translation models that are generated by Moses

1. Phrase Based Model
2. Tree based Model

Since NCT is also a phrase based system, integrating it with another phrase based system makes the integration easier than integrating it with a Syntax based SMT or any other SMT systems such as Example-based MT, Tree based MT. We attempt to present the following tasks in this paper:

1. We build an enhanced model by combining Moses phrase based model and NCT system. Moses decoder uses the enhanced model and language model to generate sentential translation.
2. We evaluate the translation system extensively both automatically and manually. We compare the output of integrated “Moses + NCT” system with Moses standalone.

We report that the integrated system has achieved a 29% improvement when evaluated with BLEU metric and a 27% improvement on human judgment over Moses standalone system. The paper has been divided into a number of sections. The next section presents the issues related to translation of English noun compound in Hindi. In section 3, we briefly review previous works done on statistical machine translation (SMT), automatic noun compound translation, and MT evaluation. We describe the integration of NCT with Moses in section 4. Section 5 presents details of data preparation for evaluation. Finally the evaluation report is presented in section 6.

## 2 Noun Compound and Its Translation

Noun Compounds occur frequently in English. In this work we concentrate only on bigram Noun Compounds where the rightmost Noun is the head(H) and the preceding Noun is the modifier(M) as found in finance minister, body weight,

**Table 1.** Distribution of translations of English NC from English-Hindi parallel corpora

Construction Type	Example	No. of Occurrence
Nominal Compound	birth rate <i>janma dara</i>	3959
Genitive(of-kA/ke/kl <sup>3</sup> )	Geneva Convention <i>jenIvA kA samajhOtA</i> Geneva of convention	1976
Purpose (for-ke liye )	Research Center <i>Sodha ke lie kendra</i> Research for Center	22
Location (at/on-par )	wax work <i>mom par citrom</i> wax on work	34
Location (in-meM <sup>4</sup> )	gum infection <i>masUDe meM roga</i> gum in infection	93
Adjective Noun Phrase	Hill Camel <i>pahARI UMTa</i> hilly camel	557
Single Word	cow dung <i>gobar</i>	766
Transliterated NCs	poultry bird <i>pOltrI barda</i>	1208

machine translation and so on. The translation becomes significantly difficult in those case when the source language NC is represented in a varied manner in the target language as is the case with English - Hindi language pair. We have done a manual study on the BNC corpus<sup>2</sup> in which we have found that English noun compounds can be translated in Hindi in following varied ways

The first column in the Table 1 indicates that English NCs can be translated into various construct types in Hindi such as NC → NC, NC → 'M PostP H', NC → 'Adj N', NC → 'Single word'. We have also come across some cases where an NC corresponds to a paraphrase construct for which we have not given a count in this table. There are .08% cases (see Table 1) when an English NC becomes a single word form in Hindi. The single word form can either be a simple word as in ('cattle dung' *gobar*) or a compounded word such as 'blood pressure' *raktacApa*, 'transition plan' *parivartana-yojanA*. There are 1208 cases (approximately 13%) where the English nominal compound is not translated but transliterated in Hindi. They are mostly technical terms, names of chemicals and so on. Furthermore, compounding is an extremely productive process in English. As stated in [1] the frequency spectrum of compound types follows a Zipfian or power-law distribution, so in effect many compound tokens encountered belong to a "long tail" of low-frequency types. Over half of the two-noun compound types in the British National Corpus occur just once. Taken together, the factors of

<sup>2</sup> The corpus size is of around 50,000 sentences in which we got 9246 sentences (i.e. 21% cases of the whole corpus) that has nominal compound.

low frequency and high productivity mean that noun-compound translation is an important goal for broad-coverage translation system. Also Hindi is a free order word language, it is more difficult to get a good BLEU score with a single reference gold translation. The human translators while building the gold standard data generally try to preserve the meaning of the whole sentence, and so they are not bound to retain the syntactic structure of the NC in target side which leads to arbitrary restructuring of the sentence.

### 3 Related Work

We present review on the following a) SMT systems b) Noun Compound Translation System c) Evaluation process of MT system

#### 3.1 SMT System

IBM introduced SMT in the early 1990s with their original approach of word-to-word translation allowing insertion and deletion of words. Phrase-based MT was originally introduced by [8]. Most of the best performing Machine Translation systems uses phrase based models including Google. [7] has introduced a tool, IRSTLM to build the language model. Language model keeps the grammar of the translated sentence in check. Moses [10] has introduced a state-of-the-art machine translation tool Moses which allows automatic training of the translation model. Moses treats a Noun Compound as just another phrase and does not perform any special operation for the translation of NC. Both of the issues - Right lexical selection and Right construct selection for the target language as discussed by Rackow et. al are not handled explicitly by Moses. Above all low-frequency of NCs means less training data (on source-target sides) and thus lower/insignificant probability scores of translation. These factors motivate us to propose for an integrated system - a statistical system with some linguistic information which performs and tackle the issue with Noun Compound specifically.

#### 3.2 Noun Compound Translation

Translation of Noun Compounds is a widely explored area and a lot of work has been done on it before. We have come across several works in this area a) Rackow et. al has implemented a transfer based approach b) Baldwin & Tanaka has worked on MT of Noun Compounds for Japanese-English language pair c) Bungum and Oepen has discussed the translation of Norwegian Noun Compounds d) Gawronska et. al. has done work on interpretation of Noun Compounds for machine translation from English to Slavic languages. While a) & d) has worked on transfer and rule based translation of Noun Compounds b ) & c) has worked on translation of Noun Compounds using statistics. Even though Mathur and Paul has closely followed the approach proposed by [15] and [5]; this work has shown that the correct sense selection of the component nouns

of a given nominal compound during the analysis stage significantly improves the performance of the system and makes the present work distinct from all the previous works done for automatic bilingual translation of Nominal compounds. Mathur and Paul has developed the Noun Compound translation tool (NCT) based on the two stage of translation as proposed in Baldwin & Tanaka and Bungum and Oopen. The stages are:

1. Generation : Setting the target lexemes in translation templates to generate translation candidates
2. Selection : Selecting the best translation out of translation candidates.

However, [13] significantly differs from [15] and [5] in the following manner: While working on source language side, both [15] and [5] disregard local contexts and do not attempt to identify the sense of nominal compound in the given context. They take into account of all possible translations of the component nouns while performing the corpus search. In this way the number of search candidates has become many. On the other hand, [13], while translating a nominal compound, has considered the meaning of that compound in the given context, that is, the sentence in which it has occurred. Thus the translation is carried out in two steps:

1. Context information is utilized for correct lexical substitution of components nouns of English NC.
2. Hindi templates for potential translation candidates are generated which are searched in the target language data.

### 3.3 MT Evaluation

One of the most difficult problems of Machine Translation is the evaluation of a proposed system. If one system is better than the other it can only be proven if there is a score attached to it, the better the score the better the system. There has been work done on evaluation of systems both automatically and by human evaluation. ALPAC was formed in 1964 to evaluate the progress of machine translation by using human translators. The translators studied two measures “intelligibility” and “fidelity”. Intelligibility measured how good is the language of the sentence and fidelity ensured that all the information is translated in the target sentence. In this work we have also used two measures Adequacy and Fluency which represents fidelity and intelligibility respectively and they are measured on a scale of 5 (as described in Section 6 ) unlike on a scale of 10 done in the previous one. Many automatic evaluation metrics ( a metric which represents the quality of the translation) have been developed in the later like WER, TER, BLEU, NIST. In 2002, Papineni et. al. introduces the metric BLEU (Bilingual Evaluation Understudy) for automatic evaluation of machine translation and it was one of the first metrics to have a high correlation with the

human judgements and is a benchmark for new evaluation metrics. Therefore, we have used BLEU for evaluation of our systems.

## 4 Integration

As mentioned earlier, both Moses and NCT are phrase based systems and therefore their integration is not difficult. This section presents the integration in detail. We have applied two different techniques for integration. They are integrating by a) Generating additional Phrase Table and b) Generating additional Training Data. For both methods, following common steps have been followed:

1. For each compound in the sentence we generate the translations using Noun Compound Translator and make a list of all the translation pairs.
2. We train the translation and the reordering models using Moses on the training data described in Section 5.

### 4.1 Generating Additional Phrase Table

After training is done, we used the translation pairs to build the phrase table of the NCs and add that phrase table to the list of phrase table already generated by Moses<sup>5</sup>. Since we want the decoder to choose the translation options provided by the NC phrase/reordering table, we raise the probabilities of the features (explained in details in [13]) to the maximum(i.e 1). Thus, the phrase table generated for the NC ‘election campaign’ would be the following:

election campaign ||| *cUnAva aBiyAna* ||| (0) (1) ||| (0) (1) ||| 1 1 1 1 2.718

The results with this method are not satisfactory (refer [13] for more details). We have done the error analysis for this system. We have found out that the decoder is complied to choose the translation option provided by NCT which in turn affects the translation of the whole sentence. We have, therefore, resorted to another method for integrating the two systems together.

### 4.2 Generating Additional Training Data

We use the translation pairs and treat them as a parallel corpora. Rather than building the whole phrase table we just use the translation pairs as the parallel text. For these translation pairs to be selected in decoding process they are required to outweigh the other possible translations. To ensure that, we build a parallel corpus which contains each translation pair 10 times in order to outweigh other translations of a given pair generated from the training corpus by Moses. In this case computational cost is much cheaper than the first method as we didnt have to align translation pairs. MERT is performed afterwards for the optimization of translation quality.

---

<sup>5</sup> Moses provides a feature of adding a phrase table with flat weights to the existing ones.

## 5 Data Preparation

Statistical machine translation uses three different data namely a) Training Data b) Development Data c) Test Data. We have used Tourism Corpus<sup>6</sup> parallel data which has three segments: training, development and test. We have used this corpus for building our training and development data. For training of translation and reorder model we have used the training data of 8169 sentences. For minimum error rate training [9] we have used a development set of 361 sentences. Gyannidhi Corpus of 12000 Hindi sentences has been used to build a trigram language model with Kneser-Ney [4] smoothing using IRSTLM [7] tool. The size of corpus for training and development data is given in Table 2:

**Table 2.** Corpus Statistics

Corpus	Sentences	Source Words	Target Words
Training	8169	0.17M	0.18M
Development	361	7741	7992
Gyannidhi	12K	N.A	0.4M

The test data has not been taken from the Tourism corpus because of two reasons: a) many NCs in the Tourism Corpus are technical terms and the translation of technical terms is often transliterated form of the input, hence, not fit for our translation purpose; b) the number of NCs in tourism test data are also insignificant. These factors motivate us to build our own test data from a general domain. To generate the test data we have used the source side of a English-Hindi parallel corpora of 50K sentences. We have used Tree-Tagger [3] to tag the source side (it gives an additional useful lemma information with the POS tag). Out of the 50K tagged sentences we have extracted about 15K sentences which contains bigram Noun Compound. Finally 300 NCs from this dataset have been handpicked to build the gold standard data which is in the following format:

1. Source Noun Compound
2. Target Correspondent of Source Noun Compound
3. Source Sentence
4. Target Sentence

This information is stored in a form of a tuple  $\langle NC-S, NC-T, SS, TS \rangle$ , where NC-S is the Noun Compound on the source side, NC-T is the translated correspond of Noun Compound on the target side, SS is the source sentence, TS is the corresponding target sentence.

---

<sup>6</sup> Hindi is a resource-poor language, the most proficient data we could manage was Tourism Corpus. This corpus have been used in English to Hindi MT task in NLP Tools Contest organized by IIIT Hyderabad.

## 6 Evaluation

The two systems described in the previous section are evaluated in this section on a test data of 300 sentences (see section 5 for preparation of data). Two other systems: Moses standalone and Google translator are also run on the same test data in order to compare the result. Two methods of evaluation are applied: a) Automatic evaluation by using BLEU metric [12] and b) human evaluation. Three human evaluators have evaluated the data and the details are given below. We will present the evaluation report of both sentence translation and NC translation. First we present in Table 3 the evaluation report of NC translation Tool, NCT and compare its performance with Moses and Google translator for NC translation alone:

**Table 3.** NC Translation Accuracy (Surface Level ) on test data

System	NC	Training Set
Moses (Baseline)	23%	180K words
Google	35%	-NA-
System 1	30%	180K words
System 2	28%	180K words

As shown in Table 3, system 1 performs much better than Moses (7% absolute improvement) and slightly better than System 2. It is because of the fact that System 1 has to use the translations of Noun Compounds given by NCT whereas System 2 is under no such compliance. System 1 is compelled to use the translation from the phrase table which in turn affects the performance of the overall sentential translation as we will see below during sentential translation evaluation. We also examine whether overall translation quality improves when our system is plugged in to an existing system. The following Table 4 shows the results of automatic evaluation of sentence translation. All these sentences contain nominal compound:

**Table 4.** BLEU scores on the test data

System	BLEU	Training Set
Moses (Baseline)	2.34	180K words
Google	8.07	-NA-
System 1	2.74	180K words
System 2	3.01	180K words

In Table 4 we observe that Google shows a pretty high BLEU score compared to other systems because of the vast amount of data they have which helps them build better translation and language models. However the score is still much low compared to other language pair such as English-French( 30 BLEU), Arabic-English( 35 BLEU). There exist other English-Hindi SMT systems but all

them have low BLEU scores for English-Hindi Translation task which is mainly because of the less and low quality training data. There is a relative improvement of 29% in System 2 w.r.t Moses system. The BLEU scores as reported in Table 4 is significantly lower than the scores we have obtained with the development set during the tuning phase as presented in Table 5.

**Table 5.** BLEU scores on the development set

System	BLEU Development Set	
Moses (Baseline)	10.49	10K words
System 1	10.55	10K words
System 2	10.70	10K words

The reason for this difference in BLEU score (compare Table 4 and 5) is because of the fact that test data could not be taken from the same domain from which training and development data has been selected as discussed in Section 5. To calculate Noun Compounds translation accuracy, the score is determined on the basis of an exact match with the gold data NCs. For example, the gold data translation for the NC sea food is given as *samudrI bhojana*. The translator tool returns the translation *samudrI khAdya*. Although food can be translated as khadya in this context, the score will give 0 because the translation output is not an exact match to the gold translation data. BLEU has the tendency of giving two semantically equivalent sentences a low score too. Moreover, Hindi is a free order word language, it is more difficult to get good BLEU score with a single reference gold translation which reflects our dataset. The human translators while building the gold standard data generally try to preserve the meaning of the whole sentence, and they are not, therefore, bound to retain the syntactic structure of the NC in target side which leads to arbitrary restructuring of the sentence. These observations have made us think that BLEU metric is not the perfect measure for evaluation in this case. We therefore employ three human evaluators (all of them are native language speakers) to evaluate the translations for all the systems and score them.

Each entry in the evaluation set contain

1. Noun Compound (Source)
2. Noun Compound (Target)
3. Source Sentence
4. Target Sentence (by all 4 systems.)

Noun Compounds are marked as correct translation if the translation by a system is semantically equivalent to the compound in gold data set. As an example, if gold standard data consists translation of sea food as *samudrI bhojana* and the system output is *samudrI khAdya* we mark the translation correct because they convey same meaning. To make evaluation an easy task NCs were scored

1. if translation Moses is correct
2. if translation by Google is correct
3. if translation by System-1 was correct
4. if translation by System-2 was correct
5. if none of the translation is correct

The human translators are asked to score the target sentence on the following scale as illustrated in Table 6.

**Table 6.** 5 point scale for Evaluation

Score	Adequacy	Fluency
5	contains all information	Flawless Grammar
4	contains most information	Good Grammar
3	contains much information	Non-native Grammar
2	contains little information	Disfluent Grammar
1	contains no information	Nonsense

Inter-annotator agreement for the sentential translations is 48% (144 entries out of 300); while for NC evaluation the agreement is 83.3% (250 entries out of 300). The following Table 7 presents human evaluation report for both sentential translation and NC translation:

**Table 7.** Human Judgment score of translation of sentences and the NC Translation accuracy

System	Sentential Translation	NC	Training Set
Moses (Baseline)	24.4%	48%	180K words
Google	40%	57%	-NA-
System 1	29.5%	64%	180K words
System 2	31%	60%	180K words

We can observe that Human Judgement scores are much higher and significant than the BLEU scores. There is a relative improvement of 27% in the performance of System 2 w.r.t Moses stand alone for sentential translations. The Noun Compound Translation accuracy is higher than the accuracy shown in Table 4.2. It is only because of the fact that in this experiment we have looked at the semantic equivalence (we have checked if the translated compound fit in the target sentence and it has conveyed the correct meaning) rather than surface-level matching. System 1 performs best in NC Translation accuracy (16% absolute improvement). An interesting point to note here is that System 2s Noun Compound Translation accuracy is higher than the System 1s but at the sentence level it is the other way round. The only probable reason for this change is that in System 1 while building a phrase table and complying the decoder to choose the translation from it, affects the translation score of the whole sentence and thus lower the scores.

## 7 Conclusion

In this paper, we have observed that an integrated system performs better than Moses stand alone on sentences containing NCs. Also, there is a significant difference in the Noun Compound Translation accuracy which evidently shows that NCT performs better than Moses and Google. This also indicates that if we have some linguistic information about the type of sentence we are translating (in this case a sentence with a NC) we can get better translations. The NCT system for which the noun compound translation accuracy is reported in this paper uses two tools (a) WordNet Sense Disambiguation tool [12] gives 72% accuracy in selecting the right sense of the constituents of Noun Compounds and (b) POS-Tagger [3] performs with an accuracy of 95%. Performance of the NCT system will be improved with improved performance in the pre-processing tools. This paper argues that BLEU metric is not suitable for the type of data we are evaluating. BLEU scores we have obtained are quite low and insignificant. As a result we have proposed human evaluation technique which have shown promising and significant results unlike the BLEU score.

## References

- [Seaghdha 2008] Seaghdha, D.O.: Learning Noun Compounds semantics. PhD Thesis, Computer Laboratory, University of Cambridge. Technical Report 735 (2008)
- [Gawronska et. al. 1994] Gawronska, B., Nordner, A., Johansson, C., Willners, C.: Interpreting compounds for machine translation. In: Proceedings of COLING 1994, Kyoto, Japan (1994)
- [Schmid 1994] Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: International Conference on New Methods in Language Processing. Manchester, UK (1994)
- [Kneser & Ney 1995] Reinhard, K., Ney, H.: Improved backing-off for n-gram language model. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Detroit, MI, vol. 1, pp. 181–182 (1995)
- [Bungum & Oepen 2009] Bungum, L., Oepen, S.: Automatic Translation of Norwegian Noun Compounds. In: Proceedings of the 13th Annual Meeting of the European Association for Machine Translation, EAMT 2009 (2009)
- [Lou Burnard 2000] Burnard, L.: User Reference Guide for the British National Corpus, Technical Report. Oxford University Computing Services (2000)
- [Federico et al. 2008] Federico, M., Bertoldi, N., Cettolo, M.: IRSTLM: an open source toolkit for handling large scale language models. In: INTERSPEECH 2008, pp. 1618–1621 (2008)
- [Och 2002] Och, F.J.: Statistical Machine Translation: From Single Words Models to Alignment Templates. PhD Thesis, RWTH Aachen, Germany (2002)
- [Och 2003] Och, F.: Minimum Error rate training in statistical machine translation. In: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), Sapporo, Japan, pp. 160–167 (2003)
- [Koehn et al. 2007] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: ACL 2007, Demonstration Session, Prague, Czech Republic (2007)

11. [Koehn et al. 2003] Koehn, P., Och, F.J., Marcu, D.: Statistical Phrase Based Translation. In: NAACL 2003 (2003)
12. [Papineni et al. 2002] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL 2002: 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
13. [Mathur & Paul 2009] Mathur, P., Paul, S.: Automatic Translation of Nominal Compounds from English to Hindi. In: The Proceedings of International Conference on Natural Language Processing, Hyderabad, ICON (2009)
14. [Patwardhan et. al. 2005] Patwardhan, S., Banerjee, S., Pedersen, T.: SenseRelate:TargetWord A Generalized Framework for Word Sense Disambiguation. In: Proceedings of the ACL Interactive Poster and Demonstration Sessions, Ann Arbor, MI (2005)
15. [Baldwin & Tanaka 2004] Baldwin, T., Tanaka, T.: Translation by Machine of Complex Nominals: Getting it Right. In: The Proceedings of ACL04 Workshop on Multiword Expression:Integrating Processing, Barcelona, Spain (2004)
16. [Rackow et al. 1992] Rackow, U., Dagan, I., Schwall, U.: Automatic Translation of Noun Compounds. In: COLING, pp. 1249–1253 (1992)

# Neoclassical Compound Alignments from Comparable Corpora

Rima Harastani, Béatrice Daille, and Emmanuel Morin

University of Nantes  
LINA, 2 Rue de la Houssinière  
BP 92208, 44322 Nantes, France  
`{Rima.Harastani,Beatrice.Daille,Emmanuel.Morin}@univ-nantes.fr`

**Abstract.** The paper deals with the automatic compilation of bilingual dictionary from specialized comparable corpora. We concentrate on a method to automatically extract and to align neoclassical compounds in two languages from comparable corpora. In order to do this, we assume that neoclassical compounds translate compositionally to neoclassical compounds from one language to another. The method covers the two main forms of neoclassical compounds and is split into three steps: extraction, generation, and selection. Our program takes as input a list of aligned neoclassical elements and a bilingual dictionary in two languages. We also align neoclassical compounds by a pivot language approach depending on the hypothesis that the neoclassical element remains stable in meaning across languages. We experiment with four languages: English, French, German, and Spanish using corpora in the domain of renewable energy; we obtain a precision of 96%.

## 1 Introduction

Describing new concepts usually requires creating new terms. Neoclassical word-formation is one process used by Romance and Germanic languages (among others) in order to produce domain-specific terms. It combines some elements borrowed from Greek or Latin, called neoclassical elements, to create neoclassical compounds. For example, combining the neoclassical elements *hydro* and *logy* leads to the neoclassical compound *hydrology*. New neoclassical elements could be borrowed when needed to form new terms. The productivity of neoclassical compounds, especially in scientific domains such as medicine, makes their translation difficult since many of them are unlikely to be found in bilingual dictionaries.

Many neoclassical compounds possess a compositional property (the meaning of the whole can be restored from the meaning of the parts) [1]. This property is as well valid for many complex terms (terms that consist of more than one component). For example, the complex term *washing machine* is in fact a machine designed to wash. Thus, some approaches have been proposed to translate complex terms depending on this compositional property [2] [3] [4] [5]. They translate a complex term by translating each of its components individually

using a bilingual dictionary. Then, they combine these individual translations according to some predefined templates to produce the final translation of the complex term. For example, a complex term that is of the form: [Adjective Noun] in English (e.g. comparable corpora), could be translated by a term that is of the form: [Noun Adjective] in French (e.g. corpus comparables). On the other hand, unlike components that compose complex terms, equivalents of neoclassical elements are not expected to be found in monolingual or bilingual dictionaries. For that reason, previous works depend on other resources than bilingual dictionaries to deal with neoclassical elements. For example, [6] use a pivot language (e.g. Japanese) in order to automatically acquire the semantical meaning of neoclassical elements. They suppose that neoclassical compounds are translated in Japanese to terms that consist of simple words. In this way, they align each neoclassical element with its equivalent simple word in Japanese. [7] build multilingual lists of neoclassical elements with their meanings and the relations between them. They define this list for each language in order to morphosyntactically analyze neoclassical compounds. [8] focuses on translating Italian constructed neologisms by prefixation into French. He relies on lexical resources and a set of bilingual lexeme formation rules in order to detect constructed neologisms and to generate their translations. Some of the differences between our work and his, is that [8] does not differentiate between native prefixes and neoclassical elements, he deals only with prefixation while we treat other forms of neologisms and experiment with four languages.

We propose a method to automatically extract and align neoclassical compounds from bilingual comparable corpora. Indeed, comparable corpora (collection of multilingual texts that belong to the same domain) have been successfully used for terminology alignment by many approaches [2] [3] for their advantages over parallel corpora (collection of multilingual texts that are translations of each other), such as their availability [9]. Identifying translations of neoclassical compounds can help in enriching bilingual dictionaries used by several applications such as Machine Translation tools (MT tools) and Computer-Aided Translation tools (CAT tools). We suppose that most neoclassical compounds in a source language translate compositionally to neoclassical compounds in a target language. We use a predefined aligned list of neoclassical elements between the two languages, and we define the template that will combine the translations of the individual parts as being the original Greco-Latin template in forming terms [10].

The paper is organized as follows. In section 2, we give a brief introduction to neoclassical compounds. Then, we explain the alignment method in section 3. We present an evaluation of this method in section 4. Finally, we conclude in section 5.

## 2 Neoclassical Compounds

We define neoclassical compounds as single-word terms consisting of at least one neoclassical element. Neoclassical elements or *combining forms* are elements

that are borrowed from Greek and Latin languages (e.g. *patho*, *bio*, *logy*, etc.). These elements are not considered as lexical units as they cannot play the role of independent words in a language syntax, i.e., they are always seen in the combined form with other elements (e.g. *biology*) [10] [11]. Each language may assimilate its borrowed neoclassical elements phonologically (but not totally) [12], in other words, a Greek or Latin word goes under a minimal adaptation before being adopted by a host language. For example, both FR<sup>1</sup> *pathie* and EN *pathy* were borrowed from the Greek word *pathos*. In addition, each element can have different allomorphs, which means that a borrowed Greek or Latin element can be assimilated to different forms in one language. For example, the English neoclassical element *neuro* can have two forms in French: *neuro* like in FR *neurologie* and *névro* like in FR *névrodermite*.

Neoclassical elements can appear at different positions in neoclassical compounds: (1) initial position in a neoclassical compound, like *homo-* in *homomorphic*, (2) final position such as *-cide* in *genocide*. According to [13], we distinguish between Initial Combining Forms (ICFs) and Final Combining Forms (FCFs). ICFs include forms of neoclassical elements that appear at initial positions (e.g. *bio-*, *cardio-*, *patho-*, etc.), while FCFs include forms of neoclassical elements that appear at final positions (e.g. *-logy*, *-cide*, *-pathy*, etc.). Moreover, more than one ICF may appear sequentially in a neoclassical compound (e.g. *histo-* and *patho-* in *histopathology*).

A neoclassical element (borrowed from the same Greco-Latin word) can be seen both as ICF and FCF, for instance, *patho-* in *pathology* and *-pathie* in *cardiopathie*, both elements being adapted from *pathos*. This property enables to distinguish between neoclassical elements and affixes. The latter appear at fixed positions: either at initial positions (prefixes, e.g. *pre-* like in *premature*) or at final positions (suffixes, e.g. *-ist* like in *chemist*). Furthermore, neoclassical elements have been introduced later than prefixes and suffixes to many European languages, around the 19th century to English [14].

### 3 Neoclassical Compound Alignment

In this section, we give the assumptions we make to align neoclassical compounds, the forms of neoclassical compounds that can be aligned, and finally we explain the main steps of our alignment approach.

#### 3.1 Assumptions

The method is based on the following assumptions:

##### Compositional Property

Neoclassical compounds are translated compositionally to neoclassical compounds. Each element in a neoclassical compound is translated individually and the final translation is the combination of the translated elements; as

<sup>1</sup> FR, EN, DE, and ES denote to French, English, German, and Spanish respectively.

the meaning of neoclassical compounds can be in many cases obtained compositionally [1]. For example, the translation of EN *hydrology* in French is *hydrologie*, which can be interpreted by the combination of the translation of the composing elements, *hydro* (water): FR *hydro* and *logy* (study): FR *logie*.

Translating a neoclassical compound compositionally to a neoclassical compound can give accurate results even in cases where a neoclassical compound is not fully compositional. From [15], we take EN *leukopathia* (a disease involving loss of melanin pigmentation of the skin) as an example of a neoclassical compound that is indeterminant by its elements; as its definition does not contain an explicit reference to *white* which is the meaning of its first composing element *leuko*. However, the equivalents of *leukopathia* in other languages are: FR *leucopathie*, DE *leukopathie*, ES *leucopatia*, this means that neoclassical compounds can still be translated compositionally even when their exact meaning cannot be restored compositionally.

### Preserving the Order of Elements

The order of the elements of a source neoclassical compound is preserved in the equivalent target neoclassical compound. Taking *hydrology* as an example, the equivalent of *hydro* will appear before the equivalent of *logy* when combining the equivalents of the constituent elements to form the final translation. This assumption is based on the fact that neoclassical word-formation in different languages follows the model of Greek and Latin languages in forming terms [10]. The Greco-Latin template for a term *XY* that consists of two elements *X* and *Y* is: [determinant determinatum]. According to this template, *cardiology* consists of *logy* (study) being the determinatum (identifies the class of which the neoclassical compound is a kind) and *cardio* (heart) being the determinant (gives the differentiating feature).

According to the second assumption, the order of elements should be respected when translating a neoclassical compound. Therefore, each neoclassical constituent element is translated with a neoclassical element of the same type (for instance an ICF by an ICF, an FCF by an FCF).

Hereafter, we assume that neoclassical compounds are either adjectives or nouns since this is true for most of the cases. In spite of the fact that some verbs can contain neoclassical elements, e.g. hydrogenate.

## 3.2 Handled Forms

Neoclassical compounds can take different forms. They contain at least one neoclassical element but also can contain native words and/or prefixes combined in different orders. Our method can only align neoclassical compounds that belong to one of the following forms:

### – ICF+ FCF

The first form includes neoclassical compounds that consist only of neoclassical elements. One or more ICFs can appear sequentially along with one

FCF. This form is equivalent to the combination of the first and the last forms presented in [11]. Examples of neoclassical compounds are given in (1).

- (1) FR histopathologie (*histo<sub>ICF</sub>* / *patho<sub>ICF</sub>* / *logie<sub>FCF</sub>*), EN radiology (*radio<sub>ICF</sub>* / *logy<sub>FCF</sub>*), DE radiometrie (*radio<sub>ICF</sub>* / *metrie<sub>FCF</sub>*), ES geomorfologia (*geo<sub>ICF</sub>* / *morf<sub>oICF</sub>* / *logia<sub>FCF</sub>*)

#### – ICF+ Word

This form includes one or more ICFs combined with a native word (lexical item that is a single word). This form is equivalent to combining the third, forth, and the last forms defined in [11]. Examples of neoclassical compounds are illustrated in (2).

- (2) FR cardiovasculaire (*cardio<sub>ICF</sub>* / *vasculaire<sub>Word</sub>*), EN photobioreactor (*photo<sub>ICF</sub>* / *bio<sub>ICF</sub>* / *reactor<sub>Word</sub>*), DE ferroelektrisch (*ferro<sub>ICF</sub>* / *elektrisch<sub>Word</sub>*), ES multidisciplinario (*multi<sub>ICF</sub>* / *disciplinario<sub>Word</sub>*)

### 3.3 Approach

The method that we propose firstly extracts neoclassical compounds for source and target languages from comparable corpora using two lists of neoclassical elements, the first for the source language ( $NE_{l_s}$ ) and the second for the target language ( $NE_{l_t}$ ); this results in lists of source and target neoclassical compound candidates:  $NC_{l_s}$  and  $NC_{l_t}$ . Then, each neoclassical compound in  $NC_{l_s}$  is aligned with its equivalent(s) in  $NC_{l_t}$ , with the help of a bilingual dictionary  $Dic_{B_i}$  and an aligned list of neoclassical elements  $NE_A$ . The method follows the three main steps of the compositional methods for aligning complex terms [2] [3]: i) the extraction of candidates, ii) the generation of translation candidates, and iii) the selection of correct translations.

#### 1. Extraction of Neoclassical Compound Candidates

Source and target neoclassical compound candidate lists ( $NC_{l_s}$  and  $NC_{l_t}$ ) are obtained by projecting  $NE_{l_s}$  on the corpus of the source language  $l_s$ , and  $NE_{l_t}$  on the corpus of the target language  $l_t$ . The adjectives or nouns that have at least one neoclassical element (ICF or FCF) are considered as neoclassical compound candidates. An ICF can appear in the beginning or anywhere in the middle of a neoclassical compound, e.g. ICFs *bio-*, *geo-* and *morpho-* appear in *biogeomorphological*. FCFs are found at the end of neoclassical compounds such as *-pathy* in *neuropathy* and *-logie* in *biotechnologie*.

#### 2. Generation of Translation Candidates

The projection made in the extraction phase results in decomposing each extracted neoclassical candidate into two or more elements, in which at least one of these elements is a potential neoclassical element. The form of a neoclassical candidate is checked, and in case it is identified as one of the two handled forms presented in section 3.2, the method will try to generate its translation candidates while respecting the assumptions explained in

section 3.1. Equivalents of identified ICFs and FCFs are found using  $NE_A$  whereas the translations of native words are obtained from  $Dic_{B_i}$ . The translation candidates are all the possible combinations (following the Greco-Latin template) of the translations of each element of the neoclassical compound candidate ( $NC_s$ ). The generation succeeds only if all elements of  $NC_s$  are identified. For example, suppose that we identify the two elements (*neuro-* and *-logy*) as neoclassical elements in the neoclassical compound EN *neurology*. To generate its French translation candidates, we search for the ICF equivalent(s) of EN *neuro-* in  $NE_A$ , which would be FR *neuro-* and FR *névro-*, as well as the FCF equivalent(s) of EN *-logy*, which would be FR *-logie*. Accordingly, two translation candidates will be generated by combining the translations of elements: *neurologie* and *névrologie*. Taking another example: we want to generate English translation candidates for FR *bio-science*, which matches the second form of neoclassical compounds. We can identify FR *bio* as neoclassical element and FR *science* as a word in the dictionary. The equivalent ICF of FR *bio* is EN *bio* that we could obtain from  $NE_A$ , while the translations of FR *science* in  $Dic_{B_i}$  could be *art*, *science*, *information*, *knowledge* and *learning*. Consequently, five translation candidates will be generated: *bioart*, *bioscience*, *bioinfomation*, *bioknowledge* and *biolearning*.

### 3. Selection of Correct Translations

We look up each translation candidate (obtained in the generation phase) in the target neoclassical compound list  $NC_{l_t}$ . In case the candidate is found, it will be considered as a correct translation for its respective source neoclassical compound  $NC_s$ . For example, if two French translation candidates were generated for EN *neurology*: *neurologie* and *névrologie*, they will be searched in the target neoclassical list  $NC_{l_t}$ . The candidate *névrologie* would not be found as it is not the correct translation, but there is a probability that *neurologie* would be found in  $NC_{l_t}$ , and therefore considered to be a valid translation.

The main steps of the method are summarized in 1.

---

#### Algorithm : Neoclassical compound alignment

---

```

NCls[] = ExtractNeoclassicalCompoundCandidates(Cs)
NClt[] = ExtractNeoClassicalCompoundCandidates(Ct)
for each NC in NCls
    Candidates[] = GenerateTranslationCandidates(NC)
    for each Candidate in Candidates
        if (Candidate exists in NClt)
            Select Candidate as translation for NC

```

---

**Fig. 1.** Algorithm for aligning neoclassical compounds. Cs = source corpus. Ct = target corpus.

## 4 Evaluation

We present in section 4.1 the resources that we used to do the experiments that led us to the results that we present in section 4.2.

### 4.1 Resources

We carried out the experiments using comparable corpora built with the BABOUK crawler [16]. The corpora are related to the renewable energy domain in four languages. We pre-processed each corpus by running a word tokenizer, a POS tagger, and a lemmatizer. Table 1 lists the languages with the corresponding number of unique nouns and adjectives in the corpus.

**Table 1.** Corpora statistics

Language	No. of unique adjectives and nouns
English	24,250
French	13,625
German	51,624
Spanish	15,785

As for neoclassical elements, we have taken 113 French neoclassical elements from [17]. Then, we have manually aligned 83 of these neoclassical elements with their English equivalents. We then aligned 61 English neoclassical elements with their German equivalents as well as 58 English neoclassical elements with their Spanish equivalents. This led us to obtaining three lists of aligned neoclassical elements for the pairs of languages (EN-FR, EN-DE, and EN-ES), and four lists of monolingual neoclassical elements (see Table 2). We have also used three bilingual dictionaries for EN-FR, EN-DE, and EN-ES that contain 145,542, 69,876, and 61,587 single-word entries respectively<sup>2</sup>.

**Table 2.** Sizes of monolingual neoclassical element lists

	EN	FR	DE	ES
NE size	83	113	61	58

<sup>2</sup> Dictionaries were obtained from EURADIC French-English dictionary [http://catalog.elra.info/product\\_info.php?products\\_id=666](http://catalog.elra.info/product_info.php?products_id=666), <http://www.dict.cc/>, and <http://www.phrozenSmoke.com/projects/pythonol/>

## 4.2 Results and Discussion

Tables 3 and 4 present the results of the experiments carried out on the method for the three pairs of languages (EN-FR, EN-ES, and EN-DE) in both directions. For example, Table 1 shows that we were able to extract 1215 English neoclassical compound candidates using the English neoclassical element list. Although many of these are false neoclassical compounds (e.g. *decision*, *communication*). For the pair of languages EN-FR, French translation candidates were automatically generated for 264 of the English extracted neoclassical compound candidates when using 83 FR-EN neoclassical aligned elements and the FR-EN bilingual dictionary (presented in section 4.1). The correct translation(s) among the generated candidates were found in the target neoclassical compound list for 100 of the 264 candidates. A generated translation candidate that is not found does not necessarily mean that it is a wrong translation; it just could possibly be a correct translation that is missing from the target corpus. The precision obtained from the alignment was about 98%, and the recall was about 37%. The recall was calculated by dividing the number of true positives (correct translations) on the sum of true positives and false negatives (identified neoclassical compounds with no suggested translation).

For all languages, false positive alignments were mainly the translations that were obtained from false extracted neoclassical compounds (noise, non-neoclassical compounds), e.g. the French word *histoire* was extracted from English corpus since *histo* was identified as neoclassical element and *ire* was identified as English word. Thus, *histoire* was aligned with FR *histoire* (*ire* is a French translation of EN *ire*). Erroneous translations were also obtained because of the fact that neoclassical compounds are not always translated to neoclassical compounds from one language to another, e.g. FR *télécommande* was translated to EN *telecontrol*, while the correct translation is EN *remote control*.

**Table 3.** Alignment of neoclassical compounds for (EN-FR, EN-DE, and EN-ES)

Languages	Aligned neoclassical elements	Neoclassical compound candidates	Generated translations	Found translations	Precision	Recall
EN-FR	83	1215	264	100	98%	37%
EN-DE	61	1215	266	100	96%	36%
EN-ES	58	1215	219	68	97%	30%

A neoclassical compound candidate  $NC_s$  can be extracted (since it contains a possible neoclassical element) but still cannot be identified as one of the neoclassical forms our method handles, the generation of its translation candidates will fail. This can be due to several reasons:

- **False neoclassical element:** a candidate like EN *decision* will be decomposed into two elements; the first of which is *deci* will be considered as

**Table 4.** Alignment of neoclassical compounds for (FR-EN, DE-EN, and ES-EN)

Languages Aligned	Neoclassical		Generated	Found	Precision	Recall
	neoclassical elements	compound candidates	translations	translations		
FR-EN	83	1068	263	94	97%	35%
DE-EN	61	3538	437	105	96%	23%
ES-EN	58	2126	363	69	97%	18%

neoclassical element (false neoclassical element). The second is *sion* which is neither a neoclassical element nor a known word in the bilingual dictionary.

- **Missing neoclassical element from  $NE_A$ :** if a candidate like FR *métronome* is extracted and only the equivalent of *métro* is found in  $NE_A$ , the generation will fail because the equivalent(s) of *nome* (a real neoclassical element) are not found in  $NE_A$ .
- **Untreated neoclassical form:** a true neoclassical candidate could be extracted but it belongs to a form that we do not handle. There exist other forms of neoclassical elements, for example, EN *antibiogram* (*anti*: prefix, *bio*: ICF, *gram*: FCF) is a form the method does not cover.

### Bilingual Alignment Using a Pivot Language

We can obtain bilingual lists of neoclassical compounds using a pivot language. Generally speaking, source-to-target translation of a word through a pivot language occurs in two steps: (1) the word is translated to the pivot language using a bilingual source-to-pivot dictionary, (2) the obtained translation from the previous step is translated to the target language using a bilingual pivot-to-target dictionary. Pivot language approach is known to be highly noisy because of polysemy in languages and intransitivity of lexicons. However, bilingual alignments of neoclassical compounds through a pivot language should be as precise as bilingual neoclassical compound alignments obtained by our method (presented in section 3) because neoclassical elements remain stable in meaning across languages.

We chose EN as pivot language to obtain a list of FR-DE alignments as well as ES-FR and ES-DE alignments using the aligned lists of neoclassical compounds for (EN-FR, EN-DE, and EN-ES). Accordingly, we obtained the results shown in Table 5. We conclude that using a pivot language to align neoclassical elements is a confident approach as languages follow the Greco-Latin template when creating neoclassical compounds.

**Table 5.** Alignment of neoclassical compounds using English as pivot language

Languages	Alignments	Precision
FR-DE	61	98%
ES-FR	50	100%
ES-DE	44	97%

## 5 Conclusion

In this paper, we presented a compositional-like method to align neoclassical compounds in two languages (source-target). The method handles two forms of neoclassical compounds. For this task, it uses predefined monolingual neoclassical elements to extract neoclassical compound candidates, and a list of aligned neoclassical elements in addition to a bilingual dictionary to align the extracted candidates. The results showed high precision (more than 96%) in aligning neoclassical compounds of the two handled structures. Moreover, we showed that a pivot language approach gives high-precision neoclassical compound alignments too. We aim at expanding the method so that it covers other possible forms of neoclassical compounds. We also aim to investigate the possibility of automatically extracting neoclassical elements for one language and aligning them with their equivalents in another language.

## References

1. Estopa, R., Vivaldi, J., Cabré, M.T.: Use of greek and latin forms for term detection. In: The 2nd International Conference on Language Resources and Evaluation, vol. 78, pp. 885–859 (2000)
2. Robitaille, X., Sasaki, Y., Tonoike, M., Sato, S., Utsuro, T.: Compiling french-japanese terminologies from the web. In: EACL, pp. 225–232 (2006)
3. Baldwin, T., Tanaka, T.: Translation by machine of complex nominals: Getting it right. In: ACL Workshop on Multiword Expressions: Integrating Processing, pp. 24–31 (2004)
4. Vintar, S.: Bilingual term recognition revisited the bag-of-equivalents term alignment approach and its evaluation. Terminology 16, 141–158 (2010)
5. Grefenstette, G.: The world wide web as a resource for example-based machine translation tasks. In: Proceedings of the ASLIB Conference on Translating and the Computer, London, vol. 21 (1999)
6. Vincent, C., Ewa, K.: Analyse morphologique en terminologie biomédicale par alignement et apprentissage non-supervisé. In: Conférence Traitement automatique des langues naturelles TALN, Montréal, Québec, Canada (2010)
7. Namer, F., Baud, R.H.: Defining and relating biomedical terms: Towards a cross-language morphosemantics-based system. I. J. Medical Informatics, 226–233 (2007)
8. Cartoni, B.: Lexical morphology in machine translation: A feasibility study. In: EACL, pp. 130–138 (2009)
9. Bowker, L., Pearson, J.: Working with specialized language: a practical guide to using corpora. Routledge, London (2002)

10. Amiot, D., Dal, G.: *La composition néoclassique en français et l'ordre des constituants. La composition dans les langues*. Artois Presses Université, 89–113 (2008)
11. Namer, F.: *Morphologie, lexique et traitement automatique des langues*. Lavoisier (2009)
12. Lüdeling, A.: *Neoclassical word-formation*. In: Keith Brown (ed) *Encyclopedia of Language and Linguistics*, 2nd edn., Elsevier, Oxford (2006)
13. Bauer, L.: *English word-formation*. Cambridge University Press (1983)
14. Baeskow, H.: *Lexical properties of selected non-native morphemes of English*. Gunter Narr Verlag (2004)
15. McCray, A., Browne, A., Moore, D.: The semantic structure of neo-classical compounds. In: *The Annual Symposium on Computer Application in Medical Care*, pp. 165–168 (1988)
16. Groc, C.D.: Babouk: Focused Web Crawling for Corpus Compilation and Automatic Terminology Extraction. In: *The IEEEWICACM International Conferences on Web Intelligence*, Lyon, France, pp. 497–498 (2011)
17. Béchade, H.D.: *Phonétique et morphologie du français moderne et contemporain*. Presses Universitaires de France (1992)

# QAlign: A New Method for Bilingual Lexicon Extraction from Comparable Corpora

Amir Hazem and Emmanuel Morin

Laboratoire d’Informatique de Nantes-Atlantique (LINA)

Université de Nantes, 44322 Nantes Cedex 3, France

{Amir.Hazem, Emmanuel.Morin}@univ-nantes.fr

**Abstract.** In this paper, we present a new way of looking at the problem of bilingual lexicon extraction from comparable corpora, mainly inspired from information retrieval (IR) domain and more specifically, from question-answering systems (QAS). By analogy to QAS, we consider a word to be translated as a part of a question extracted from a source language, and we try to find out the correct translation assuming that it is contained in the correct answer of that question extracted from the target language. The methods traditionally dedicated to the task of bilingual lexicon extraction from comparable corpora tend to represent the whole contexts of a word in a single vector and thus, give a general representation of all its contexts. We believe that a local representation of the contexts of a word, given by a window that corresponds to the query, is more appropriate as we give more importance to local information that could be swallowed up in the volume if represented and treated in a single whole context vector. We show that the empirical results obtained are competitive with the standard approach traditionally dedicated to this task.

**Keywords:** Comparable corpora, bilingual lexicon extraction.

## 1 Introduction

The use of comparable corpora for the task of bilingual lexicon extraction has attracted great interest since the beginning of 1990. Introduced by [25] he assumes that algorithms for sentence and word alignment from parallel texts should also work for non parallel and even unrelated texts. Comparable corpora offer a great alternative to the inconvenience of parallel corpora. Parallel corpora are not always available and are also difficult to collect especially for language pairs not involving English and for specific domains, despite many previous efforts in compiling parallel corpora (Church & Mercer, 1993; Armstrong & Thompson, 1995). According to Rapp [25] : *The availability of a large enough parallel corpus in a specific field and for a given pair of languages will always be the exception, not the rule.* Since then, many investigations and a number of studies have emerged, [10,11,12,24,26,2,7,14,23,21, among others]. All these works are based on a general representation of the contexts of a given word by collecting all its co occurrences in a single large vector. We want to give particular attention to each context as it represents a specific idea that can be lost if treated in a whole context vector. QAS systems alleviate this drawback and offer a suitable environment for our

task. Basically, the aim of a question answering system is to find the correct answer to a given question. The main idea of such a QAS is to consider segments or paragraphs of documents that share several words with a given question and then order them according to a similarity measure [15]. Those  $n$  best segments are most likely to provide the correct answer. Complex systems will not only use the words of the question but also synonyms or other semantically related words. More sophisticated systems will reformulate the question and so on [28][19][20][22][16]. In a multilingual context, the question is first translated and then the same treatments are applied as stated previously. In our case, we want to push QAS systems a step further by considering the bilingual lexicon extraction from comparable corpora as a question answering system, where the question is one of the contexts of the word to be translated, and the best answer should be the one containing the correct translation in the target language. In this case and for a given word we have as many questions as this word occur. This can be a problem if a word has a high frequency. We obviously cannot consider all the contexts of such a word, this is not our aim. On the contrary we will consider the  $n$  best contexts which is one part of the problem that we have to deal with. The remainder of this paper is organised as follows. Section 2 presents the standard approach based on lexical context vectors dedicated to word alignment from comparable corpora. Section 3 describes our Q-Align approach that can be viewed as a question answering system for alignment. Section 4 describes the different linguistic resources used in our experiments. Section 5 evaluates the contribution of the standard and Q-Align approaches on the quality of bilingual terminology extraction through different experiments. Section 6 presents our discussion and finally, Section 7 presents our conclusions and some perspectives.

## 2 Standard Approach

The main work in bilingual lexicon extraction from comparable corpora is based on lexical context analysis and relies on the simple observation that a word and its translation tend to appear in the same lexical contexts. The basis of this observation consists in the identification of first-order affinities for each source and target language: “*First-order affinities describe what other words are likely to be found in the immediate vicinity of a given word*” [17, p. 279]. These affinities can be represented by context vectors, and each vector element represents a word which occurs within the window of the word to be translated (for instance a seven-word window approximates syntactical dependencies). The implementation of this approach can be carried out by applying the four following steps [25,13]:

**Context Characterisation.** All the lexical units in the context of each lexical unit  $i$  are collected, and their frequency in a window of  $n$  words around  $i$  extracted. For each lexical unit  $i$  of the source and the target languages, we obtain a context vector  $\mathbf{i}$  where each entry,  $i_j$ , of the vector is given by a function of the co-occurrences of units  $j$  and  $i$ . Usually, association measures such as the mutual information [9] or the log-likelihood [8] are used to define vector entries.

**Vector Transfer.** The lexical units of the context vector  $\mathbf{i}$  are translated using a bilingual dictionary. Whenever the bilingual dictionary provides several translations for a lexical unit, all the entries are considered but weighted according to their frequency in the target language. Lexical units with no entry in the dictionary are discarded.

**Target Language Vector Matching.** A similarity measure,  $\text{sim}(\bar{\mathbf{i}}, \mathbf{t})$ , is used to score each lexical unit,  $t$ , in the target language with respect to the translated context vector,  $\bar{\mathbf{i}}$ . Usual measures of vector similarity include the cosine similarity [27] or the weighted jaccard index (WJ) [18] for instance.

**Candidate Translation.** The candidate translations of a lexical unit are the target lexical units ranked following the similarity score. The translation of the lexical units of the context vectors, which depends on the coverage of the bilingual dictionary vis-à-vis the corpus, is an important step of the standard approach, as more elements of the context vector are translated, the context vector will be more discriminating in selecting translations in the target language. This drawback can be partially circumvented by combining a general bilingual dictionary with a specialised bilingual dictionary or a multilingual thesaurus [3,7]. Moreover, this approach is sensitive to the choice of parameters such as the size of the context, the choice of the association and similarity measures. The most complete study about the influence of these parameters on the quality of bilingual alignment has been carried out in [21]. Another approach has been proposed to avoid the insufficient coverage of the bilingual dictionary required for the translation step of the standard approach [7,5]. The basic intuition of this approach is that words that have the same meaning will share the same environments. Here, the approach consists in the identification of “Second-order affinities” for the source language: “*Second-order affinities show which words share the same environments. Words sharing second-order affinities need never appear together themselves, but their environments are similar*” [17, p. 280]. For a word to be translated its affinities can be extracted through distributional techniques. In this case, the translation of a word consists of the translation of similar words. Since this approach is sensitive to the size of the comparable corpus, this study focuses on the standard approach.

### 3 Q-Align Approach

The Q-Align approach is described in three steps as follows :

#### 3.1 Collecting the Queries

The first step of the Q-Align approach, is to collect the set of all the windows (queries) in which a word to be translated appears. The size of this set corresponds to the frequency of the candidate. We have to deal with two parameters, the first one is the size of each query, it can be seen as the window surrounding the word to be translated as usually done in the state of the art. Let us call this parameter  $w_q$ . For instance, let us take **replica** as the word to translate, if  $w_q = 5$  this means that there are two words on the

**Table 1.** English query of the word **replica**

<i>detail<sub>V</sub></i>	<i>paintings</i>	<b>replica<sub>S</sub></b>	<i>line<sub>V</sub></i>	<i>separate<sub>J</sub></i>
---------------------------	------------------	----------------------------	-------------------------	-----------------------------

left of **replica** and two words on its right. After the POS-Tagging and filtering process, we obtain the resulting query for the word **replica** as shown in Table 1.

The second parameter is the number of queries we need for our task. We start from the assumption that not all contexts are useful when trying to find the correct translation. On the contrary, some of them are useless and can be considered as noise. Following this principle, we believe that a good choice of a context maximises the chances of matching the correct translation. Several ways can be followed to deal with this parameter in order to find the best tuning. As we wanted to focus on the comparison between the standard and Q-Align approaches in term on context characterisation, we did not investigate the different possibilities of choosing the number of queries, which is on its own a great matter of interest for future work. We fixed this parameter empirically. The choice of the  $n$  best queries was merely done following equation 1 :

$$Score(query_n) = \sum_{i=1}^{w_q-1} freq(word_i) \quad (1)$$

After applying the calculation of the score for all the queries, we sorted in a decremental order the  $n$  queries according to  $Score(query_n)$ .

### 3.2 Translation of Queries

Each collected query has to be translated into the target language, if we use the previous example of the word **replica**, and if we consider French as the target language, we obtain the corresponding translation query in Table 2 :

**Table 2.** Representation of the English query of the word **replica** and its translation into French

Word	Translation
<i>detail<sub>V</sub></i>	<i>désigner<sub>V</sub></i>
<i>paintings</i>	<i>peintures<sub>S</sub></i>
<b>replica<sub>S</sub></b>	<b>Unknown<sub>S</sub></b>
<i>line<sub>V</sub></i>	<i>marquer<sub>V</sub></i>
<i>separate<sub>J</sub></i>	<i>indépendant<sub>J</sub></i>

The translated query that will be used in the target language is given in Table 3 :

**Table 3.** Translated query of the word **replica**

<i>désigner<sub>V</sub></i>	<i>peintures<sub>S</sub></i>	<i>marquer<sub>V</sub></i>	<i>indépendant<sub>J</sub></i>
-----------------------------	------------------------------	----------------------------	--------------------------------

It is worth noting that the words of the query are translated using a bilingual dictionary while preserving the POS-Tagging relation of each translation pair. When several translations for a word are given, we consider the one with the highest frequency in the target language. Words with no entry in the dictionary are discarded.

### 3.3 Extraction of the Translation Candidates

To select a translation candidate, we use the compactness [28] as similarity measure. The principle of compactness in QAS is to measure a similarity between a question and a given segment. A segment can be : a sentence, a paragraph or a document. In our case, and by analogy, we measure the compactness between a translated query and a given segment of a given document in the target corpus. The final compactness  $Compact_{All}(\bar{w}_x)$  of  $\bar{w}_x$  is simply the sum of its compactness according to all translated queries, as given by the following equation :

$$Compact_{All}(\bar{w}_x) = \sum_{i \in nbQuery} Compact(\bar{w}_x)_i \quad (2)$$

All the documents of the target language are divided into segments. We investigate each segment to find out if it contains the correct translation. We need to fix the size of the segments. Let us denote  $w_{seg}$  as the size of a given segment corresponding to the number of words that belongs to this segment. For a given translated query and a given segment, the compactness of  $\bar{w}_x$  for a segment  $s$  is given by :

$$Compact_s(\bar{w}_x) = \frac{1}{|WQ|} \sum_{i \in WQ} Contrib(w_i)_{\bar{w}_x} \quad (3)$$

where  $Contrib(w_i)_{\bar{w}_x}$  is the contribution of each word of the query. Let us give an example to illustrate how to compute the contribution and the compactness. We denote  $QR$  as the set of words of the translated query as shown in table 4, with  $w_q = 5$  and  $w_i$  a word of the given query. In the example  $QR = \{w_1, w_2, w_3, w_4\}$  and  $\mathbf{Cand}_S$  is the word to be translated.

**Table 4.** English query of the word to be translated

$w_1$	$w_2$	$\mathbf{Cand}_S$	$w_3$	$w_4$
-------	-------	-------------------	-------	-------

Let us consider also, a segment with  $w_{seg} = 8$ . Each word of the segment which is not part of the question is considered as a translation candidate, we can take  $\bar{w}_x$  as a candidate :

We compute the contribution of each word  $w_i \in QR$  surrounding  $\bar{w}_x$  following this equation:

$$Contrib(w_i)_{\bar{w}_x} = \frac{|Z|}{D + 1} \quad (4)$$

**Table 5.** Representation of a given segment

$w_1$			$w_2$	$\bar{w}_x$		$w_3$	$\bar{w}_4$	$w_4$
-4	-3	-2	-1	0	1	2	3	4

Where :  $D = \text{distance}(w_i, \bar{w}_x) = |\text{pos}(w_i) - \text{pos}(\bar{w}_x)|$

$\text{pos}(w_i)$  is the position of  $w_i$  in a given segment, for instance, in Table 5,  $\text{pos}(w_1) = -4$ .

$Z = \{Y \setminus \text{distance}(Y, \bar{w}_x) < D \text{ and } Y \in QR\} \cup \{\bar{w}_x\}$

For example the contribution of  $w_1$  is given by :

$$\text{Contrib}(w_1)_{\bar{w}_x} = \frac{2+1}{4+1} = \frac{3}{5} \quad (5)$$

We wanted to consider differently the words of a given translated query, one way was to weight the contribution of a word by its Inverse Segment Frequency (ISF) by analogy to Inverse Document Frequency (IDF)[16][15] , assuming that words with high ISF should be more important. This can be seen in equation 6 :

$$\text{Compact}_s(\bar{w}_x) = \frac{1}{|WQ|} \sum_{i \in WQ} \text{ISF}(w_i) \times \text{Contrib}(w_i)_{\bar{w}_x} \quad (6)$$

We have given above the compactness of a word computed from a given segment. As there is thousands of segments in a corpus, we chose the maximum compactness of a given word according to equation 7. Some other alternatives have been explored as the mean compactness or the sum but no significant differences or improvements have been noticed.

$$\text{Compact}(\bar{w}_x) = \max(\text{Compact}_s(\bar{w}_x)) \quad (7)$$

Starting from the intuition that the translation of a rare word in a source language should also be rare in the target language following the principle of comparable corpora, we weighted the final compactness for rare words by the ISF. This is represented in equation 8 :

$$\text{Compact}_{All}(\bar{w}_x) = \sum_{i \in nbQuery} \text{ISF}(\bar{w}_x) \times \text{Compact}(\bar{w}_x)_i \quad (8)$$

No other alternative except the weighted sum showed a significant improvements, but more investigations have to be conducted especially on the choice of the good queries. This represents our next challenge.

## 4 Linguistic Resources

The experiments have been conducted on two different French-English corpora: a specialised corpus from the medical domain within the sub-domain of 'breast cancer' and a general corpus from newspapers 'LeMonde/New-York Times'. Due to the small size

of the specialised corpus we wanted to conduct additional experiments on a large corpus to have a better idea of the behaviour of our approach. Both corpora have been normalised through the following linguistic pre-processing steps: tokenisation, part-of-speech tagging, and lemmatisation. The function words have been removed and the words occurring less than twice (i.e. hapax) in the French and the English parts have been discarded.

#### 4.1 Specialized Corpus

We have selected the documents from the Elsevier website<sup>1</sup> in order to obtain a French-English specialised comparable corpus. We have automatically selected the documents published between 2001 and 2008 where the title or the keywords contain the term ‘cancer du sein’ in French and ‘breast cancer’ in English. We collected 130 documents in French and 118 in English and about 530,000 words for each language. The comparable corpus comprised about 7,400 distinct words in French and 8,200 in English. In bilingual terminology extraction from specialised comparable corpora, the terminology reference list required to evaluate the performance of the alignment programs is often composed of 100 single-word terms (SWTs) (180 SWTs in [6], 95 SWTs in [2], and 100 SWTs in [5]). To build our reference list, we selected 400 French/English SWTs from the UMLS<sup>2</sup> meta-thesaurus and the *Grand dictionnaire terminologique*<sup>3</sup>. We kept only the French/English pair of SWTs which occur more than five times in each part of the comparable corpus. As a result of filtering, 122 French/English SWTs were extracted.

#### 4.2 General Corpus

We chose newspapers as they offer a large amount of data. We selected the documents from the French newspaper ‘Le Monde’ and the English newspaper ‘The New-York Times’. We automatically selected the documents published between 2004 and 2007 and obtained 5 million words for each language. The comparable corpus comprised about 70,400 distinct words in French and 80,200 in English. The terminology reference list is much more consequential and contains 1004 SWTs, it has been extracted from ELRA-M0033. We divided this list into 8 sub-lists according to word frequency as presented in Table 6 :

#### 4.3 Bilingual Dictionary

The French-English bilingual dictionary required for the translation phase was the ELRA-M0033 dictionary. It contains, after linguistic pre-processing steps, 32,000 English single words belonging to the general language with an average of 1.6 translations per entry.

<sup>1</sup> [www.elsevier.com](http://www.elsevier.com)

<sup>2</sup> [www.nlm.nih.gov/research/umls](http://www.nlm.nih.gov/research/umls)

<sup>3</sup> [www.granddictionnaire.com/](http://www.granddictionnaire.com/)

**Table 6.** Representation of each evaluation list

Name List	Interval	#occ
<i>List_sup_1000</i>	$\#occ > 1000$	4
<i>List_500_1000</i>	[500, 1000[	20
<i>List_100_500</i>	[100, 500[	180
<i>List_50_100</i>	[50, 100[	200
<i>List_10_50</i>	[10, 50[	400
<i>List_2_10</i>	[2, 10[	200

## 5 Experiments and Results

In this section, we first give the parameters of the standard and Q-Align approaches, than we present the results conducted on the two corpora presented above: "Breast cancer" and "LeMonde/New-YorkTimes".

### 5.1 Experimental Setup

Three major parameters need to be set to the standard approach, namely the similarity measure, the association measure defining the entry vectors and the size of the window used to build the context vectors. Laroche and Langlais [21] carried out a complete study of the influence of these parameters on the quality of bilingual alignment. As a similarity measure, we chose to use the weighted jaccard index [18]. The entries of the context vectors were determined by the log-likelihood [8], and we used a seven-word window since it approximates syntactic dependencies. Other combinations of parameters were assessed but the previous parameters turned out to give the best performance. For the Q-Align approach we also used a seven-word window that corresponds to the query length. The size of segments in the target language was fixed to one hundred words even if several combinations were assessed. This size gave the best performance on a fixed length query of seven words. The choice of one hundred as the length of a segment is due to the fact that it is more or less the length of a paragraph.

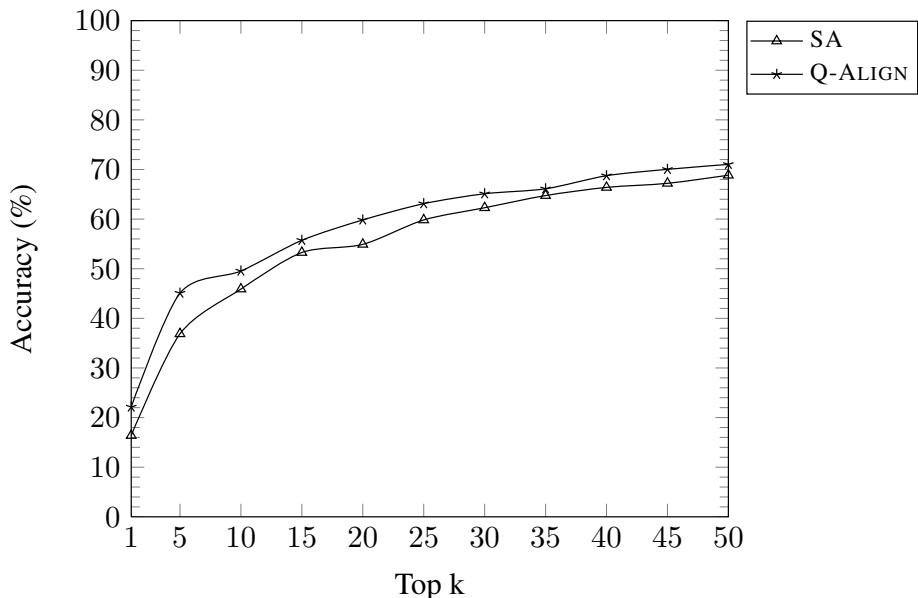
### 5.2 Results

To evaluate the performance of our approach, we used the standard approach (SA) proposed by [26] as a baseline. The accuracy is given in percentage in all the graphics.

#### Evaluation on the Breast Cancer Corpus

We investigate the performance of the Standard and Q-Align approaches on the breast cancer corpus, using the evaluation list of 122 words.

We can see in Figure 1 that Q-Align approach always outperforms the standard approach for all values of  $k$ . The accuracy at the top 20 for the standard approach is 54.91% while Q-Align approach gives 60.62%. The Q-Align model can be considered as a competitive approach according to its results as shown in Figure 1 for the breast cancer corpus.



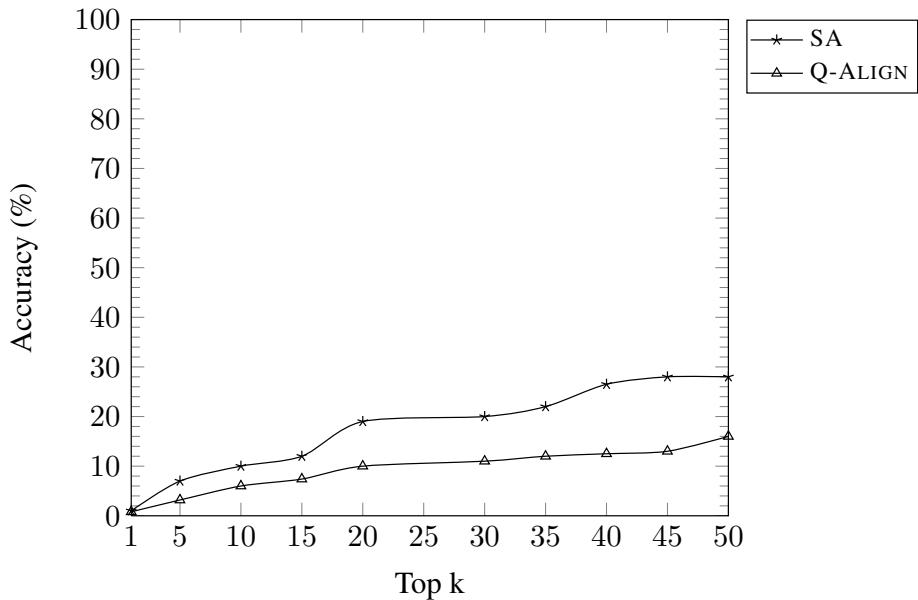
**Fig. 1.** Accuracy at top k for the breast cancer corpus (SA vs Q-Align)

### Evaluation on the LeMonde/New-YorkTimes Corpus

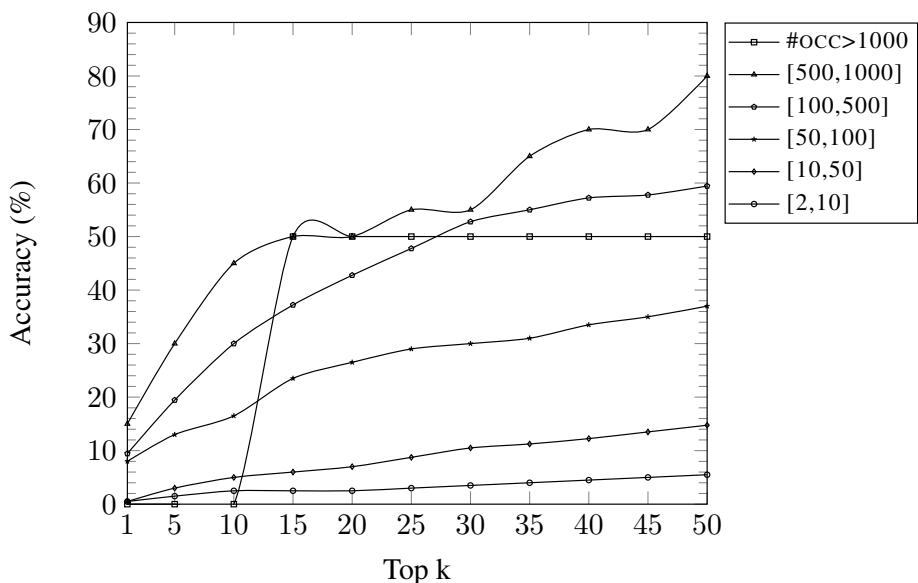
We then investigate the performance of the Standard and Q-Align approaches on LeMonde/New-YorkTimes corpus, using an evaluation list of 1004 words.

Let us see in Figures 3 and 4 the details of the ranges of frequency into which Q-Align and standard approaches failed.

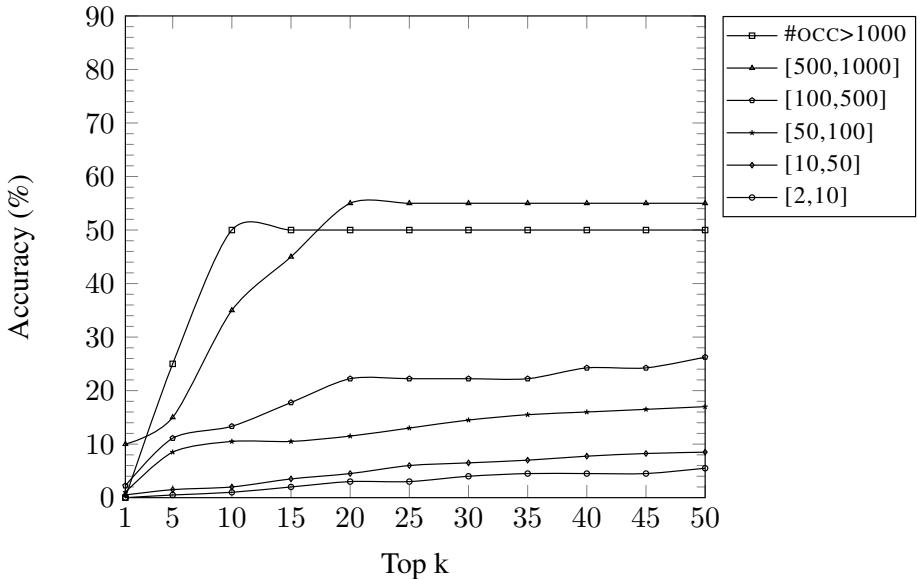
Figure 3 and 4 show that both approaches are sensitive to the variations of word's frequencies. It seems that the Q-Align approach is slightly less efficient for rare words with frequencies less than 50 while the standard approach (SA) is slightly better. Similarly for very frequent words with frequencies higher than 500 the standard approach outperforms Q-Align approach except for top 1, 5 and 10. The main gap between SA and Q-Align in term of accuracy can be seen for the lists where the words frequencies are between 50 and 500. Due to the small list of words with frequencies higher than 1000, we cannot give an appropriate conclusion for both approaches as the number of words in this list is equal to 4. The main reason for the weakness of the Q-Align approach in a general corpus is probably the lack of markers or seed words that are more present in a specialised corpus. In the light of these results more investigations have to be conducted on general corpus to improve the performance of the Q-Align approach.



**Fig. 2.** Accuracy at top k for LeMonde/NewYorkTimes (SA vs Q-Align)



**Fig. 3.** Accuracy at top k for LeMonde/NewYorkTimes (Standard Approach)



**Fig. 4.** Accuracy at top k for LeMonde/NewYorkTimes (Q-Align Approach)

## 6 Discussion

The aim of this work was not to try to find the best results by looking for the best tuning of each method (SA, Q-Align). Here, the main interest was to show another way of looking at the task of bilingual lexicon extraction from comparable corpora by looking at local information captured in a given segment while the standard approach looks at global information captured in the whole corpus. The Q-Align approach imitates QAS systems by choosing a query in the source language and tries to find an answer in a particular segment of the target language which contains the correct translation. This process was done by comparing the translated query with segments that we consider more or less close to paragraphs in the target language, assuming that if a given paragraph shares some words with the translated query then, there is more chance that this paragraph contains the correct translation.

It is important to say that Q-Align is a naive approach. The process of looking for the right translation does not take into account any linguistic or semantic information, it is just based on words that are in common between a query and a segment. Surprisingly, this naive approach (Q-Align) outperforms the standard approach (SA) for each top on the specialised corpus. Thus, there is much to do to improve the Q-Align approach while no semantic or linguistic information was taken into account. Q-Align can be considered as promising for future work. Many improvements need to be done. QAS systems are a great source of inspiration for it.

It would be interesting to merge both approaches to see whether there is a complementarity or not between them and if obviously there is an increase in accuracy.

We can reflect upon a double interest by using both approaches. The first one is given by the standard approach (SA) as it gives a global view and a global representation of information. The second one is given by the Q-Align approach which gives a local view and a local representation of information. We can imagine that both local and global information could be useful taken together to improve the representation of information and thus to obtain increased accuracy.

In the Q-Align approach one drawback is probably the choice of queries. Indeed, not all the queries are useful when trying to find the right translation. Some of them bring more confusion than good information. We can consider these queries as noise, because if taken, they could lead us to wrong translations. Following this principle, we believe that a good choice of a query maximises the chances of matching the right translation. Several alternatives to our arbitrary choice of the number of queries can be applied in order to find the best tuning. In this paper we did not explore the way of choosing the number of queries. This question remains opened and represents one of the unanswered questions. It is for us one of the next challenges.

We must add that our research concentrated on specialised corpora as it is our main center of interest. We were however curious to see the behaviour of our approach on a general corpus and the results were as expected. At the moment, Q-Align approach is more appropriate to a specific rather than to a general domain as all our efforts were conducted in this way. This performance can be explained by the specificity of specialised corpora such as the medical domain for instance, which contains strong markers that are for the greater part technical words or words specific to the domain. These markers that we can see as seed words are very useful for the Q-Align approach. The results obtained clearly point to one conclusion which is that Q-Align approach is more appropriate for corpora of specialised domains while for general corpora it remains unstable due to the lack of specific markers. More efforts on general domain data have to be made to adapt Q-Align approach to general domain corpora.

## 7 Conclusion

We have presented a novel way of looking at the problem of bilingual lexicon extraction from comparable corpora based on the principle of question answering systems. We explored two different corpora, the first concerned a corpus of medical domain (Brest Cancer) and the second concerned the corpus of newspapers (LeMonde/New-YorkTimes). Regarding the empirical results of our proposition, performances were better than the baseline proposed by [26] on specialised corpus. While the standard approach remained more robust in a general domain corpora. Further research is certainly needed but our current findings support the idea that local information has its importance and should not be neglected for the task of bilingual lexicon extraction from comparable corpora. We believe that our model is simple and sound. The most significant result is that the new approach to finding single word translations has been shown to be competitive and promising for future work. We hope that this new paradigm can lead to insights that could be unclear in other models. Dealing with this problem is an interesting line for future research.

**Acknowledgments.** The research leading to these results has received funding from the French National Research Agency under grant ANR-08-CORD-013.

## References

1. Armstrong, S., Thompson, H.: A presentation of MLCC: Multilingual Corpora for Cooperation. In: Linguistic Database Workshop, Groningen (1995)
2. Chiao, Y.C., Zweigenbaum, P.: Looking for candidate translational equivalents in specialized, comparable corpora. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Tapei, Taiwan, pp. 1208–1212 (2002)
3. Chiao, Y.C., Zweigenbaum, P.: The Effect of a General Lexicon in Corpus-Based Identification of French-English Medical Word Translations. In: Baud, R., Fieschi, M., Le Beux, P., Ruch, P. (eds.) The New Navigators: from Professionals to Patients, Actes Medical Informatics Europe. Studies in Health Technology and Informatics, vol. 95, pp. 397–402. IOS Press, Amsterdam (2003)
4. Church, K.W., Mercer, R.L.: Introduction to the Special Issue on Computational Linguistics Using Large Corpora. Computational Linguistics 19(1), 1–24 (1993), <http://dblp.uni-trier.de>
5. Daille, B., Morin, E.: French-English Terminology Extraction from Comparable Corpora. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCLNP 2005), Jeju Island, Korea, pp. 707–718 (2005)
6. Déjean, H., Gaussier, E.: Une nouvelle approche à l'extraction de lexiques bilingues à partir de corpus comparables. Lexicometrica, Alignement lexical dans les corpus multilingues, pp. 1–22 (2002)
7. Déjean, H., Sadat, F., Gaussier, E.: An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Tapei, Taiwan, pp. 218–224 (2002)
8. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)
9. Fano, R.M.: Transmission of Information: A Statistical Theory of Communications. MIT Press, Cambridge (1961)
10. Fung, P.: Compiling Bilingual Lexicon Entries From a non-Parallel English-Chinese Corpus. In: Farwell, D., Gerber, L., Hovy, E. (eds.) Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA 1995), Langhorne, PA, USA, pp. 1–16 (1995)
11. Fung, P.: A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora. In: Farwell, D., Gerber, L., Hovy, E. (eds.) AMTA 1998. LNCS (LNAI), vol. 1529, pp. 1–17. Springer, Heidelberg (1998)
12. Fung, P., Lo, Y.Y.: An ir approach for translating new words from nonparallel, comparable texts. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING 1998), pp. 414–420 (1998)
13. Fung, P., McKeown, K.: Finding Terminology Translations from Non-parallel Corpora. In: Proceedings of the 5th Annual Workshop on Very Large Corpora (VLC 1997), Hong Kong, pp. 192–202 (1997)
14. Gaussier, E., Renders, J.M., Matveeva, I., Goutte, C., Déjean, H.: A Geometric View on Bilingual Lexicon Extraction from Comparable Corpora. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain, pp. 526–533 (2004)

15. Gillard, L., Bellot, P., El-Bèze, M.: D'une compacité positionnelle à une compacité probabiliste pour un système de questions / réponses. In: CORIA, pp. 271–286 (2007)
16. Gillard, L., Sitbon, L., Blaudez, E., Bellot, P., El-Bèze, M.: Relevance Measures for Question Answering, The Lia at qa@clef-2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 440–449. Springer, Heidelberg (2007)
17. Grefenstette, G.: Corpus-Derived First, Second and Third-Order Word Affinities. In: Proceedings of the 6th Congress of the European Association for Lexicography (EURALEX 1994), Amsterdam, The Netherlands, pp. 279–290 (1994)
18. Grefenstette, G.: Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publisher, Boston (1994)
19. Hickl, A., Wang, P., Lehmann, J., Harabagiu, S.M.: Ferret: Interactive question-answering for real-world environments. In: ACL (2006)
20. Huang, Z., Thint, M., Qin, Z.: Question classification using head words and their hypernyms. In: EMNLP, pp. 927–936 (2008)
21. Laroche, A., Langlais, P.: Revisiting Context-based Projection Methods for Term-Translation Spotting in Comparable Corpora. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, pp. 617–625 (2010)
22. Lavenus, K., Grivolla, J., Gillard, L., Bellot, P.: Question-answer matching: Two complementary methods. In: RIAO, pp. 244–259 (2004)
23. Morin, E., Daille, B., Takeuchi, K., Kageura, K.: Bilingual Terminology Mining – Using Brain, not brawn comparable corpora. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), Prague, Czech Republic, pp. 664–671 (2007)
24. Peters, C., Picchi, E.: Cross-language information retrieval: A system for comparable corpus querying. In: Grefenstette, G. (ed.) Cross-Language Information Retrieval, ch.7, pp. 81–90. Kluwer Academic Publishers (1998)
25. Rapp, R.: Identify Word Translations in Non-Parallel Texts. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL 1995), Boston, MA, USA, pp. 320–322 (1995)
26. Rapp, R.: Automatic Identification of Word Translations from Unrelated English and German Corpora. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999), College Park, MD, USA, pp. 519–526 (1999)
27. Salton, G., Lesk, M.E.: Computer evaluation of indexing and text processing. Journal of the Association for Computational Machinery 15(1), 8–36 (1968)
28. Voorhees, E.M.: Overview of the trec 2004 question answering track. In: TREC (2004)

# Aligning the Un-Alignable — A Pilot Study Using a Noisy Corpus of Nonstandardized, Semi-parallel Texts

Florian Petran

Ruhr-University Bochum, Linguistics Department, Bochum, Germany  
[petran@linguistics.rub.de](mailto:petran@linguistics.rub.de)

**Abstract.** We present the outline of a robust, precision oriented alignment method that deals with a corpus of comparable texts without standardized spelling or sentence boundary marking. The method identifies comparable sequences over a source and target text using a bilingual dictionary, uses various methods to assign a confidence score, and only keeps the highest scoring sequences. For comparison, a conventional alignment is done with a heuristic sentence splitting beforehand. Both methods are evaluated over transcriptions of two historical documents in different Early New High German dialects, and the method developed is found to outperform the competing one by a great margin.

**Keywords:** word alignment, noisy text processing, semi-parallel corpora.

## 1 Introduction<sup>1</sup>

Word and sentence alignment is largely regarded as a solved problem. Yet the common approaches to this either presuppose sentence splitting and standardized spelling [1], or they only work on completely parallel texts where no parts are deleted or inserted over the source and target texts (e.g. [2]). With texts that have neither of these properties, the consensus seems to be that in such situations, alignment is impossible to do automatically in a generalized way, and has to be done manually. We present an approach to the alignment of semi-parallel texts without reliable sentence breaks and with non-standardized spelling, and compare it to a standard approach using a sentence splitting heuristics. It was tested with historical texts, but there are conceivably comparable situations in Internet communication, especially in forum or newsgroup posts on common topics.

This paper is outlined as follows. As stated above, work with this exact type of scenario is rather scarce, but a few related approaches are described in section 2.

---

<sup>1</sup> We would like to thank the anonymous reviewers for helpful comments. The research reported here was financed by Deutsche Forschungsgemeinschaft (DFG), Grant DI 1557/4-1.

In section 3, we introduce the text corpus used for the experiments, and in section 4 we explain the extraction of the translation dictionary we used for our alignment method. The alignment algorithm itself is described in section 5. For comparative purposes, we also tried a traditional alignment method with a heuristic sentence breaking, described in section 6. Evaluation method and results are detailed in section 7. Finally, section 8 discusses possible and desirable directions for future research.

## 2 Related Work

State of the art sentence breaking algorithms are usually mostly concerned with disambiguation of existing punctuation marks. There is some work on using conditional random fields to determine sentence boundaries in languages usually written without punctuation such as Chinese (e.g. [3]). However, we only have small samples available for each dialect, so it is unlikely we would have enough training data for a machine learning approach. Spelling variations even within a single text, and the abundance of inflectional morphology would be additional obstacles to the collection of training data. The clause breaking we employed is novel as far as we are aware.

As stated in section 1, alignment is largely regarded as a solved problem, so approaches that differ from current state of the art methods are largely found in older literature. Since we currently use cognates for the dictionary, our method is technically related to that of [4], who use the number of cognates found in a sentence to extend length based sentence alignment [5]. But one of the advantages of our method is that it does not presuppose any text segmentation, even though it would likely profit from a paragraph segmentation and alignment. It would furthermore be trivial to make it work with a bilingual dictionary that is not based on cognates, and going beyond the stage where we align similar words would in fact be high up on the list of future work to do (see below).

Other approaches construct a vector of the number of tokens between each occurrence of a token, and then infer an ad-hoc translation dictionary from a comparison of those vectors ([2], [6]). This is similar to our approach in that the dictionary is then used to give possible points of alignment, and that it is also designed to work with noisy texts without punctuation. But even though it is designed to work with only roughly parallel texts, the index differences will not work at all if there are larger parts of text inserted or omitted. Furthermore, the spelling variation present in our texts make it difficult to determine the identity between words in a reliable way.

Probably most similar to our approach is the `char_align` algorithm [7]. It was also conceived to deal with noisy text (OCR documents), and it uses cognates as well. The difference is that it tries to find alignments on the character level by constructing a scatterplot of character correspondences. The plot is then smoothed by signal processing techniques such as low-pass filtering, and a search

heuristics is employed to find the path with the largest average weight. It appears that probably due to recent improvements on sentence boundary detection algorithms, this venue was not much explored afterwards. The paper does not offer a quantitative evaluation of the performance, and in fact it would be difficult to compare to other methods. Finally, a huge problem with this kind of approach is that it does not seem like it would generalize too well over languages that are not related, and do not have too many corresponding cognates. This is something that our approach should be able to do if one were to substitute our cognate based dictionary for a real translation dictionary.

### 3 Corpus

The texts this study is based on are two versions of the medieval religious text *Interrogatio Sancti Anselmi de Passione Domini* (Questions by Saint Anselm on the Lord’s passion). There are at least 40 versions of the text from the Early Modern period in different dialects of Early New High German (ENHG). Even though these texts have the same topic and roughly the same content, they also differ greatly in language use, and there are passages missing or inserted in between the texts. Although there is a Latin version that is believed to be older, the fact that there are completely novel passages inserted in some versions indicates that they are not necessarily translations of the same text, or of each other. The absence of standardized spelling and consistent sentence boundary marking complicates matters even further.

With a length of 8,000 tokens on average, they are just about too long to do alignments and annotation all manually, yet still too short to employ machine learning approaches. The texts come in prose and verse forms, and the prose forms come in long, medium, and short lengths. As already mentioned, sentences and whole passages are missing between the texts. Additionally, the fact that a text is shorter than another one does not necessarily mean that it cannot have passages that may be absent in the longer ones, and frequently the ordering between passages is changed. In sum, even though the texts cover the same topic and indeed tell the same story, they are extremely heterogeneous, and present a very difficult scenario for automatic alignment.

For the experiments described below, we used two prose versions of the story that originated in the broader Bavarian region with slightly different dialectal background. The first is a transcription of a manuscript written in the 14th century.<sup>2</sup> With about 10,000 tokens it is one of the longest versions of the story; this will be our source text. The second one is a transcription of manuscript written in the late 15th century.<sup>3</sup> It is a medium length version with about 5,900 tokens; this will be the target text for our alignments.<sup>4</sup> Fig. 1 shows the first proper sentence of each text.

<sup>2</sup> Clm. 23371, fol. 126v – 138v from the Bavarian State Library in Munich.

<sup>3</sup> Lit. 176 Ed. VII, fol. 13v – 58v from the State Library Bamberg.

<sup>4</sup> Our versions also contained a low amount of noise where transcribers did not properly follow markup conventions. This is not unsimilar to what might be the noisy output of a web crawler or the result of an OCR.

**Source:**

*Sand Anhelm der pat vner vrowē vō himelreich lange zeit. mit vaten vñ mit wachen Vnd mit andechtigem gepet.*

“Saint Anselm, he begged our lady of the heavens for a long time, fasting, waking, and with devout prayers”

**Target:**

*Ain hoher lerer hiez anhelmus der pat vnerfrauen lange weill vnd zeit wainent vaten vnd peten.*

“A high teacher called Anselm, he begged our lady for a long time, crying, fasting, and praying”

**Fig. 1.** The beginning of source and target texts respectively. Capitalization and punctuation are the same as in the original.

## 4 Dictionary

The alignment method is based on a translation dictionary. There is no such preexisting dictionary, but since the dialects are very similar, we were able to automatically extract a list of cognates<sup>5</sup> using the BI-SIM measure [8]. BI-SIM returns a similarity value between 0 and 1, where 1 stands for an identical form, and 0 stands for two completely distinct strings.

BI-SIM has been successfully used to extract seed dictionaries for Slovenian and Croatian [9], with a similarity cutoff of 0.7. For our experiments, we set that cutoff to 0.8, based on the intuition that ENHG dialects are related more closely than Croatian and Slovenian. This was confirmed by calculating the average BI-SIM value of a small sample of cognates extracted from the texts, and also empirically determined to work better with the experiments than a lower cutoff value. To mitigate the number of false positives in the dictionary, we further excluded words with only three characters from the cognate extraction, unless they were identical. The texts are from a restricted domain, dealing with religious topics. As a consequence, we made the simplifying assumption that similar words largely have a similar meaning, and did no additional verification of the entries.

We cannot easily stem the words due to the lack of standard orthography. Hence the dictionary contains inflected forms. Since these forms may occur in different syntactic contexts and, moreover, the texts sometimes use slightly different inflectional paradigms, not all inflectional variants of a word may be recognized as translations of the same word. If the noise stays below a certain threshold, the alignment method should be able to discard the wrong translations at a later stage because they will not form sequences with other token pairs.

Table 1 shows some sample entries from the dictionary. Lines 1–3 and 5 illustrate correct mappings of non-identical forms. Lines 4 and 6–7 show false positive entries, and lines 8 and 9 illustrate ambiguous mappings. Line 8 shows a mapping of different inflectional variants of the possessive pronoun *seine* “his.” It is not strictly correct in the lexicographic sense, since the case differs, but it

<sup>5</sup> We use the term *cognates* to refer to words with a similar form and similar meaning. They do not need to have common ancestors, as in the linguistically strict sense.

**Table 1.** Sample entries from the dictionary

	Source		Target	
1	<i>auzsetzigen</i>	“leper”	<i>aussetziger</i>	“leper”
2	<i>enphangen</i>	“receive”	<i>enpfangen</i>	“receive”
3	<i>ewangelist</i>	“Evangelist”	<i>eubangelist</i>	“Evangelist”
4 *	<i>gewant</i>	“garment”	<i>gewalt</i>	“violence”
5	<i>grozz</i>	“great/tall”	<i>grosz</i>	“great/tall”
6 *	<i>land</i>	“land”	<i>lang</i>	“long”
7 *	<i>leit</i>	“suffering”	<i>leib</i>	“body”
8 [*]	<i>seine</i>	“his” NOM PL	<i>seinem</i>	“his” DAT SG
	[*] <i>seine</i>		<i>seinen</i>	“his” ACC SG
9	<i>weib</i>	“woman”	<i>weip</i>	“woman”
	* <i>weib</i>		<i>wein</i>	“wine”
	* <i>weib</i>		<i>weil</i>	“because”

captures the sense in a way that can be used for alignments. In line 9, the first mapping is correct, while the other two are false positives. Overall, the dictionary covers about 69% of the types in the longer source text S, and about 99% of the types in the shorter text T.

## 5 Alignment Method

The basic idea of the alignment method is to find the longest non-conflicting, corresponding sequence of translation candidates in both texts. In this section, we explain the procedure used to find those sequences in detail.

Be  $s_i$  a token  $s \in S$  at position  $i$ , and  $t_j$  a token  $t \in T$  at position  $j$ . As a first step, we collect all alignment candidates for each  $s$ . That is, for each  $s$ , we retrieve its translation from the dictionary and record their occurrences (indices) in T as alignment candidates. This way, 56.2% of the tokens in S are assigned at least one candidate, with an overall average of 28.2 candidates per source token.

In the next step, we merge candidates to bigram sequences if both their source and target components are *close* to each other. The *proximity condition*  $C(r_i, r_j)$  states that the difference between their indices may not exceed 2.

$$C(r_i, r_j) = \begin{cases} 1, & \text{if } j - i < 3 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

So for each pair  $s_i : t_j$  we check if the token at  $s_{i+1}$  has a translation candidate  $t_k$  that is close to  $t_j$ . If it does, we merge both translation pairs to a bigram sequence. Then we remove them from the set of data still to be treated, and continue with the next pair that is not yet part of any sequence.

Often, there are tokens that cannot be aligned to a counterpart in the other text. There may be punctuation marks that are absent in the other text; some tokens are missing from the dictionary, or some may not have a counterpart in the other text at all. Hence, we allow the algorithm to skip a single token in S

when looking for the next component of a bigram sequence, instead of moving onward token by token.<sup>6</sup> The amount of candidates in S that may be skipped as well as the maximum difference of the indices still allowed to satisfy the proximity condition were empirically determined by investigating a small sample. Note that the treatment of S and T is asymmetrical here. This is possible because the same process is later applied in the other direction as well.

<b>Gloss</b>	then	went	Judas	Iscariot		to	the	prince	of	the	jews
<b>Source</b>	<i>Do</i>	<i>giench</i>	<i>Iudas</i>	<i>scarioth</i>	.	<i>zve</i>	<i>den</i>	<i>fursten</i>	<i>der</i>	<i>Iuden</i>	.
<b>Target</b>	<i>Do</i>	<i>gieng</i>	<i>iudas</i>	<i>zu</i>	<i>den</i>	<i>iuden</i>					
<b>Gloss</b>	then	went	Judas	to	the	Jews					

**Fig. 2.** Example of a correctly aligned sequence

Fig. 2 shows a sequence of correct alignment pairs. For this example, the algorithm previously arrived at *Do* “then” which has 137 translation candidates in T, and finds *giench* “went” as the next token, which has 8 translation candidates. Among the  $137 \cdot 8$  possible pairings, there are only two alignments where the translation candidates occur in proximity of each other, so that we merge these pairings to a bigram. For other bigrams in S, there are multiple instances of translation candidates occurring close to each other in T, which means we would have multiple competing bigram sequences at that position. Overall, at each position where some bigram starts there are 3.72 competing sequences on average. The combination of candidates results in 7,557 bigram pairs starting at 2,034 different positions. As stated above, 56.2% of the tokens in S are assigned one or more translation candidates. Among those, 35.72% are at the beginning of at least one bigram sequence after this step.

Now follows the sequence expansion step. Here, we try to expand the sequences at their tail end by adding alignment pairs that are not yet part of any other sequence. We also relax the conditions for proximity slightly: while in the bigram merging step, we allow a one-token skip in S, the skip may now also occur on the target side, in addition to the maximum allowable index difference from the proximity condition. That means the difference between the indices may be three if one of the tokens does not have a candidate assigned to them, and two otherwise. The reasoning behind this is that we first try to find a close bigram pair to anchor the alignment to, and then gradually expand outwards from it.

In the example given in Fig. 2, we find the next token in S to be *Iudas* “Judas,” which aligns well with its counterpart in T. The following token *scarioth* “Iscariot” is entirely absent from T, as is the full stop after that. Then follows *zve*

<sup>6</sup> The skipping of candidates when looking for similar n-grams in different texts is also successfully employed in the ROUGE-S metrics [10].

– *zu* “to,” which should be aligned. In this case, the translation is missing from the dictionary because we excluded words with three characters or less from cognate extraction process (see section 4). Because we relaxed the proximity criteria, as just described, we are now able to add the following token *den* and its counterpart in T to the alignment sequence. It may appear as if the maximum allowable index difference from the proximity condition was exceeded in this case. However, *scarioth* has no translation candidate assigned to it, and hence, can be skipped and does not count towards the maximum.

Each pair that has been added to a sequence is removed from the collection of candidates. We iterate through the sequences until we have made a full round without adding any more pairs. Then the process is repeated for the reverse direction, from T to S to account for the asymmetrical treatment in the first part of the process.

In our texts, the sequence expansion step results in a collection of sequences of up to 4 token pairs, with an average length of 2.1. The fact that the average only slightly exceeds the minimum sequence length indicates that the majority of sequences did not get expanded in this way, since they are chance co-occurrences by chance of two token pairs. They will get discarded at a later stage when preferable matches for their participating tokens have been found.

The next step concerns sequence merging. Since the algorithm only add pairs to the tail ends of the sequences, and the proximity criteria is laxer than in forming the original sequences, it is possible that the expansion has brought the end of a sequence so close to the beginning of another one that they can be merged to one. We iterate through all the sequences at each positions and apply the proximity criteria to the end of this sequence and the next position where a sequence starts.

For the example in Fig. 2, we do not find any such sequence, but consider the correctly aligned sequence in Fig. 3. In the bigram merging step, we found bigrams starting at *gewalt* and at *stat*. The subsequent sequence expansion step has created a situation in which the tail end of the first sequence, *in*, is close to the beginning of the second one. Consequently, both are merged to one long sequence in the sequence merging step. This increases the maximum sequence length to 10, while the average remains at 2.1. Again, this indicates that the majority of sequences are neither expanded nor merged.

The final step is score assignment, where a confidence score is determined for each sequence. This score is based on the length of the sequence, the average difference between the indices in S and T, and the average BI-SIM value of the aligned words. The employment of the BI-SIM measure to grade the alignments again relies on the assumption that the languages are related, as did the dictionary extraction method discussed above. But since this measure did not contribute as much to the quality of the results as the other two components, it might be possible to replace or omit it should the alignment method be employed with unrelated languages.

Be  $r$  a sequence of aligned indexes in T and S respectively, with a length of  $n$  and with  $r_{i,1}$  denoting the source index part of alignment pair  $i$ , and  $r_{i,2}$  the

<b>Gloss</b>	O	violence	is	in	this	city	today	happened
<b>Source</b>	<i>Owe</i>	<i>gewalt</i>	<i>ist</i>	<i>in</i>	<i>dirre</i>	<i>stat</i>	<i>heut</i>	<i>geschechen</i>
<b>Target</b>	<i>Aube</i>	<i>wie</i>	<i>grosser</i>	<i>gewalt</i>	<i>ist</i>	<i>in</i>	<i>diser</i>	<i>stat</i>
<b>Gloss</b>	O	how	great	violence	is	in	this	city

**Fig. 3.** Example of two sequences that have been merged

target index part of that same alignment pair. The BI-SIM measure can then be defined as follows in equation 2.

$$b = \frac{\sum_{i=1}^n \text{bi\_sim}(s_{r_{i,1}}, t_{r_{i,2}})}{n} \quad (2)$$

The index difference measure is defined in the following equation 3. As above,  $s_i$  and  $t_i$  are tokens at index position  $i$  from  $s$  and  $t$  respectively, while  $|S|$  and  $|T|$  are the total amount of tokens in the respective text. Note that a lower index difference is desirable here, so we subtract from 1 to invert the values.

$$d = 1 - \left( \frac{\sum_{i=1}^n \left| \frac{r_{i,1}}{|S|} - \frac{r_{i,2}}{|T|} \right|}{n} \right) \quad (3)$$

Now be  $B$  the set of  $b$  values,  $D$  the set of all  $d$  values, and  $N$  the set of all lengths of sequences for all  $r \in R$ . In order to give all measures of confidence the same weight, every one is normalized by the maximum value over all sequences. Then we take the mean value.

$$c = \frac{1}{3} \left( \frac{b}{\max B} + \frac{d}{\max D} + \frac{n}{\max N} \right) \quad (4)$$

Other confidence measures to the score we considered include the amount of alternate sequences starting at the same position, and the average length difference between the words, and the average difference in relative frequency. These were tested but ultimately not employed, since the first two had a detrimental effect on the quality of the results, and the latter did not make any difference at all.

After the scores are assigned, lower-scored sequences are discarded. That is, if any token of a sequence (from S or T) is simultaneously a member of a higher scoring sequence, the lower scoring one is discarded. Since we aim for a high-precision result rather than for a strong recall, we also discard all ambiguous sequences. Those are sequences that conflict with others, but a decision could not be made for either one because they have the same score. As a final result, this method gives us 1,280 1:1 alignment pairs. That is a coverage of 22.92% of the shorter text T.

## 6 Alternate Alignment

For the reasons outlined in section 1, a conventional alignment is difficult to accomplish. The main reason was that punctuation marks cannot be used as segment boundaries. To apply conventional alignment tools, we therefore have to come up with some segmentation for the text. Since statistical alignment methods do not usually use linguistic knowledge, we assumed that it is not necessarily required that those units be sentences.

We used a list of possible spelling alternates of frequent conjunctions such as *und* “and” or *oder* “or” to split the text into segments that might resemble clauses. This will enable us to align those segments, and afterwards align the words within them. We did not extract the conjunction alternates from the text, but had them supplied by a historical linguist based on their intuition of what possible alternates could occur. An additional presupposition for the segmentation step was that a sentence was not allowed to have only one word, since conjunctions frequently occurred directly next to each other.

However, the results of the segmentation step already seems problematic. For one, clauses or sentences do not always start with a conjunction, so the segments frequently crossed clause boundaries. Furthermore, ENHG texts make excessive use of binomial pairs of synonymous or partly synonymous words (“Zwillingsformel”) joined with a conjunction to express a single concept, a stylistic device borrowed from classical Latin. For example, Jesus is described as *gefangen und gebunden* “caught and bound,” and his disciples upon hearing this as *schreiend und weinend* “crying and weeping.” Using our segmentation heuristics, those would then be taken as two clause segments of their own, even when, grammatically speaking, they should be part of one clause. According to a tentative look at the output of this step, the latter problem seemed to make up the majority of the mis-segmentations, indicating that this step might well work better with a modern language text.

The results of the segmentation step were sentence aligned using the Gargantua toolkit [11]. It is basically an extension to other work in sentence alignment that combines length based alignment with multiple iterations over translation model based alignment [12], but it handles M:N alignments with  $N, M > 2$ . Due to the way the results of our segmentation step turned out (see section 7), this was deemed highly desirable. Word alignment within the aligned segments was then done using GIZA++ [1]. Alignments involving the NULL token and N:1/1:N alignments had to be specially considered for the output. For the former, our method produces no output at all if it does not find an alignment, as does Gargantua, but GIZA explicitly aligns it to NULL. Those were accordingly not considered in the evaluation, although further manual alignments seem to indicate that they may constitute up to about a third of the alignments. Our method does not yet support multi-token alignments, so to compare the results, they had to be converted into sequences of 1:1 alignments. For example,  $s_i : ( t_j t_k )$  became  $s_i : t_j; s_i : t_k$ . After all that, the method produced about 4,979 unique pairs, which amounts to a coverage of 84.62% of the shorter text.

## 7 Evaluation and Conclusion

Since the method is very much a work in progress on an ongoing project, there is no complete gold standard so far with these two texts, which complicates evaluation. Instead, we evaluated recall on a subset of 2,500 pairs that have been aligned by one annotator. The evaluation of precision, however, is impeded, since the subset does not cover all of the tokens in the text. Running the alignments on just a subset of the texts is not an option either, since the ordering of sequences is heavily changed between the texts, so it is not easily possible to evaluate on the first  $n$  tokens of both. All tokens that were not yet annotated are counted as errors in the results presented below, even if the alignments produced by the two algorithms may in fact be correct.

Two properties of our alignment method further complicate the matter. First, we currently only output 1:1 pairs, whereas a lot of N:M alignments seem to occur in the part that is already annotated. For the evaluation, we converted those into sequences of N:M 1:1 alignments, as described in section 6. But even if our system's output for these is partly correct, this would decrease the count of correct results. Since our algorithm only gives one alignment for each token, for every 1:1 pair that is correctly produced, there would be N-1 pairs that can definitely not be in the output. Alternatively, we could exclude all N:M alignments from the evaluation until our system is able to handle such cases. Second, NULL alignments occur quite frequently, as should be expected with semi-parallel texts. We could count every token where we did not produce an alignment as an alignment with the NULL token if we assumed that a NULL alignment was the default case. This would increase recall at the expense of precision. If we did not include NULL alignments at all in our method, this would increase precision, but at the expense of recall, since we would not cover all of the tokens in the text.

**Table 2.** Evaluation of our method

NULL	N:M	precision	recall	F-measure
-	+	42.2%	23.4%	30.1
+	+	22.1%	50.7%	30.8
-	-	42.2%	25.7%	31.9
+	-	22.1%	55.5%	31.6

Table 2 shows precision, recall, and balanced F-score for all four possible cases. The F-measure remains more or less constant as we trade off between precision and recall, as should be expected. As just explained, we suspect that the actual precision may be higher than our partial gold standard accounts for, so we had our annotator manually examine the non-NUL pairs our method produced. It was found that 51.7% of these were actually correct ones, which puts the actual precision we can achieve higher than reported in the table, even though this value is not exactly comparable.

Evaluating the output of the traditional method is simpler, since it does account for both NULL and N:M alignments. Its output was found to contain only two of the correct pairs in our incomplete gold standard, which amounts to a precision of 0.04% and recall of 0.08%. It should be noted, however, that all pairs that were not covered by the (partial) gold standard are again counted as wrong answers, so actual performance may be higher. On the other hand, we already know that those responses involving a token our method aligned correctly are wrong if they differ. In addition, we manually evaluated 540 pairs involving a token from T or S that had been incorrectly aligned by our system. After all, it is possible that the traditional method was right in those cases where our method guessed wrong. Of all those, not a single one was correct. This means that we have 23% of the output of the traditional method we know are incorrect.

All in all, according to the cursory evaluation outlined here, this method seems to perform far worse than the algorithm proposed in this paper. Since about 77% of the results of the traditional method were still not checked at all, it is not technically impossible that it successfully produced correct alignments where our method did not find anything. Judging from the quality of the evaluated alignments, however, that is highly unlikely, to say the least. Further evaluation will help clarify all these points.

A tentative qualitative error analysis of both methods seems to indicate that the errors of our method comprise mostly sequences of function words, and that it might benefit from a list of stop words to disregard in the alignments. The traditional method is difficult to analyze since there are few sensible alignments in the final output. Based on the problems with the segmentation step outlined in section 6, the sentence aligner did not seem to produce a lot of sensible output, either, which, of course, posed problems for the word aligner. It seems that it did not handle the large omissions between the texts very well. The overall method seems to suffer greatly from the haphazard way of sentence splitting, and would likely benefit greatly from improvements upon that.

In conclusion, we have shown that it is possible to word-align only roughly comparable texts. Since the environment the method was developed for does not have standardized spelling, or punctuation, we do not make use of such clues, and accordingly do not rely on pre-aligned larger beads to accomplish the final word alignment. The only resource our method makes use of is a translation dictionary. In our case, this is extracted using cognates - hence in its current state, the method is only applicable to closely related languages. It might be extended to unrelated languages by employing a translation dictionary from another source, or by employing a different way of extracting the dictionary. Even though the method is still in its infancy, it already outperforms conventional tools in this setting by a great margin. It succeeds in delivering alignments in a very difficult environment, and this is a success we hope to further improve upon.

## 8 Future Work

Future work would of course first and foremost include finishing the creation of a gold standard covering the whole document to evaluate more thoroughly against.

This concerns especially the evaluation of the traditional method. Regarding the evaluation, another venue that should be explored is the performance of the `char_align` algorithm [7] for our particular problem.

Regarding the algorithm itself, the most pressing issues are to have it output more alignments, aligning tokens that are not found in the dictionary, and handling alignment patterns other than 1:1. In its current state, the method does omit a lot of the output it produces because it cannot decide between the sequences with our scoring method. Improvements on the assignment of confidence scores could help to improve the coverage of our system. The alignment of tokens not covered by the dictionary is something that could be handled based on cognates, such as alignment of the closest match, or possibly word length, although it would be prudent to limit those to a window around established sequences.

Employing a stemmer could theoretically help, but since those mostly work with a list of possible affixes, it would be difficult to do. As for the alignment patterns, N:1/1:N and even N:M alignments did form a considerable part of the manually annotated data, since expressions are often paraphrased between the texts. So this is a crucial issue, but so far, not one where an obvious solution presents itself. Handling of alignments involving the `NULL` token is connected to that, since a `NULL` token alignment means that something should not be added to a multi-alignment. As stated above, these appear to be fairly frequent as well.

Yet a different option to explore is if and how our method could benefit from a combination with the traditional, or other methods. Since the sequences provide a kind of text segmentation that could be similar to paragraphs, the traditional method might produce better results if it were to be combined somehow, and the output of the traditional method could be used to enhance the results of our method. An open question in this regard is whether the texts are at all long enough to train a translation model on them.

A final option we want to explore comes from the specific scenario we work in. Since we have about 40 different versions of the text that are all supposed to be aligned to each other, we could try to use the amount of text to our advantage by exploiting alignment transitivity [13]. This means that if  $a_i$  aligns to  $c_j$ , and  $c_j$  to  $b_k$ , then we can assume that  $a_i$  aligns to  $b_k$ . It could conceivably contribute greatly to the overall coverage of the results. Since we would need multiple passes over various texts for this, improving the performance of the algorithm would also be an issue.

## References

1. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29, 19–51 (2003)
2. Fung, P., Church, K.W.: K-vec: a new approach for aligning parallel texts. In: *Proceedings of the 15th Conference on Computational Linguistics*, vol. 2, pp. 1096–1102. Association for Computational Linguistics (1994)
3. Huang, H., Chen, H.: Pause and Stop Labeling for Chinese Sentence Boundary Detection. In: *Proceedings of Recent Advances in Natural Language Processing*, pp. 146–153 (2011)

4. Simard, M., Foster, G.F., Isabelle, P.: Using cognates to align sentences in bilingual corpora. In: Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing, vol. 2, pp. 1071–1082. IBM Press (1993)
5. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. *Computational Linguistics* 19, 75–102 (1993)
6. Fung, P., McKeown, K.: Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. In: Proceedings of the Association for Machine Translation in the Americas (AMTA 1994), pp. 81–88 (1994)
7. Church, K.W.: *char\_align*: a program for aligning parallel texts at the character level. In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics, pp. 1–8. Association for Computational Linguistics (1993)
8. Kondrak, G., Dorr, B.: Identification of confusable drug names: A new approach and evaluation methodology. In: Proceedings of the 20th International Conference on Computational Linguistics, pp. 952–958. Association for Computational Linguistics (2004)
9. Ljubešić, N., Fišer, D.: Bootstrapping Bilingual Lexicons from Comparable Corpora for Closely Related Languages. In: Habernal, I., Matoušek, V. (eds.) *TSD 2011. LNCS*, vol. 6836, pp. 91–98. Springer, Heidelberg (2011)
10. Lin, C.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), pp. 25–26 (2004)
11. Braune, F., Fraser, A.: Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora. In: *Coling 2010: Poster Volumes*, pp. 81–89 (2010)
12. Moore, R.: Fast and Accurate Sentence Alignment of Bilingual Corpora. In: Richardson, S.D. (ed.) *AMTA 2002. LNCS (LNAI)*, vol. 2499, pp. 135–144. Springer, Heidelberg (2002)
13. Simard, M.: Text-Translation Alignment: Three Languages Are Better Than Two. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 2–11. Association for Computational Linguistics (1999)

# Parallel Corpora for WordNet Construction: Machine Translation vs. Automatic Sense Tagging<sup>\*</sup>

Antoni Oliver and Salvador Climent

Universitat Oberta de Catalunya, Barcelona, Spain  
{aoliverg, sclement}@uoc.edu  
www.uoc.edu

**Abstract.** In this paper we present a methodology for WordNet construction based on the exploitation of parallel corpora with semantic annotation of the English source text. We are using this methodology for the enlargement of the Spanish and Catalan versions of WordNet 3.0, but the methodology can also be used for other languages. As big parallel corpora with semantic annotation are not usually available, we explore two strategies to overcome this problem: to use monolingual sense tagged corpora and machine translation, on the one hand; and to use parallel corpora and automatic sense tagging on the source text, on the other.

With these resources, the problem of acquiring a WordNet from parallel corpora can be seen as a word alignment task. Fortunately, this task is well known, and some aligning algorithms are freely available.

**Keywords:** lexical resources, wordnet, parallel corpora, machine translation, automatic sense tagging.

## 1 Introduction

WordNet [7] is a lexical database that has become a standard resource in Natural Language Processing research and applications. In WordNet nouns, verbs, adjectives and adverbs are organised in sets of synonyms, the so called synsets. These synsets are connected to other synsets by semantic relations (hiponymy, antonymy, meronymy, troponomy, etc.). For instance, in WordNet 3.0, the synset identified by the offset and pos 06171040-n has two *variants*: *linguistics* and *philology*. Each synset has a *gloss* or *definition*, for the synset of the example being: *the humanistic study of language and literature*. It also has a hypernym 06153846-n (*humanistic\_discipline*, *humanities*); and two hyponyms 06171265-n (*dialectology*) and 06178812-n (*lexicology*).

The English WordNet (PWN - *Princeton WordNet*) is being updated regularly, so that its number of synsets increases with every new version. The current version of PWN is 3.1, but we are using in our experiments the 3.0 version.

---

\* This research has been carried out thanks to the Project MICINN, TIN2009-14715-C04-04 of the Spanish Ministry of Science and Innovation.

WordNet versions in other languages are also available: in the EuroWordNet project [26] WordNet versions in Dutch, Italian and Spanish have been developed; the Balkanet project [24] developed WordNets for Bulgarian, Greek, Romanian, Serbian and Turkish; and RusNet [2] for Russian, among others. On the Global WordNet Association<sup>1</sup> website a comprehensive list of WordNets available for different languages can be found.

According to [26], we can distinguish two general methodologies for WordNet construction: (i) the *merge model*, in which a new ontology is constructed for the target language and relations between PWN and this local WordNet are generated; and (ii) the *expand model*, in which English variants associated with PWN synsets are translated following several strategies. In this work and for our purposes we are following this second strategy.

The PWN is a free resource available at the University of Princeton website<sup>2</sup>. Many of the available WordNets for languages other than English are subject to proprietary licenses, although some others are available under free license, for example: Catalan [3], Danish [19], French WOLF WordNet [21], Hindi [23], Japanese [10], Russian [2] or Tamil [20] WordNets among others. The goal of this project is to enlarge and improve the Spanish and Catalan versions of WordNet 3.0 and distribute them under free license.

## 2 Use of Parallel Corpora for the Construction of WordNets

There are several works using parallel corpora for tasks related to WordNet or WordNet-like ontologies. In [11], an approach for acquiring a set of synsets from parallel corpora is outlined. Such synsets are derived by comparing aligned words in parallel corpora in several languages. If a given word in a given language is translated by more than one word in several other languages, this probably means that the given word has more than one sense. This assumption also works the other way around. If two words in a given language are translated by only one word in several other languages, this probably means that the two words share the same meaning. A similar idea along with a practical implementation is found in [9], and their results show that senses derived by this approach are at least as reliable as those made by human annotators.

In [8], the Slovene WordNet is constructed using a multilingual corpus, a word alignment algorithm and existing WordNets for some other languages. With the aligned multilingual dictionary, all synsets of the available WordNets are assigned. Of course, some of the words in some of the languages are polysemic, so that more than one synset is assigned. In some of these cases, a word can be monosemic at least in one language, with a unique synset assigned. This synset is used to disambiguate and assign a unique synset in all languages, including Slovene. A very similar methodology is used for French in [21], along with other methods based on bilingual resources.

<sup>1</sup> <http://www.globalwordnet.org>

<sup>2</sup> <http://wordnet.princeton.edu>

The construction of an Arabic WordNet using an English-Arabic parallel corpus and the PWN is depicted in [6]. In this parallel corpus the English content words were annotated with PWN synsets.

### 3 Use of Machine Translation for the Construction of WordNets

Two projects related to WordNet using machine translation systems can be mentioned: the construction of the Macedonian WordNet and the Babelnet project.

In the construction of the Macedonian version of WordNet [22], the monosemic entries are directly assigned using a bilingual English-Macedonian dictionary. For polysemic entries the task can be seen as a Word Sense Disambiguation problem, and thus be solved using a big monolingual corpus and the definitions from a dictionary. However, none of these resources was available. To get Macedonian definitions, PWN glosses were automatically translated into Macedonian using Google Translate. Instead of using a corpus, they took the web as a corpus through the *Google Similarity Distance* [5].

The Babelnet project [15] aims to create a big semantic network by linking the lexicographic knowledge from WordNet to the encyclopedic knowledge of Wikipedia. This is done by assigning WordNet synsets to Wikipedia entries, and making these relations multilingual through the interlingual relations in Wikipedia. For those languages lacking the corresponding Wikipedia entry, the authors propose the use of Google Translate to translate a set of English sentences containing the synset in the Semcor corpus and in sentences from Wikipedia containing a link to the English Wikipedia version. After that, the most frequent translation is detected and included as a variant for the synset in the given language.

In [16], some preliminary experiments on WordNet construction from English machine translated sense tagged corpora are presented. In this paper, this task is presented as a word alignment problem, and some very basic algorithms are evaluated. In [17], these basic algorithms are compared with the Berkeley Aligner for the same task. These papers show that the methodology proposed is promising to build WordNets from scratch, as well as to enlarge and improve existing WordNets.

## 4 Our Approach

### 4.1 Goal

In this paper we present two approaches for the construction of WordNets based on sense tagged parallel corpora from English to the target language (in our case Spanish). The English part of the corpus must be annotated with PWN synsets. The target part of the corpus does not need to be annotated. To our knowledge, there is no such a corpus freely available for the languages of interest. There

are some English sense tagged corpora available, as well as some English and Spanish parallel corpora.

With the available resources, we get a parallel corpora with the English part tagged with PWN synsets in two ways:

- Automatically translating the available English sense tagged corpora into Spanish and Catalan
- Automatically tagging with PWN senses the available English-Spanish parallel corpora

With such a parallel corpus available, the task of constructing a target WordNet can be reduced to a word-alignment task. The relations between the synset in the target WordNet are copied from PWN, assuming that the relations are linguistically and culturally independent from each other.

## 4.2 Corpora

**Sense Tagged Corpora.** We have used two freely available sense tagged corpora for English, the tags being the PWN 3.0 synsets:

- The Semcor corpus<sup>3</sup> [14].
- The Princeton WordNet Gloss Corpus (PWGC)<sup>4</sup>, consisting of the WordNet 3.0 glosses semantically annotated.

In table 1 we observe the total number of sentences and words in the corpus.

**Table 1.** Size of the sense tagged corpora

Corpus	Sentences	Words
Semcor	37.176	721.622
PWGC	117.659	1.174.666
Total	154.835	1.896.288

**Parallel Corpora.** We have used several subsets of the European Parliament Proceedings Parallel Corpus<sup>5</sup> [12] consisting in the first 200K, 500K and 1M sentences of the corpus. In table 2 we can observe the number of sentences and words of these subsets and of the full corpus.

<sup>3</sup> <http://www.cse.unt.edu/~rada/downloads.html>

<sup>4</sup> <http://wordnet.princeton.edu/glosstag.shtml>

<sup>5</sup> <http://www.statmt.org/europarl/>

**Table 2.** Size of the Europarl corpus

Corpus	Sentences	Words-eng	Words-spa
Full	1.786.594	44.652.439	46.763.624
200K subset	200.000	5.415.925	5.659.496
500K subset	500.000	13.611.548	14.208.128
1M subset	1.000.000	26.830.587	28.121.665

### 4.3 Machine Translation

For our experiments we need a machine translation system able to perform good lexical selection, that is, to select the correct target words for the source English sentence. In case of ambiguous words, the system must be able to disambiguate it and choose the correct translation. In our study, other translation errors are less important. Therefore we used a statistical machine translation system: Google Translate<sup>6</sup>. In previous works [16] and [17] we also used Microsoft Bing Translator<sup>7</sup> obtaining very similar results.

We did not assess in deep the ability of the system to do a correct lexical selection, but we performed some successful tests. Consider the English word *bank*. According to PWN, it has 10 meanings as a noun, but we will concentrate on only two of them: 09213565n (*sloping land (especially the slope beside a body of water)*) and 08420278n (*a financial institution that accepts deposits and channels the money into lending activities*). The first meaning has three possible variants in Spanish (*margen, orilla, vera*), according to the preliminary version of the Spanish 3.0; whereas the second meaning has only one Spanish variant (*banco*). If we take sentence correspondings to these senses and we translate them with the given MT systems we get:

She waits on the bank of the river. Ella espera en la orilla del río.  
 She puts money into the bank. Ella pone el dinero en el banco.

As we can see, the systems does, at least in certain situations, a good lexical selection. Few references on figures about lexical selection precision for Google Translate can be found in the literature. In [25], a *position-independent word error rate* (PER) of 29.24% is reported for Dutch-English. In [4], a PER of 28.7% is reported for Icelandic-English.

### 4.4 Automatic Sense Tagging

For the semantic annotation of the parallel corpora we use Freeling [18]. This linguistic analyser has recently added the UKB algorithm for sense disambiguation, and it is able to tag English texts with PWN 3.0 senses. As we have an English corpus manually tagged with PWN 3.0 senses, we can perform an evaluation of the automatic tagging task. Hence, we have automatically tagged the sense

<sup>6</sup> <http://translate.google.com>

<sup>7</sup> <http://www.microsofttranslator.com/>

tagged corpus and we have compared each tag with the corresponding one in the manually tagged version of the corpus. In this experiment we got an overall precision of 73.7%.

#### 4.5 Word Alignment Algorithms

Once we have a parallel corpus sense tagged English - Target Language, the task of deriving the local WordNet can be viewed as a word alignment problem. We need an algorithm capable to select from the following corpus...

**English:**

Then he noticed that the dry wood of the wheels had swollen.

**Sense Tagged English:**

00117620r he 02154508v that the 02551380a 15098161n of the 04574999n had 00256507v .

**Spanish Translation:**

Entonces se dio cuenta de que la madera seca de las ruedas se había hinchado.

...the following set of relations:

00117620r - entonces	02154508v - darse cuenta
02551380a - seco	15098161n - madera

Fortunately, word alignment is a well-known task and there are several algorithms available to solve it. In this project we use the Berkeley Aligner<sup>8</sup> [13]. This freely available algorithm performs the alignment task and gives a probability score for each word alignment.

At this stage, we work with the Berkeley Aligner assuming two restrictions: (i) we only detect as a variant for a given synset simple lexical units, that is, no multiwords; and (ii) we only detect one variant for each synset. In a future work we will try to overcome such restrictions. We are using the Berkeley aligner with a combination of MODEL 1 and HMM models with 5 iterations for each model.

### 5 Evaluation

In this section we present the results of the evaluation of our experiments. Firstly, we present the results for the experiments using machine translation of sense disambiguated corpora. Secondly, we present the evaluation for the experiments using automatic sense tagging of parallel corpora. At the end of this section we present a comparison of the results obtained by each of the two methods.

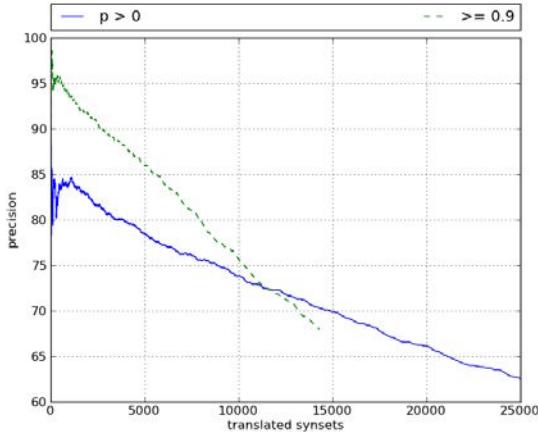
The evaluation has been carried out automatically using the preliminary version of the Spanish 3.0 WordNet. This evaluation method has a major drawback: since the WordNet of reference is not complete, some correct proposals can be evaluated as incorrect.

The evaluation is performed in an accumulative way, starting with the most frequent synset in the corpus. Results are presented in graphics where the *y* values represent the accumulate precision and the *x* values represent the number of extracted synsets.

<sup>8</sup> <http://code.google.com/p/berkeleyaligner/>

### 5.1 Machine Translation of Sense Tagged Corpora

In figure 1 we observe the results of the machine translated sense tagged corpus, as well as the evaluation for all alignments and the evaluation for the subsets of alignments with a probability of 0.9 or higher.



**Fig. 1.** Precision Berkeley Aligner for the machine translated sense tagged corpus

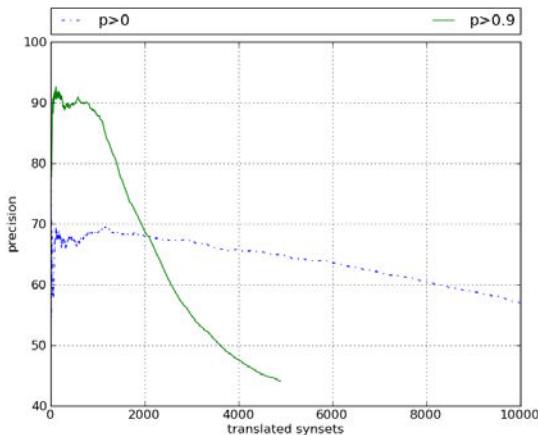
With this setting we obtain a variant for 3.880 synsets with a precision of 80% or higher and 8.866 with 75%. If we take only the alignments with a probability of 0.9 or higher these figures improve, and we obtain one variant for 7.996 synsets with 80% of precision and 10.306 with a precision of 75%.

### 5.2 Automatic Tagging of Parallel Corpora

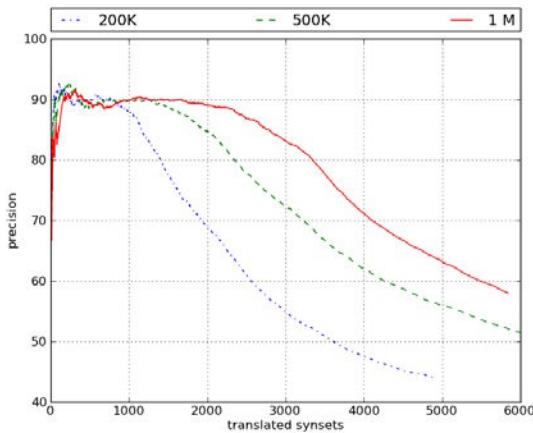
In figure 2 we observe the results for the 200K subset of sentences of the Europarl corpus with automatic sense tagging of the English part. Please, note the change of scale when comparing it with figure 1.

In this experiment we obtain poorer results in comparison with results presented in 5.1. If we take into account all the alignments we can not obtain any variant with a precision higher than 75% (in fact, we do not obtain any variant with a precision higher than 70%). If we concentrate on alignments with a precision of 0.9 or higher, we obtain 1.360 variants with a precision of 80% and 1.622 with a precision of 75% or higher. These results, compared with results presented in 5.1, suggest that sense tagging is a more error prone task than lexical selection in statistical machine translation systems.

Now we are interested in the effects that a bigger corpus could have. In figure 3 we present the results corresponding to alignments with a probability of 0.9 or higher for the 200K, 500K and 1M subsets of sentences of the Europarl corpus.



**Fig. 2.** Precision Berkeley Aligner for the automatically sense tagged 200K sentences Europarl corpus subset

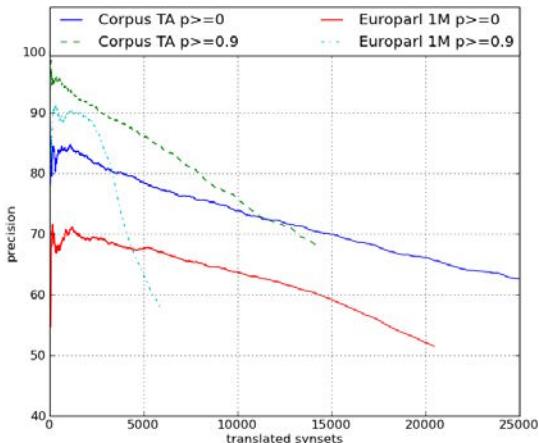


**Fig. 3.** Comparison of results for different subsets of the Europarl corpus for  $p \geq 0.9$

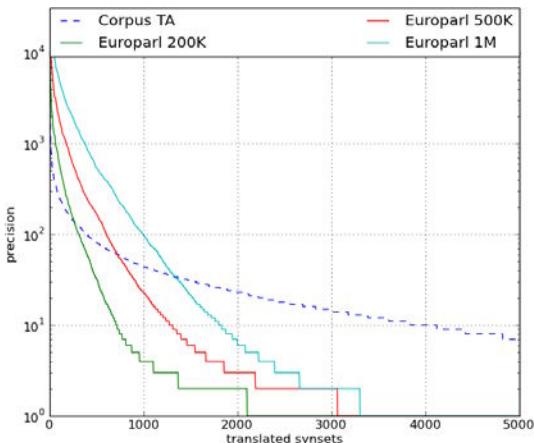
Increasing the size of the corpus has a positive effect in the results. For instance, with the 500K subset of sentences we get variants for 2.355 synsets with a precision of 80% or higher, instead of 1.360 corresponding to the 200K subset of sentences. This figure rises up to 3.390 for the 1M sentences subset.

### 5.3 Comparison of Results for Both Methods

In figure 4 we observe the results for both corpora: the machine translated manual sense tagged corpus and the automatic sense tagged parallel corpus (1M subset of sentences). As we see, we get better results using the method based on machine translation of sense disambiguated corpora. This suggests that lexical



**Fig. 4.** Comparison of the results for both methods



**Fig. 5.** Comparison of frequency distribution of synsets in both corpora

selection errors made by machine translation systems are less important than semantic tagging errors. But we need to further analyse the results in order to find other possible causes.

Another reason may be the different distribution of frequencies in both corpora, as shown in figure 5. As we observe, the frequency of synsets decreases more rapidly in the automatically sense tagged corpus (please, note the log  $y$  axis). This can be an additional reason, along with the sense tagging precision (about 73%).

## 6 Conclusions and Future Work

In this paper we present a methodology for WordNet construction and enlargement following the expand model based on the exploitation of sense tagged parallel corpora, taking English as a source text. Only the source text needs to be tagged with PWN synsets. With this resource, the task of constructing or enlarging a WordNet can be seen as a word alignment problem. Fortunately, this task is well known and several free algorithms are available. Unfortunately, the required corpus is not easily available. For this reason, we present two proposals for constructing such a corpus in an automatic way: (i) machine translation of a manually sense tagged corpus, and (ii) automatic sense tagging of a manually translated parallel corpus.

The methodology based on machine translation of sense disambiguated corpora achieves values of precision and number of synsets comparable to methodologies based on bilingual dictionaries for Spanish [1]. To perform a good comparison we need to further analyse our results to group variants according to the degree of polysemy. For Spanish, our best algorithm (Berkeley Aligner for  $p \geq 0.9$ ) performs better than all the criteria presented in [1] except monosemic-1 criterion. Nevertheless, our proposal performs worse than their combination of criteria, as they obtain 7.131 with a precision higher than 85%, whereas we only obtain 5.562 variants in the same conditions.

The methodology based on automatic tagging of parallel corpora performs much worse. Some reasons can be depicted, but it must be further studied, namely the precision of the sense tagging algorithm and the distribution of synset frequency in the corpus (maybe due to tagging errors or by the corpus typology).

Both methods are prone to errors but our experiments show that the methodology based on automatic sense tagging performs worse. In addition, increasing the size of the corpus has a beneficial effect on the automatic sense tagging method. As it is much easier to construct big parallel corpora than manually tagging monolingual corpora, the increase of the size of the corpus, as well as the selection of more general corpora are aspects to explore in the future.

An important aspect in these experiments that also must be further studied is the order of selection of the candidates. The task is aimed to get the maximum number of variants with the highest possible precision. In the experiments presented in this paper we get the variants in decreasing order of synset frequency in the corpus. Synsets with higher frequency are expected to get the corresponding translated variant with higher probability to be correct, but this is not always the case. In further experiments we plan to take advantage of the information given by the alignment algorithm to calculate a function that will allow us to select the variants in a better order.

One drawback of the method based on parallel corpora is the relatively low precision of the automatic sense tagging. To improve the precision we plan to use a multilingual parallel corpora to reduce the degree of ambiguity as depicted in [9].

All the experiments presented in this paper try to get a complete WordNet for a given language. Preliminary local WordNet versions are used only to au-

tomatically evaluate the results. In the future, we plan to take advantage of the acquired knowledge to use the preliminary versions to semantically tag the Spanish and Catalan part of the corpus. By doing this we will reduce the difficulty of the task, as some word alignment will be directly done by aligning the same synset ids in both languages.

We also plan to overcome some of the restrictions of the methods presented here: (i) to get more than one variant for each synset, observing the assigned probability of each alignment and taking more than one candidate if probability scores are similar enough; and (ii) to be able to get a lexical unit formed by more of one word as a variant.

## References

1. Atserias, J., Climent, S., Farrereres, X., Rigau, G., Rodriguez, H.: Combining multiple methods for the automatic construction of multi-lingual WordNets. In: Recent Advances in Natural Language Processing II. Selected papers from RANLP, vol. 97, pp. 327–338 (1997)
2. Azarova, I., Mitrofanova, O., Sinopalnikova, A., Yavorskaya, M., Oparin, I.: Russnet: Building a lexical database for the Russian language. In: Workshop on WordNet Structures and Standardisation, and how these affect WordNet Application and Evaluation, Las Palmas de Gran Canaria (Spain), pp. 60–64 (2002)
3. Benítez, S., Escudero, G., López, M., Rigau, G., Taulé, M.: Methods and tools for building the catalan WordNet. In: Proceedings of the ELRA Workshop on Language Resources for European Minority Languages (1998)
4. Brandt, M., Loftsson, H., Sigurðórsson, H., Tyers, F.: Apertium-IceNLP: a rule-based icelandic to english machine translation system. Reykjavík University, Reykjavík (2011) (unpublished paper)
5. Cilibraši, R.L., Vitanyi, P.M.: The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3), 370–383 (2007)
6. Diab, M.: The feasibility of bootstrapping an arabic WordNet leveraging parallel corpora and an english WordNet. In: Proceedings of the Arabic Language Technologies and Resources, NEMLAR, Cairo (2004)
7. Fellbaum, C.: WordNet: An electronic lexical database. The MIT Press (1998)
8. Fišer, D.: Leveraging parallel corpora and existing wordnets for automatic construction of the slovene wordnet. In: Proceedings of the 3rd Language and Technology Conference, vol. 7, p. 3–5 (2007)
9. Ide, N., Erjavec, T., Tufis, D.: Sense discrimination with parallel corpora. In: Proceedings of the ACL 2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, vol. 8. p. 61–66 (2002)
10. Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., Kanzaki, K.: Development of the japanese WordNet. In: Proceedings of the 6th LREC (2008)
11. Kazakov, D., Shahid, A.: Unsupervised construction of a multilingual WordNet from parallel corpora. In: Proceedings of the Workshop on Natural Language Processing Methods and Corpora in Translation, Lexicography, and Language Learning, pp. 9–12 (2009)
12. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit, vol. 5 (2005)
13. Liang, P., Taskar, B., Klein, D.: Alignment by agreement. In: Proceedings of the HLT-NAACL 2006 (2006)

14. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: Proceedings of the Workshop on Human Language Technology, HLT 1993, pp. 303–308. Association for Computational Linguistics, Stroudsburg (1993), ACM ID: 1075742
15. 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, ACL 2010, pp. 216–225. Association for Computational Linguistics, Stroudsburg (2010), ACM ID: 1858704
16. Oliver, A., Climent, S.: Construcción de los wordnets 3.0 para castellano y catalán mediante traducción automática de corpus anotados semánticamente. In: Proceedings of the 27th Conference of the SEPLN, Huelva, Spain (2011)
17. Oliver, A., Climent, S.: Building wordnets by machine translation of sense tagged corpora. In: Proceedings of the Global WordNet Conference, Matsue, Japan (2012)
18. Padró, L., Reese, S., Agirre, E., Soroa, A.: Semantic services in freeling 2.1: Wordnet and UKB. In: Proceedings of the 5th International Conference of the Global WordNet Association (GWC 2010) (2010)
19. Pedersen, B., Nimb, S., Asmussen, J., Sørensen, N., Trap-Jensen, L., Lorentzen, H.: DanNet: the challenge of compiling a wordnet for danish by reusing a monolingual dictionary. *Language resources and evaluation* 43(3), 269–299 (2009)
20. Rajendran, S., Arulmozi, S., Shanmugam, B., Baskaran, S., Thiagarajan, S.: Tamil WordNet. In: Proceedings of the First International Global WordNet Conference, Mysore, vol. 152, pp. 271–274 (2002)
21. Sagot, B., Fišer, D.: Building a free french wordnet from multilingual resources. In: Proceedings of OntoLex 2008, Marrakech, Morocco (2008)
22. Saveski, M., Trajkovski, I.: Automatic construction of wordnets by using machine translation and language modeling. In: 13th Multiconference Information Society, Ljubljana, Slovenia (2010)
23. Sinha, M., Reddy, M., Bhattacharyya, P.: An approach towards construction and application of multilingual indo-wordnet. In: 3rd Global Wordnet Conference (GWC 2006), Jeju Island, Korea (2006)
24. Tufis, D., Cristea, D., Stamou, S.: BalkaNet: aims, methods, results and perspectives: a general overview. *Science and Technology* 7(1-2), 9–43 (2004)
25. Vandeghinste, V., Martens, S.: PaCo-MT-D4. 2. report on lexical selection. Tech. rep., Centre for Computational Linguistics - KULeuven (2010)
26. Vossen, P.: Introduction to Eurowordnet. *Computers and the Humanities* 32(2), 73–89 (1998)

# Method to Build a Bilingual Lexicon for Speech-to-Speech Translation Systems

Keiji Yasuda, Andrew Finch, and Eiichiro Sumita

National Institute of Information and Communications Technology,  
3-5, Hikaridai, Keihanna Science City, Kyoto, 619-0289, Japan  
`{keiji.yasuda, andrew.finch, eiichiro.sumita}@nict.go.jp`

**Abstract.** Noun dropping and mis-translations occasionally occurs with Machine Translation (MT) output. These errors can cause communication problems between system users. Some of the MT architectures are able to incorporate bilingual noun lexica, which can improve the translation quality of sentences which include nouns. In this paper, we proposed an automatic method to enable a monolingual user to add new words to the lexicon. In the experiments, we compare the proposed method to three other methods. According to the experimental results, the proposed method gives the best performance in both point of view of Character Error Rate (CER) and Word Error Rate (WER). The improvement from using only a transliteration system is very large, about 13 points in CER and 32 points in WER.

## 1 Introduction

As a result of the dramatic advances in technical innovations of spoken language processing, speech-to-speech translation systems are starting to be used in real applications [1,2]. Especially for speech-to-speech translation systems in the travel domain, the coverage of nouns (such as names of accommodations, landmarks, places, restaurants and food) highly influences their performance.

To enable high portability to another domain, [3] proposed a method to introduce a bilingual lexicon into a phrase-based statistical machine translation (SMT) system [4,5]. Since this method requires a bilingual lexicon, monolingual users still can not register new words in the lexicon by themselves.

In this paper, we propose a method which enables monolingual MT users to register new words. The proposed method automatically finds the translation of new word by using a bilingual dictionary<sup>1</sup>, a target language  $n$ -gram model and a transliteration system.

Section 2 describes backgrounds and motivation of this research. Section 3 explains related works and the proposed method. Section 4 details the experiments using the field experiment data. Section 5 concludes the paper and presents some directions for future work.

---

<sup>1</sup> In this paper, we distinctly use the terms “dictionary” and “lexicon”. They express dictionary for human and speech translation system, respectively.

## 2 Backgrounds and Motivation

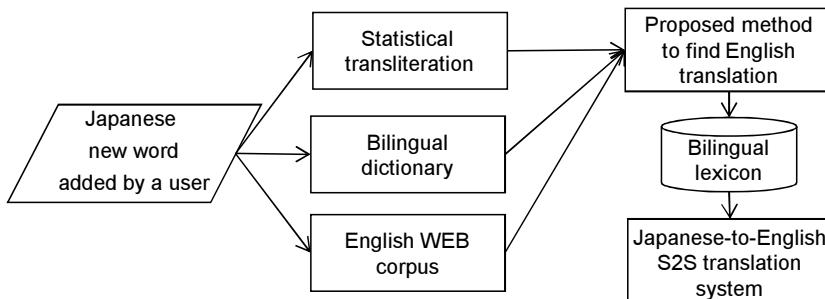
A typical speech-to-speech translation system consists of three subsystems: a speech recognition subsystem, an MT subsystem and a speech synthesis subsystem. Of the three subsystems, only MT requires bilingual information. Consequently, in this paper, we only deal with the MT component.

For the experiments in this paper we used a phrase-based SMT system incorporating a bilingual lexicon. Prior to translation, nouns in source sentences are replaced with high-frequency words from the same category in the training corpus, in the same manner as [3]. The target sentences are then acquired by translating the modified source sentences. Finally, the high-frequency words in the target sentences are replaced with target words for the untrained words.

The reason why high-frequency words are used is that we expect them to be already trained well. In other words, the high-frequency words may already appear frequently in the phrase tables and therefore provide ample statistics.

An advantage of this method is that a bilingual lexicon can be seen as independent from the SMT system. That is, even without knowledge of SMT development, people can add new words to the lexicon. A disadvantage of the method, however, is that it can not handle lexica with multiple translations for each entry as the translation probabilities for these should be trained. Hence, when we use a typical bilingual dictionary, which consists of one-to-many word translation pairs, it must be first somehow compiled into a one-to-one bilingual lexicon.

Figure 1 shows this process within the framework of a speech-to-speech translation system. As shown in the figure, the proposed method finds the English translation of a new word added by a user.



**Fig. 1.** Framework of the speech-to-speech translation system

### 3 Related Works and the Proposed Method

#### 3.1 Related Works

For our task, there are two conventional approaches that can be applied: a bilingual dictionary-based [6] approach and a transliteration approach [7,8,9]. Table 1 shows the advantages and disadvantages of these approaches.

First, with the bilingual dictionary approach, as mentioned in Section 2, we have to narrow down the candidate English words if the dictionary has more than one English translation. Since there is no contextual information for our task, it is difficult to apply the conventional word choice approach. In addition to the word choice problem, there is also a low-coverage problem in the dictionary-based approach. New words are often proper nouns, especially in our task, and existing bilingual dictionaries usually have poor coverage for proper nouns.

With the transliteration approach, since a typical system uses phoneme or grapheme mapping rules to produce transliterations, the systems sometimes yield non-word output or incorrect word output for the English translation.

The proposed method is a synergistic usage of both the dictionary-based and transliteration approaches. In addition, we also use web data for the verification of the transliteration. There is similar research on an Arabic-to-English news translation task [7]. However, there are differences between their task and ours. Compared to the news task, there is less possibility for a correct translation to be found in the web monolingual data because our task requires dealing with rare proper nouns used in local areas as needed by users. Additionally, there is no available contextual information for our task setting, where as [7] are able to exploit contextual information in their models.

**Table 1.** Advantages (+) and disadvantages (−) of the conventional methods and the proposed method

Method	Coverage	Robustness for non-word generation	Translation adequacy
Bilingual dictionary	−	+	+
Transliteration	+	−	+
Proposed method	+		

#### 3.2 Proposed Method

Figure 2 shows the flow of the proposed method. Firstly, the statistical based transliteration [9] module produces  $k$ -best<sup>2</sup> transliterations ( $T_n$ ) results by processing an input source language word  $s$  (1 in Fig. 2).

Secondly, by conducting dictionary look up, translation candidates ( $T_d$ ) are collected from a bilingual dictionary (2 in Fig. 2). In case there is no entry of  $s$  in

<sup>2</sup> We use 100-best in our experiments.

the bilingual dictionary, the proposed method simply reranks  $k$ -best based on  $n$ -gram count trained on web corpus. Here, there is no  $n$ -gram for any of  $k$ -best, the proposed method simply outputs 1-best in the original  $k$ -best order (3 in Fig. 2). In case there is one or more translation candidates in the dictionary, we search the most similar word pair  $(\hat{t}_n, \hat{t}_d)$  between  $T_n$  and  $T_d$  based on the similarity computed by eq. 1 (4 in Fig. 2). Should there be more than one best candidate, the one with the highest score from the transliteration model is selected.

$$(\hat{t}_n, \hat{t}_d) = \arg \min_{(t_n \in T_n, t_d \in T_d)} D(t_n, t_d) \quad (1)$$

Here,  $D(t_n, t_d)$  is the normalized edit distance computed by the following formula.

$$D(t_n, t_d) = \frac{D_{edit}(t_n, t_d)}{L_{t_d}} \quad (2)$$

where  $L_{t_d}$  is the length (number of characters) of the  $t_d$ , and  $D_{edit}(t_n, t_d)$  is the edit distance between  $t_n$  and  $t_d$ . If  $D(\hat{t}_n, \hat{t}_d)$  is greater than or equal to a threshold, the proposed method outputs  $\hat{t}_d$  (5 in Fig. 2). Otherwise, the  $k$ -best list is reranked by using  $n$ -gram counts (3 in Fig. 2).

Intuitively, the proposed method decides to use a translation pair from the bilingual dictionary, if  $s$  and  $\hat{t}_d$  are phonetically similar. In this case, the closest target language word is chosen from the translation candidates. Otherwise, the proposed method abandons using the bilingual dictionary and outputs word with the highest  $n$ -gram counts score from transliteration  $k$ -best.

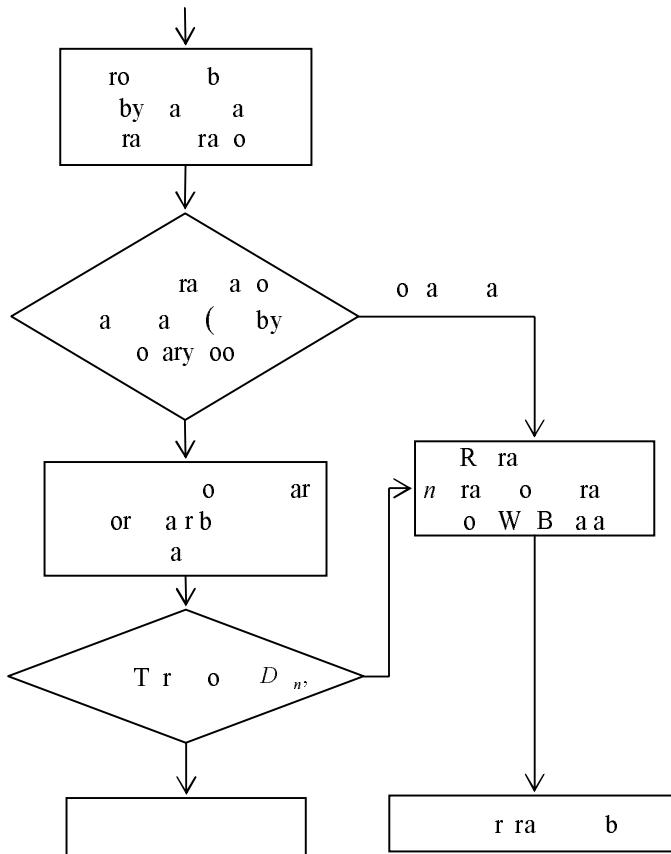
## 4 Experiments

### 4.1 Experimental Settings

The data set for the experiments came from speech-to-speech translation field experiments that occurred in the fiscal year 2009 [2]. As shown in Figure 3, the field experiments were undertaken in nationwide in Japan in five broad regions of Japan: Kanto, Kansai, Kyushu, Hokkaido and Chubu. The field experiments were undertaken as part of the Ministry of Internal Affairs and Communications initiative titled “Field Testing of Automatic Speech Translation Technology and its Contribution to Local Tourism.”

Prior to the field experiments, each of 5 projects added regional proper nouns to their bilingual lexica. To build a test set for the experiments, firstly, we extracted Katakana parts<sup>3</sup> from the bilingual lexicon of an accommodation and landmarks categories. It includes Katakana constituents of compound words.

<sup>3</sup> Katakana is a set of phonograms which is used in the Japanese writing system. Katakana is primarily used for transcription of loan words. Certain Japanese language words such as names of Japanese companies, building, and accommodation are also written in katakana rather than the other systems.

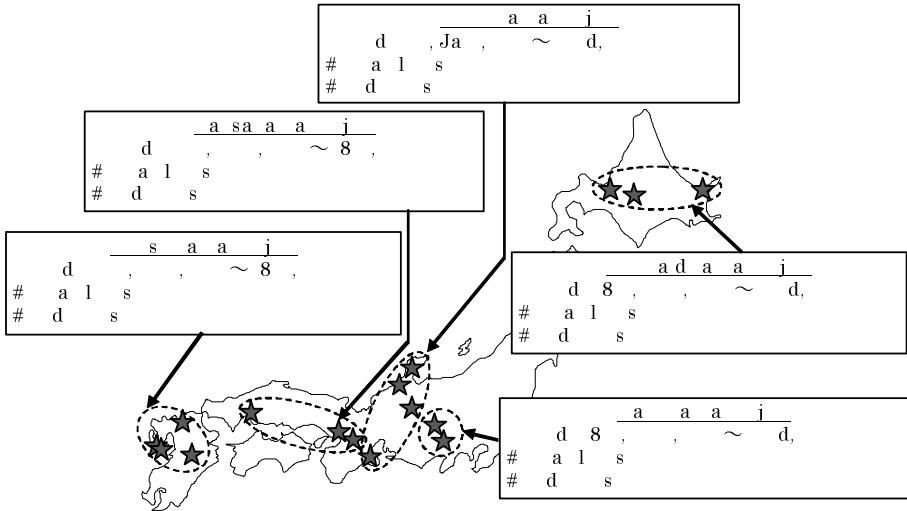
**Fig. 2.** Flow of the proposed method

1032 unique Katakana sequences. Out of 1032 sequences, 411 Katakana sequences have entries in the Eijiro bilingual dictionary ver.79<sup>4</sup> which we used for our experiments. We use the 411 Katakana sequences for the evaluation in our experiments.

For the  $n$ -gram count based reranker, we used the google  $n$ -gram corpus. The translation direction in all our experiments was from Japanese to English.

In the experiments, we use a transliteration system [10] trained on a bilingual lexicon extracted from Wikipedia [11]. A character sequence-pair is a tuple  $(\bar{\mathbf{k}}, \bar{\mathbf{a}})$  consisting of a sequence of katakana characters together with a sequence of English alphabet characters  $(\bar{\mathbf{k}}, \bar{\mathbf{a}}) = (<k_1, k_2, \dots, k_i>, <a_1, a_2, \dots, a_j>)$ .

<sup>4</sup> <http://eijiro.jp/>



**Fig. 3.** Overview of the five local projects

The training lexicon probability is simply the probability of all possible derivations of the lexicon given the set of sequence-pairs and their probabilities.

$$\begin{aligned} p(\bar{\mathbf{k}}_1^M, \bar{\mathbf{a}}_1^N) &= P(k_1, k_2, \dots, k_M, a_1, a_2, \dots, a_N) \\ &= \sum_{\gamma \in \Gamma} P(\gamma) \end{aligned}$$

where  $\gamma = ((\bar{\mathbf{k}}_1, \bar{\mathbf{a}}_1), \dots, (\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j), \dots, (\bar{\mathbf{k}}_J, \bar{\mathbf{a}}_J))$  is a derivation of the lexicon characterized by its co-segmentation, and  $\Gamma$  is the set of all derivations (co-segmentations) of the lexicon.

The probability of a single derivation is given by the product of its component character sequence-pairs.

$$p(\gamma) = \prod_{j=1}^J P((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)) \quad (3)$$

The lexicon for our experiments is segmented into katakana and alphabet word-pairs. We therefore constrain our model such that both katakana and alphabet character sequences of each character sequence-pair in the derivation of the lexicon are not allowed to cross a word segmentation boundary. Equation 3 can therefore be arranged as a product of word-pair  $w$  derivations of the sequence of all word-pairs  $\mathcal{W}$  in the lexicon.

$$p(\gamma) = \prod_{w \in \mathcal{W}} \prod_{(\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j) \in \gamma_w} P((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)) \quad (4)$$

where  $\gamma_w$  is a derivation of katakana and alphabet word-pair  $w$ .

The Dirichlet process model we use in our approach is a simple model that resembles the cache models used in language modeling [12]. Intuitively, the model has two basic components: a model for generating an outcome that has already been generated at least once before, and a second model that assigns a probability to an outcome that has not yet been produced. Ideally, to encourage the re-use of model parameters, the probability of generating a novel sequence-pair should be considerably lower than the probability of generating a previously observed sequence-pair. This is a characteristic of the Dirichlet process model we use and furthermore, the model has a preference to generate new sequence-pairs early on in the process, but is much less likely to do so later on. In this way, as the cache becomes more and more reliable and complete, so the model prefers to use it rather than generate novel sequence-pairs. The probability distribution over these character sequence-pairs (including an infinite number of unseen pairs) can be learned directly from unlabeled data by Bayesian inference of the hidden cosegmentation of the lexicon.

For the *base measure* that controls the generation of novel words, we use a joint spelling model that assigns probability to new words according to the following joint distribution:

$$G_0((\bar{\mathbf{k}}, \bar{\mathbf{a}})) = p(|\bar{\mathbf{k}}|)p(\bar{\mathbf{k}}|\bar{\mathbf{k}}|) \times p(|\bar{\mathbf{a}}|)p(\bar{\mathbf{a}}|\bar{\mathbf{a}}|) \\ = \frac{\lambda_k^{|\bar{\mathbf{k}}|}}{|\bar{\mathbf{k}}|!} e^{-\lambda_k} v_k^{-|\bar{\mathbf{k}}|} \times \frac{\lambda_a^{|\bar{\mathbf{a}}|}}{|\bar{\mathbf{a}}|!} e^{-\lambda_a} v_a^{-|\bar{\mathbf{a}}|} \quad (5)$$

where  $|\bar{\mathbf{k}}|$  and  $|\bar{\mathbf{a}}|$  are the length in characters of the katakana and alphabet sides of the character sequence-pair;  $v_k$  and  $v_a$  are the vocabulary (alphabet) sizes of the katakana and alphabet respectively; and  $\lambda_k$  and  $\lambda_a$  are the expected lengths of katakana and alphabet. In our experiments, we set  $|\bar{\mathbf{k}}| = 1$ , in other words, we only allow to align single katakana character and one or more alphabet characters.

The generative model is given in Equation 6 below. The equation assigns a probability to the  $j^{\text{th}}$  character sequence-pair  $(\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)$  in a derivation of the lexicon, given all of the other phrase-pairs in the history so far  $(\bar{\mathbf{k}}_{-j}, \bar{\mathbf{a}}_{-j})$ . Here  $-j$  is read as: “up to but not including  $j$ ”.

$$p((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)|(\bar{\mathbf{k}}_{-j}, \bar{\mathbf{a}}_{-j})) = \frac{N((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)) + \alpha G_0((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j))}{N + \alpha} \quad (6)$$

In this equation,  $N$  is the total number of character sequence-pairs generated so far,  $N((\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j))$  is the number of times the phrase-pair  $(\bar{\mathbf{k}}_j, \bar{\mathbf{a}}_j)$  has occurred in the history.  $G_0$  is the base measure and  $\alpha$  is a concentration parameter that determines how close the resulting distribution over sequence-pairs is to  $G_0$ .

For the model training, a blocked version of a Gibbs sampler is used. Details of the algorithm are explained in [12,13,10].

## 4.2 Experimental Results

Table 2 shows the evaluation results of the proposed method. Here, we conducted four kinds of experiments. The first system is a simple statistical transliteration system (1st block of the Table 2) [9]. The second system is the statistical transliteration system combined with the  $n$ -gram reranker. This system chooses the best transliteration result from 100-best transliteration output using  $n$ -gram count-based reranking (2nd block of the Table 2). The third system is the proposed method without  $n$ -gram reranking; this setting simply outputs the 1-best in the original transliteration  $k$ -best list, in the case where the dictionary driven part yields no output (3rd block of the Table 2). The fourth system is the proposed method (4th block of the Table 2). For the third and fourth systems, we vary the threshold (5 in Fig. 2) between 0.0 to 0.4. To evaluate the systems we measure: Character Error Rate (CER) and Word Error Rate(WER).

**Table 2.** Evaluation results of the proposed method

Resource usage			Character Error Rate	Word Error Rate
Transliteration	ngram	Dictionary (threshold)		
Used	Not used	Not used	30.55%	69.34%
Used	Used	Not used	27.06%	57.18%
Used	Not used	Used (0.0)	18.98%	45.99%
Used	Not used	Used (0.1)	17.93%	40.63%
Used	Not used	Used (0.2)	17.55%	38.44%
Used	Not used	Used (0.3)	18.71%	38.44%
Used	Not used	Used (0.4)	19.41%	38.20%
Used	Used	Used (0.0)	21.27%	43.31%
Used	Used	Used (0.1)	18.40%	38.69%
Used	Used	Used (0.2)	17.16%	37.23%
Used	Used	Used (0.3)	18.32%	37.71%
Used	Used	Used (0.4)	19.18%	37.96%

Looking at the table, the single transliteration system is improved (about 3 points in CER and 12 points in WER) by using  $n$ -gram data. The proposed method gives the best performance with a threshold of 0.2. The improvement from the single transliteration system is substantial, about 13 points in CER and 32 points in WER.

## 5 Conclusions and Future Work

We proposed a method of combining dictionaries, automatic transliteration and web data in complementary manner to build a bilingual lexicon for speech-to-speech translation systems. We carried out Japanese to English word translation experiments using travel dialogue data collected by speech translation field experiments conducted in Japan.

In the experiments, we compare the proposed method to three other methods. According to the experimental results, the proposed method gives the best performance in term of both CER and WER. The improvement over the single transliteration system is very large, which is about 13 points in CER and 32 points in WER.

In this paper we only deal with words written in the Katakana character set, however, the Japanese writing system has two more character sets: Hiragana and Kanji. In the future work, we will address the words written in these character set. In reality, one compound word can be a mixture of these three kinds of characters. One such example from the field experiments' landmark lexicon being the expression in Japanese: *kashuni notaki*. This should be correctly translated as "Kashuni Falls" and is formed using a combination of transliteration and translation. Future research will need to address ways of identifying when to transliterate and when to translate inside these compound units.

## References

1. Bach, N., Hsiao, R., Eck, M., Charoenpornsawat, P., Vogel, S., Schultz, T., Lane, I., Waibel, A., Black, A.W.: Incremental adaptation of speech-to-speech translation. In: Proc. of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pp. 149–152 (2009)
2. Kawai, H., Isotani, R., Yasuda, K., Sumita, E., Masao, U., Matsuda, S., Ashikari, Y., Nakamura, S.: An overview of a nation-wide field experiment of speech-to-speech translation in fiscal year 2009. In: Proceedings of 2010 Autumn Meeting of Acoustical Society of Japan, pp. 99–102 (2010) (in Japanese)
3. Okuma, H., Yamamoto, H., Sumita, E.: Introducing a translation dictionary into phrase-based smt. The IEICE Transactions on Information and Systems 91-D(7), 2051–2057 (2008)
4. Koehn, P., Och, F.J., Marcu, D.: Statistical Phrase-Based Translation. In: Proc. of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pp. 127–133 (2003)
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180. Association for Computational Linguistics (2007)
6. Tonoike, M., Kida, M., Takagi, T., Sasaki, Y., Utsuro, T., Sato, S.: Translation Estimation for Technical Terms using Corpus collected from the Web. In: Proceedings of the Pacific Association for Computational Linguistics, pp. 325–331 (2005)
7. Al-Onaizan, Y., Knight, K.: Translating named entities using monolingual and bilingual resources. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 400–408 (2002)
8. Sato, S.: Web-Based Transliteration of Person Names. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 273–278 (2009)

9. Finch, A., Dixon, P., Sumita, E.: Integrating a joint source channel model into a phrase-based transliteration system. In: Proceedings of NEWS 2011 (2011) will be appeared
10. Finch, A., Sumita, E.: A bayesian model of bilingual segmentation for transliteration. In: Proceedings of the Seventh International Workshop on Spoken Language Translation (IWSLT), pp. 259–266 (2010)
11. Fukunishi, T., Finch, A., Yamamoto, S., Sumita, E.: Using features from a bilingual alignment model in transliteration mining. In: Proceedings of NEWS 2011 (2011)
12. Goldwater, S., Griffiths, T.L., Johnson, M.: Contextual dependencies in unsupervised word segmentation. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 673–680. Association for Computational Linguistics, Morristown (2006)
13. Mochihashi, D., Yamada, T., Ueda, N.: Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In: ACL-IJCNLP 2009: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, vol. 1, pp. 100–108. Association for Computational Linguistics, Morristown (2009)

# A Fast Subspace Text Categorization Method Using Parallel Classifiers

Nandita Tripathi<sup>1</sup>, Michael Oakes<sup>1</sup>, and Stefan Wermter<sup>2</sup>

<sup>1</sup> Department of Computing, Engineering and Technology, University of Sunderland,  
St. Peters Way, Sunderland, SR6 0DD, United Kingdom

Nandita.Tripathi@research.sunderland.ac.uk,  
Michael.Oakes@sunderland.ac.uk

<sup>2</sup> Institute for Knowledge Technology, Department of Computer Science,  
University of Hamburg, Vogt Koelln, Str. 30, 22527 Hamburg, Germany  
wermter@informatik.uni-hamburg.de

**Abstract.** In today's world, the number of electronic documents made available to us is increasing day by day. It is therefore important to look at methods which speed up document search and reduce classifier training times. The data available to us is frequently divided into several broad domains with many sub-category levels. Each of these domains of data constitutes a subspace which can be processed separately. In this paper, separate classifiers of the same type are trained on different subspaces and a test vector is assigned to a subspace using a fast novel method of subspace detection. This parallel classifier architecture was tested with a wide variety of basic classifiers and the performance compared with that of a single basic classifier on the full data space. It was observed that the improvement in subspace learning was accompanied by a very significant reduction in training times for all types of classifiers used.

**Keywords:** Text Categorization, Subspace Learning, Semantic Subspaces, Maximum Significance Value, Conditional Significance Vectors.

## 1 Introduction

The huge amount and variety of data available to us today makes document search and classifier training a lengthy process. Due to the ever increasing volume of documents on the web, classifiers have to be periodically retrained to keep up with the increasing variation. Reduced classifier training times are therefore a big asset in keeping classifiers up to date with the current content. Classifier application efficiency (test efficiency) is also very important in returning search results. Retrieving a relevant document quickly in the presence of millions of records (the web) is an essential characteristic for a search engine today. In addition to this, the *curse of dimensionality* [1] degrades the performance of many learning algorithms. The large number of dimensions reduces the effectiveness of distance measures [2]. Today's data also contains a large number of data domains which can be as diverse as

medicine and politics. These data domains can be considered as independent subspaces of the original data.

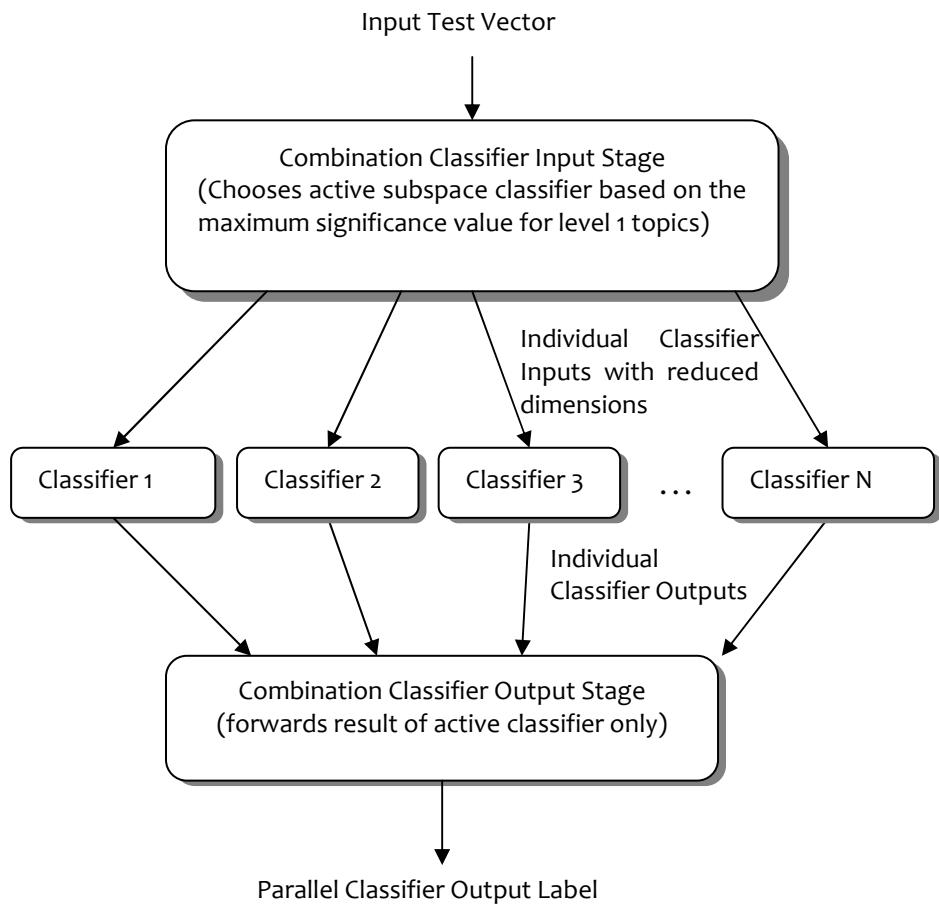
Independent data domains give rise to the idea of using parallel classifiers. Instead of training a single classifier on the full dataset, we can use many classifiers in parallel to process these independent subspaces. Classifier performances can be improved further by using only a subset of the dimensions. Active research is going on in the area of dimension reduction [3].

Random Projections [4] have also been used in dimensionality reduction. In the Random Subspace Method (RSM) [5], classifiers were trained on randomly chosen subspaces of the original input space and the outputs of the models were then combined. However, random selection of features does not guarantee that the selected inputs have the necessary distinguishing information. Several variations of RSM have been proposed by various researchers such as Relevant Random Feature Subspaces for Co-training (Rel-RASCO) [6], Not-so-Random Subspace Method (NsRSM) [7] and Local Random Subspace Method [8].

There are many methods of classifier combination. One method is to use many classifiers of the same or different types on different portions of the input data space. The combining classifier decides which part of the input data has to be applied to which base classifier. Two special types of classifier combinations are Bagging [9] and Boosting [10] which use a large number of primitive classifiers of the same type (e.g. a decision stump) on weighted versions of the original data.

Many classifier combination methods have been applied to text categorization. In one method [11], text and metadata were considered as separate descriptions of the same object. Another text categorization method [12] was based on a hierarchical array of neural networks. The problem of large class imbalances in text classification tasks was addressed by using a mixture-of-experts framework [13].

In the real world, documents can be divided into major semantic subspaces with each subspace having its own unique characteristics. The above research does not take into account this division of documents into different semantic subspaces. Therefore, we present here a novel parallel architecture (Fig. 1) which takes advantage of the different semantic subspaces existing in the data. We further show that this new parallel architecture improves subspace classification accuracy as well as it significantly reduces training time. For this architecture, we use parallel combinations of classifiers with a single type of base classifier. We use the conditional significance vector representation [14] which is a variation of the semantic significance vector [15], [16] to incorporate semantic information into the document vectors. The conditional significance vector enhances the distinction between subtopics within a given main topic. The region of the test data is determined by the maximum significance value which is evaluated in  $O(k)$  time where  $k$  is the number of level 1 topics and thus can be very effective where time is critical for returning search results.



**Fig. 1.** Parallel Classifier Architecture for Subspace Learning

## 2 Methodology and Architecture

In our experiments, we used the Reuters Corpus [17] as it is a well-known test bench for text categorization experiments. It also has a hierarchical organization with four major groups which is well suited to test the classification performance of a parallel architecture. We used the Reuters Corpus headlines for our experiments as they provide a concise summary of each news article. Each Reuters headline consists of one line of text with about 3 – 12 words. Some examples of Reuters Headlines are:

*"Estonian president faces reelection challenge."*

*"Guatemalan sides to sign truce in Norway report."*

The topic codes in the Reuters Corpus represent the subject areas of each news story. They are organized into four hierarchical groups, with four top-level nodes: Corporate/Industrial (CCAT), Economics (ECAT), Government/Social (GCAT) and Markets (MCAT). Ten thousand headlines along with their topic codes were extracted from the Reuters Corpus. These headlines were chosen so that there was no overlap at the first level categorization. Each headline belonged to only one level 1 category. At the second level, since most headlines had multiple level 2 subtopic categorizations, the first subtopic was taken as the assigned subtopic. Thus, each headline had two labels associated with it – the main topic (Level 1) label and the subtopic (Level 2) label. Headlines were then preprocessed to separate hyphenated words. Dictionaries with term frequencies were generated based on the TMG toolbox [18] and were then used to generate the Full Significance Vector [14], the Conditional Significance Vector [14] and the tf-idf [19] representation for each document. The datasets were then randomized and divided into a training set of 9000 documents and a test set of 1000 documents.

The WEKA machine learning workbench [20] provided various learning algorithms which we used as base classifiers to test our parallel architecture. Six algorithms were used as base classifiers in parallel classifier representations to examine the performance of various classification algorithms. Classification accuracy, training time and testing time were recorded for each experiment run. The average of ten runs for each representation was used to compare the various classifiers.

### 3 Data Processing for Experiments

#### 3.1 Text Data Processing

Ten thousand Reuters headlines were used in these experiments. The Level 1 categorization of the Reuters Corpus divides the data into four main topics. These main topics and their distribution in the data along with the number of subtopics of each main topic in this data set are given in Table 1. Level 2 categorization further divides these main topics into subtopics. Here we took the direct (first level nesting) subtopics of each main topic occurring in the 10,000 headlines. A total of 50 subtopics were included in these experiments. Since all the headlines had multiple subtopic assignments, e.g. C11/C15/C18, only the first subtopic e.g. C11 was taken as the assigned subtopic. Our assumption here is that the first subtopic used to tag a particular Reuters news item is the one which is most relevant to it.

**Table 1.** Reuters Level 1 Topics

No.	Main Topic	Description	Number Present	Number of Subtopics
1.	CCAT	Corporate/Industrial	4600	18
2.	ECAT	Economics	900	8
3.	GCAT	Government/Social	1900	20
4.	MCAT	Markets	1600	4

### 3.2 Semantic Significance Vector Generation

We used a vector representation which represents the significance of the data and weighs different words according to their significance for different topics. Significance Vectors [15], [16] were determined based on the frequency of a word in different semantic categories. A modification of the significance vector called the semantic vector uses normalized frequencies where each word  $w$  is represented with a vector  $(c_1, c_2, \dots, c_n)$  where  $c_i$  represents a certain semantic category and  $n$  is the total number of categories. A value  $v(w, c_i)$  is calculated for each element of the semantic word vector as follows:

$$v(w, c_i) = \frac{\text{Normalized Frequency of } w \text{ in } c_i}{\sum_{k=1}^n \text{Normalized Frequency of } w \text{ in } c_k}$$

For each document, the document semantic vector is obtained by summing the semantic vectors for each word in the document and dividing by the total number of words in the document. Henceforth it is simply referred to as the *significance vector*. The TMG Toolbox [18] was used to generate the term frequencies for each word in each news document. Word vectors were generated for the main and subtopic levels separately and then concatenated. The final word vector consisted of 54 columns (for 4 main topics and 50 subtopics) for the Reuters Corpus. While calculating the *significance vector* entries for each word, its occurrence in all subtopics of all main topics was taken into account. This was called the *Full Significance Vector* [14]. We also generated the *Conditional Significance Vector* [14] where a word's occurrence in all subtopics of *only a particular main topic* is taken into account while calculating the word significance vector entries.

For each document, the document significance vector was obtained by summing the significance vectors for each word in the document and dividing this sum by the total number of words in the document.

### 3.3 Data Vector Sets Generation

Three different vector representations (Full Significance Vector, Conditional Significance Vector and tf-idf) were generated for our data. The Conditional Significance Vectors were processed further to generate four main category-wise data vector sets.

#### 3.3.1 Full Significance Vector

Here, the document vectors were generated by using the full significance word vectors as explained in section 3.2. For each Reuters Full Significance document vector the first four columns, representing four main topics – CCAT, ECAT, GCAT & MCAT, were ignored leaving a vector with 50 columns representing 50 subtopics. The order of the data vectors was then randomised and divided into two sets – a training set of 9000 vectors and a test set of 1000 vectors.

#### 3.3.2 Category-Based Conditional Significance Vectors

Here, the conditional significance word vectors were used to generate the document vectors. The order of the 10,000 Reuters Conditional Significance document vectors was randomised and divided into two sets – a training set of 9000 vectors and a test set of 1000 vectors. The training set was then divided into 4 sets according to the main topic labels. For each of these sets, only the relevant subtopic vector entries (e.g. C11, C12, etc for CCAT; E11, E12, etc for ECAT) for each main topic were retained. Thus, the CCAT category training data set had 18 columns for the 18 subtopics of CCAT. Similarly the ECAT training data set had 8 columns, the GCAT training data set had 20 columns and the MCAT training data set had 4 columns. These 4 training sets were then used to train the 4 separate base classifiers of the Reuters parallel classifier. The main category of a test data vector was determined by the maximum significance vector entry for the first four columns representing the four main categories. After this, the entries corresponding to the subtopics of this predicted main topic were extracted along with the *actual* subtopic label and given to the classifier trained for this predicted main category.

For the Reuters Corpus, the accuracy of choosing the correct main topic by selecting the maximum significance level 1 entry was 96.80% for the 1000 test vectors, i.e. 968 vectors were assigned to the correct trained classifiers whereas 3.20% or 32 vectors were assigned to a wrong classifier – resulting in a wrong classification decision for all these 32 vectors. Hence the upper limit for classification accuracy was 96.80% for our parallel classifier for the Reuters Corpus.

#### 3.3.3 TF-IDF Vector Generation

The tf-idf weight (Term Frequency–Inverse Document Frequency) measures how important a word is to a document in a data set. This importance increases with the

number of times a word appears in the document but is reduced by the frequency of the word in the data set. Words which occur in almost all the documents have very little discriminatory power and hence are given very low weight. The TMG toolbox [18] was used to generate the tf-idf vectors for our experiments. The tf-idf vector datasets were then randomized and divided into 9000 training /1000 test vectors.

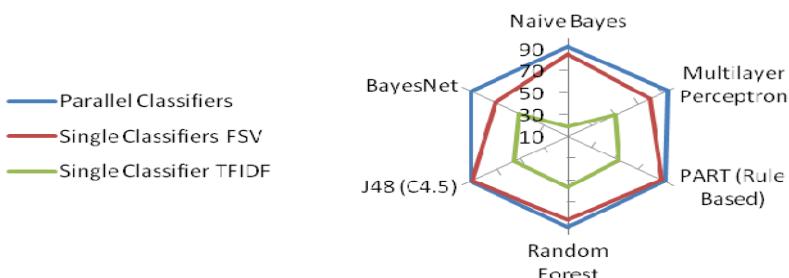
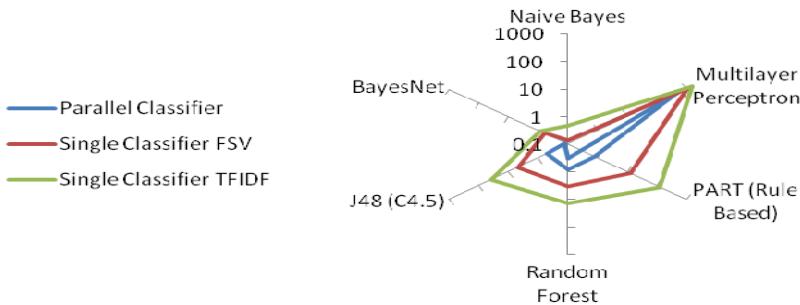
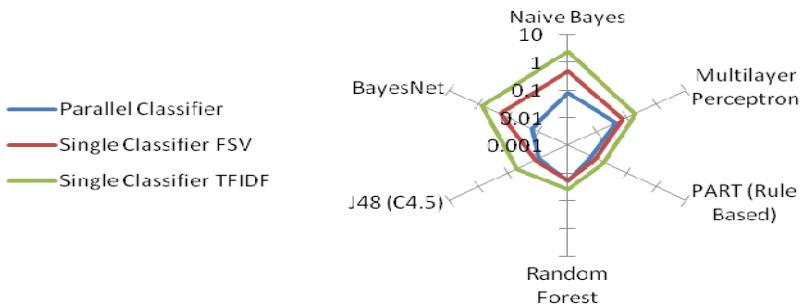
### 3.4 Classification Algorithms

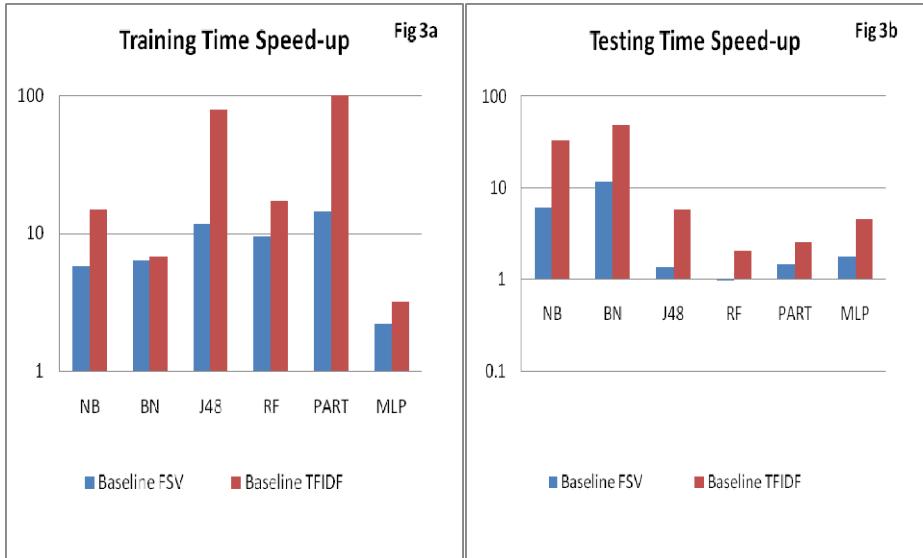
Six classification algorithms were tested with our data sets namely Random Forest, J48(C4.5), the Multilayer Perceptron, Naïve Bayes, BayesNet, and PART. Random Forests [21] are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently. C4.5 [22] is an inductive tree algorithm with two pruning methods: subtree replacement and subtree raising. The Multilayer Perceptron [23] is a neural network which uses backpropagation for training. Naive Bayes [24] is the simplest form of Bayesian network, in which all attributes are independent given the value of the class variable. BayesNet [25] implements Bayes Network learning using various search algorithms and quality measures. A PART [26] decision list uses C4.5 decision trees to generate rules.

## 4 Results and Analysis

We tested our parallel classifier architecture using six different types of base classifiers. In the parallel classifier using Naïve Bayes, four different Naïve Bayes classifiers were trained on the four subspaces of the Reuters Corpus namely CCAT, ECAT, GCAT and MCAT. Similarly for the parallel classifier using Multilayer Perceptrons, four different Multilayer Perceptron classifiers were trained on the four subspaces of the Reuters Corpus and so on. The performance of each single classifier on the full data was compared with the performance of the parallel classifier combination in which this particular classifier was used as a base classifier. For the baseline single classifier experiments, the Full Significance Vector and the tf-idf vector representations were used whereas for the parallel classifier experiments, the category-wise separated Conditional Significance Vector representation was used.

In all comparisons, it was observed that the parallel classifier combination performed better than the single basic classifier. The classification accuracy was improved (Friedman test,  $p=0.0025$ ), the training times were reduced (Friedman test,  $p=0.0025$ ) and the testing times were reduced (Friedman test,  $p=0.0094$ ). The baseline using Full Significance Vectors (FSV) performed better than the baseline using tf-idf. Fig 2 shows the subtopic classification accuracy, training time and testing time for the parallel classifiers along with the baselines. Fig 2a shows that the maximum improvement in subtopic classification accuracy is achieved by the Naïve Bayes Classifier while the other classifiers also show a substantial improvement.

**Subtopic Classification Accuracy %****Fig 2a****Training Time (Log Scale)****Fig 2b****Testing Time (Log Scale)****Fig 2c****Fig. 2.** Parallel Classifier Performance Metrics

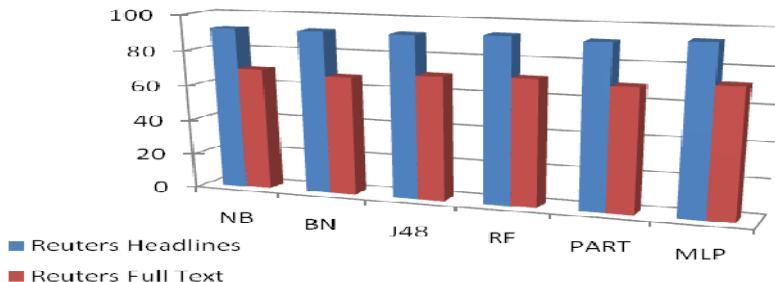


**Fig. 3.** Parallel Classifier Speed-up

Fig 3 shows the speed-up of the parallel classifiers with respect to both baselines. Speed-up is calculated by dividing the baseline time by the corresponding parallel classifier time. The timing diagrams in Fig. 2 and the speed-up diagrams in Fig. 3 are shown on a log scale to accommodate a wide range of values. The maximum training speed-up was achieved by the rule-based classifier PART (14.4 with reference to the FSV baseline and 149 with reference to the tf-idf baseline) which was followed by the tree-based classifier J48(C4.5) at speed-up 11.76 with reference to the FSV baseline and 79.5 with reference to the tf-idf baseline. The testing time speed-up was maximum for the Bayesian classifiers. Naïve Bayes achieved a speed-up of 6 with respect to FSV and 32.8 with respect to tf-idf while BayesNet achieved a speed-up of 11.75 and 48.75 with the corresponding baselines. Naïve Bayes achieved significant speed-up in both training and as well as testing (Train/Test speed-up of 5.8/6.0 and 15.1/32.8 for FSV and tf-idf respectively).

We also ran the parallel classifier experiments on 10,000 Reuters Full Text news items (containing headlines and body text). It was observed that the subtopic classification accuracy of Reuters news items was better with Reuters Headlines than with Reuters Full Text (Wilcoxon Signed Rank test,  $p=0.031$ ). A possible explanation for this can be that the extra text present in Reuters Full Text acts as noise which degrades classifier performances. Fig 4 shows the corresponding subtopic classification accuracies.

### Subtopic Classification Accuracy (Parallel Classifiers)



#### Classifier Index:

NB – Naïve Bayes

BN – BayesNet

J48 – C4.5 Tree

RF – Random Forest

PART – Rule Based Classifier

MLP – Multilayer Perceptron

**Fig. 4.** Comparison of Reuters Headlines and Reuters Full Text

## 5 Conclusion

Our results show that combining classifiers of the same type in parallel improves the classification accuracy of the concerned basic classifier where the underlying data has distinct semantic categories. They also show that Reuters Headlines perform better than Reuters Full Text for the purpose of news categorization. These results show further that a parallel combination of classifiers results in a very sharp reduction in training and testing times. The speed-up achieved is very significant in all cases. Naïve Bayes achieved a significant speed-up in *both* training and test timings along with the maximum improvement in classification accuracy. Since Naïve Bayes is already a fast classifier, further speedup can be put to good use especially in search technology. The experiments confirm the fact that the Maximum Significance Value is very effective in detecting the relevant subspace of a test document and that training separate classifiers on different subsets of the original data enhances overall classification accuracy and significantly reduces training/testing times.

## References

1. Friedman, J.H.: On Bias, Variance, 0/1—Loss, and the Curse-of- Dimensionality. *Data Mining and Knowledge Discovery* 1(1), 55–77 (1997)
2. Parsons, L., Haque, E., Liu, H.: Subspace Clustering for High Dimensional Data: A Review. *ACM SIGKDD Explorations Newsletter* 6(1), 90–105 (2004)

3. Varshney, K.R., Willsky, A.S. : Learning dimensionality-reduced classifiers for information fusion. In: Proceedings of the 12th International Conference on Information Fusion, pp. 1881–1888 (July 2009)
4. Fradkin, D., Madigan, D.: Experiments with Random Projections for Machine Learning. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517–522 (2003)
5. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844 (1998)
6. Yaslan, Y., Cataltepe, Z.: Co-training with relevant random subspaces. *Neurocomputing* 73, 1652–1661 (2010)
7. Garcia-Pedrajas, N., Ortiz-Boyer, D.: Boosting Random Subspace Method, vol. 21, pp. 1344–1362 (2008)
8. Kotsiantis, S.B.: Local Random Subspace Method for constructing multiple decision stumps. In: International Conference on Information and Financial Engineering, pp. 125–129 (2009)
9. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
10. Schapire, R.E.: The boosting approach to machine learning: An overview. In: Nonlinear Estimation and Classification. Lecture Notes in Statist., vol. 171, pp. 149–171. Springer, New York (2003)
11. Al-Kofahi, K., et al.: Combining multiple classifiers for text categorization. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM 2001, pp. 97–104 (2001)
12. Ruiz, M.G., Srinivasan, P.: Hierarchical Neural Networks for Text Categorization. In: SIGIR 1999 (1999)
13. Estabrooks, A., Japkowicz, N.: A mixture-of-experts framework for text classification. In: Proceedings of the 2001 Workshop on Computational Natural Language Learning, Toulouse, France, July 6–7, vol. 7, pp. 1–8 (2001)
14. Tripathi, N., et al.: Semantic Subspace Learning with Conditional Significance Vectors. In: Proceedings of the IEEE International Joint Conference on Neural Networks, Barcelona, pp. 3670–3677 (July 2010)
15. Wermter, S., Panchev, C., Arevian, G.: Hybrid Neural Plausibility Networks for News Agents. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 93–98 (1999)
16. Wermter, S.: Hybrid Connectionist Natural Language Processing. Chapman and Hall (1995)
17. Rose, T., Stevenson, M., Whitehead, M.: The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), pp. 827–833 (2002)
18. Zeimpekis, D., Gallopoulos, E.: Generating Term Document Matrices from Text Collections. In: Kogan, J., Nicholas, C. (eds.) Grouping Multidimensional Data: Recent Advances in Clustering, Springer, Heidelberg (2005)
19. Manning, C., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
20. Hall, M., et al.: The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter* 11(1), 10–18 (2009)
21. Breiman, L.: Random Forests. *Machine Learning* 45(1), 5–32 (2001)
22. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)

23. Verma, B.: Fast training of multilayer perceptrons. *IEEE Transactions on Neural Networks* 8(6), 1314–1320 (1997)
24. Zhang, H., Su, J.: Naive Bayes for Optimal ranking. *Journal of Experimental and Theoretical Artificial Intelligence* 20(2), 79–93 (2008)
25. Pernkopf, F.: Discriminative learning of Bayesian network classifiers. In: *Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, pp. 422–427 (2007)
26. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Shavlik, J. (ed.) *Machine Learning: Proceedings of the Fifteenth International Conference*. Morgan Kaufmann Publishers (1998)

# Research on Text Categorization Based on a Weakly-Supervised Transfer Learning Method

Dequan Zheng, Chenghe Zhang, Geli Fei, and Tiejun Zhao

MOE-MS Key Laboratory of Natural Language Processing and Speech  
Harbin Institute of Technology, Harbin, China  
dqzheng2007@gmail.com

**Abstract.** This paper presents a weakly-supervised transfer learning based text categorization method, which does not need to tag new training documents when facing classification tasks in new area. Instead, we can take use of the already tagged documents in other domains to accomplish the automatic categorization task. By extracting linguistic information such as part-of-speech, semantic, co-occurrence of keywords, we construct a domain-adaptive transfer knowledge base. Relation experiments show that, the presented method improved the performance of text categorization on traditional corpus, and our results were only about 5% lower than the baseline on cross-domain classification tasks. And thus we demonstrate the effectiveness of our method.

**Keywords:** Transfer learning, Text Categorization.

## 1 Introduction

With the explosion of the text documents on the web, text classification technique has been playing a more and more essential role in helping people find, collect and organize these data. As a result, the study on how to use computer to classify texts automatically has become a key realm in both natural language processing and artificial intelligence field.

Traditional text classification techniques are usually based on machine learning, which means people have to train a categorization model in advance. However, traditional machine learning methods rely on strong assumptions: the first assumption is that training and testing data set should be evenly distributed; and the other is that they should be in homogeneous feature space. Unfortunately, this is not always true in reality, which may lead to the failure of the text classifier in many cases, such as outdated training data set.

At this time, people have to label a large quantity of texts in new domains to meet the needs of training, but tagging new texts and training new models are extremely expensive and time consuming. From another point of view, if we already have a large volume of labeled texts in one domain, it is wasteful to totally abandon them. So how to make full use of these data is what a method called transfer learning aims to solving. Transfer learning means people may extract transfer knowledge from the data

available at present and use them in the future, or extract transfer knowledge from one domain and use it in other domains.

As a result, this paper proposes a novel domain-adaptive transfer learning method, which combines linguistic information and statistics. Through learning from available data or knowledge base at hand, we construct a new transfer knowledge base in heterogeneous feature space without tagging new corpuses.

The rest of this paper is organized as follows. In section 2, we give an introduction of related works. In section 3, we describe our method of acquiring transfer knowledge. In section 4, we describe how our text classifier is implemented and how transfer knowledge is used. In section 5, we introduce the results of our experiments and evaluation method. In section 6, we introduce some of our future work.

## 2 Related Works

### 2.1 Transfer Learning

Current research work in the field of transfer learning can be mainly divided into three parts. One is example-based transfer learning in homogenous feature space, one is feature-based transfer learning in homogenous space, another is transfer learning in heterogeneous feature space.

The main idea of example-based transfer learning is that, although the training data set is, to some extent, different from the testing set, there must exists some part of it that is suitable for training a reliable categorization model.

As reported in Pan and Yang's survey [1], many transfer learning methods have already proposed. For example, feature-representation-transfer approaches. In the field of feature-based transfer learning, scholars have proposed several algorithms for learning, such as CoCC [2], TPLSA [3], Spectral Domain-Transfer Learning [4], and self-taught clustering [5]. Its basic idea is that by clustering the target and training data simultaneously to allow the feature representation from the training data to influence the target data through a common set of features. Under the new data representation, classification on the target data can be improved.

Some scholars have proposed a method called translated learning [6,7] to solve the problem of using labeled data from one feature space to enhance the classification of other entirely different learning spaces. An important aspect of translated learning is to build a "bridge" to link one feature space (known as the "source space") to another space (known as the "target space") through a translator in order to transfer the knowledge from source to target. The translated learning solution uses a language model to link the class labels to the features in the source spaces, which in turn is translated to the features in the target spaces. Finally, this chain of linkages is completed by tracing back to the instances in the target spaces. Another proposed method is domain adaptation with structural correspondence learning, which plays a 'pivot' features in transfer learning [8].

## 2.2 Text Categorization

Text categorization aims at automatically classifying text documents into certain predefined categories and is an important technique in processing and organizing significant number of texts. It classifies a text into a most possible category by extracting its features and comparing them with those of the predefined categories and, in consequence, enhances the relationship among different texts.

The invention of text classifier can be traced back to the work of Maron in 1961. In 1970, Salton [9] proposed the VSM (Vector Space Model), which has become a classic model for text categorization. In 1990's, with the rapid development of the Internet and vast volume of texts on it, text classifier accordingly developed at a faster speed. A variety of text categorization methods appeared and the one which is based on machine learning has become a dominant one and achieves a very good result.

Most methods of text categorization come from pattern classification and can be divided into three categories. One is statistics-based, such as Naïve Bayes [10], KNN [11], Centroid-Based Model [12], Regression Model [13], SVM [14], and Maximum Entropy Model [15]; one is based on Neural Network Approach [16]; another is rules-based, such as decision-tree [17] and association rules [18]. And the method that is based on statistics is mainly studied and used nowadays.

## 3 Transfer Knowledge Acquisition

In order to challenge the conventional assumption of machine learning, and to take full advantage of the available data, we propose a novel transfer learning method, hoping to mine knowledge from available data, and applying this transfer knowledge to heterogeneous feature space, so as to help new tasks in new domains.

We learn keywords' syntactic and semantic use by quantifying their contextual information, such as part-of-speech, semantics, possibility of co-occurrence, and position to form transfer knowledge, so as to help cross-domain machine learning tasks.

This paper proposes a novel strategy to automatically acquire transfer knowledge from existing training data. Since automatic text classification is a typical task that involves machine learning, we use text classification tasks to testify the efficacy of our proposed method.

### 3.1 Knowledge Acquisition in Homogeneous Feature Spaces

#### Algorithm 1

Step1: corpus pre-processing

For any Chinese document D we do Chinese word segmentation, POS tagging, and then, we wipe off the word that can do little contribution to the linguistic knowledge, such as preposition, conjunction, auxiliary word and etc.

Step2: establish a temporary text for later processing.

Extract  $k$  keywords for the text by using TF-IDF method ( $k \leq 50$ ). And then, we extract all the sentences that contain these 50 keywords and form a temporary document  $D'$  for acquiring co-occurrence knowledge.

Step3: Calculate the co-occurrence distance.

For a single keyword in the document  $D'$ , we consider the keyword as the center, and respectively get the left-side and the right-side co-occurrence distance from keyword to its co-occurrence, which is calculated as (1) and (2) ( $m, n \leq 5$ ).

$$C_{li} = \left( \frac{1}{2} \right)^{i-1} B_l \quad (i=1,2, \dots, m) \quad (1)$$

$$C_{rj} = \left( \frac{1}{2} \right)^{j-1} B_r \quad (j=1,2, \dots, n) \quad (2)$$

In (1) and (2),  $B_l$  and  $B_r$  are the importance decay factor from keywords, which is calculated as (3). We consider the closer a word is to a keyword, the more important it is.

$$B_l = \frac{1}{\sum_{i=1}^m \left( \frac{1}{2} \right)^{i-1}} \quad B_r = \frac{1}{\sum_{j=1}^n \left( \frac{1}{2} \right)^{j-1}} \quad (3)$$

In (3),  $m$  and  $n$  represent the number of left and right number of words from a keyword.

Step4: accumulate co-occurrence distance.

For the  $i$ th co-occurrence of a keyword, we extract its part-of-speech POS, and position L. Then we regard the keyword and its  $i$ th co-occurrence ( $\text{co-occurrence}_i$ ,  $\text{POS}_i$ , L) as a relation pair, and  $C_i$  as its co-occurrence distance, which is calculated from L. To a single keyword, we accumulate all the  $C_i$  of the relation pairs which have the same  $\text{co-occurrence}_i$  and  $\text{POS}_i$  and, at the same time, record the accumulation times.

Step5: Calculate the average co-occurrence distance.

We calculate the average value of  $C_i$  that appears in corpus known as the average co-occurrence distance  $\bar{C}_i$  between the keyword and its co-occurrence ( $\text{co-occurrence}_i$ ,  $\text{POS}_i$ , L).

Step6: Build up transfer knowledge base.

When all of documents are learned, all keywords and their co-occurrence information ( $\text{co-occurrence}$ ,  $\text{POS}$ ,  $\bar{C}$ ) compose our transfer knowledge base.

Step7: Build index.

In order to improve the processing speed, for the acquired transfer knowledge base, we use such keys as (keyword, co-occurrence, POS) to build an index for our transfer knowledge base.

### 3.2 Knowledge Acquisition in Heterogeneous Feature Spaces

Acquiring transfer knowledge from heterogeneous feature spaces means to learn transfer knowledge from available data or knowledge base, and to apply it to different domains.

Traditional machine learning requires that training and testing data should be evenly distributed and in homogeneous feature space. If the feature space of testing data changes dramatically, or changes in domain, the training data fails. So we propose a method for acquiring transfer knowledge in heterogeneous feature space.

Our method is based on people's assumption of natural language: In a certain period of time, words in natural language basically remain their syntactic and semantic use, no matter in what circumstance and context. So in this paper, we assume that before processing documents, we can acquire some keywords that can represent their categories from domain experts. For these keywords, we then acquire transfer knowledge for their respective categories from the existing knowledge base.

### Algorithm 2

Step1, to simulate the process of acquiring keywords from domain experts, we extract 50 keywords from each category by using TF-IDF methods.

Step2, check if these keywords exist in existing knowledge bases, extract the part-of-speech and co-occurrence distance of their co-occurrences, and form such relation pair (keyword, POS, L). For each keyword, we calculate the times of the same POS, and, at the same time, accumulate co-occurrence distance.

Step3, we divide the accumulated co-occurrence distance by its times and regard this value as the average co-occurrence distance  $\bar{C}$ . For every keyword and all its co-occurrence's part-of-speech, we consider the relation pair (keyword, POS,  $\bar{C}$ ) as the transfer knowledge of this keyword; all the above mentioned relation pairs of in a category compose the transfer knowledge bank for this category.

Step4, In order to improve the processing speed, for the acquired transfer knowledge bank, we use such keys as (keyword, POS) to build an index for our transfer knowledge bank.

## 4 Application of Transfer Knowledge

In order to test the feasibility and effectiveness of our transfer learning method, we apply our method to automatic text categorization, which is a typical use in machine learning. We firstly extract transfer knowledge from each category and form a transfer knowledge base according to Algorithm 1 and Algorithm 2. And then we use these transfer knowledge bases as the text classifier. When we have a document to be classified, we will calculate its possibility of belonging to one category based on each category's transfer knowledge base, and choose the category with the highest possibility as the document's final category. Algorithm 3 describes the process of text classification tasks in homogeneous feature space, and Algorithm 4 describes the process of text classification tasks in heterogeneous feature space.

## 4.1 Text Classification in Homogeneous Feature Space

### Algorithm 3

Step1: Documents Pre-processing.

For any Chinese document D, we do Chinese word segmentation, POS tagging and then, wipe off the word that can do little contribution to the linguistic ontology knowledge, such as preposition, conjunction, auxiliary word and etc. Next, we extract k keywords from this document by using the TF-IDF method, and regard these k keywords as the features of this document;

Step2: Get the average co-occurrence distance from transfer knowledge base.

For the  $i$ th keyword  $\text{Keyword}_i$  in document D and its relation pair  $\langle \text{Keyword}_i, (\text{CoexistWord}, \text{POS}) \rangle$ , we search them in the  $j$ th transfer knowledge base we constructed according to Algorithm 1 and add up the average co-occurrence distance, and the result is known as  $C_j^i$ ;

Step3: Calculate the possibility for a document in the  $j$ th category.

Repeat step 2 until  $k$  keywords in document D are calculated. The possibility of belonging to the  $j$ th category is calculated as (4), known as  $\text{Eval\_}D_j$ .

$$\text{Eval\_}D_j = \sum_{i=1}^k C_j^i \quad (4)$$

Step4: Choose a category for documents

After we have calculated the document's possibility of belonging to each category, we put the document into the category with the highest possibility as (5).

$$\text{Eval\_}D = \text{Max}(\text{Eval\_}D_j) \quad (5)$$

In which  $j$  represents the total number of categories of training document.

## 4.2 Text Classification in Heterogeneous Feature Space

### Algorithm 4

Step1: Documents Pre-processing.

This process is similar to the Step1 in Algorithm 3;

Step2: Get the average co-occurrence distance from transfer knowledge base.

For the  $i$ th keyword  $\text{Keyword}_i$  in document D and its relation pair  $\langle \text{Keyword}_i, (\text{POS}) \rangle$ , we search them in the  $j$ th transfer knowledge bank we constructed according to Algorithm 2 and add up the average co-occurrence distance, and the result is marked as  $C_j^i$ ;

Step3: Get the possibility value for a document in the  $j$ th category.

This process is similar to the Step3 in Algorithm 3;

Step4: Decide the final category for a document.

This process is similar to the Step4 in Algorithm 3;

## 5 Experiments and Analysis

### 5.1 Experimental Data and Evaluation Methods

We choose three Chinese data sets to evaluate our method of transfer learning. One is 863 corpus (2003), one is 863 corpus (2004), another is Tan corpus [19, 20]. In this paper, we design four groups of experiments to test our method.

In the first group of experiments, we aim to test the effectiveness of our transfer learning method when it is used in homogeneous feature space. We use the 863 corpus (2003) as the training data set, acquiring transfer knowledge and constructing transfer knowledge base according to Algorithm 1, and then using this knowledge base to help classify text documents in the 863 corpus (2004). Since the 863 corpus (2003) and the 863 corpus (2004) consist of the same categories of documents, and they are evenly distributed, so we can consider these two date sets in homogeneous feature space.

In the second group of experiments, although we still test the effectiveness of our transfer learning method when it is used in homogeneous feature space, the documents in training and testing data set are unevenly distributed and come from different sources. In the experiment, we choose the same five categories from both 863 corpus (2003) and Tan corpus, and use the former date set as the training data, and the latter as the testing data set. Although unevenly distributed, these documents belong to the same categories, so we still consider them under homogeneous feature space.

In the third experiment, we aim at testing the effectiveness of our transfer learning method itself. We choose our training and testing data set from the same corpus, and do three-cross validation. We test our method in Tan corpus and the 863 corpus respectively.

In the last experiment, we try to test the effectiveness of transfer learning method among heterogeneous feature spaces. We focus on how to take full advantage of the existing knowledge base when testing data is greatly changed in category or there is no training data available. By acquiring transfer knowledge from the 863 corpus according to Algorithm 2, and applying it to 20 categories in Tan corpus which do not in the 863 corpus, we test our transfer learning method in heterogeneous feature space. In this experiment, the 20 selected categories from the 863 corpus which are used as training data set consist of politics and laws, philosophy, economy, literature, art, biology, architecture, transportation and another 12 categories. However, the 20 categories selected from Tan corpus which are to be used as testing data set consist of fortune, computer science, house, automobile, basketball, health, decoration and another 13 categories.

In the first three experiments, we select 20 categories that contain approximately the same number of documents from the 863 corpus (2003) as the training data set, and select the corresponding 20 categories from the 863 corpus (2004) as the testing data set. We also choose the 5 categories which also exist in the 863 corpus and 20 random categories from Tan corpus. In the last experiment, we use all categories in the 863 corpus as the training data set, and select the non-exist 20 categories in the 863 corpus from Tan corpus as the testing data set.

In the evaluation phase, we use MacroF1 and MicroF1 values to measure the performance of our classifier.

## 5.2 Test from the Same Source in Homogeneous Feature Space

We select the 863 corpus (2004) as the testing data set, and the twenty categories selected are the same as those selected from the 863 corpus (2003). The result of transfer learning is as shown in Table 1.

**Table 1.** Result of transfer learning from the same source corpus (%)

Testing data	MacroF1	MicroF1
863 corpus (2004)	73.0	71.2

Table 1 shows that MacroF1 and MicroF1 are to some extent lower than the best evaluation result in previous years, and it is due to the low value of “History and Geography”, “Economy” and “Art” that directly influence the final result. After careful analysis of their knowledge bases, we find the reason is that the keywords selected from training documents cannot precisely represent their categories and cannot cover their whole keywords.

So, to examine the influence of different keyword selection to our categorizing result, we manually add some words into our list of stop words to make the selection of keywords more precise. We manually add some verbs, adjectives and adverbs such as “提” ‘Propose’, “good”, “high”, “express”. Then, we get a new transfer knowledge base, and do the categorization again. Table 2 shows how the categorization result changes after improving the precision of keywords in our experiment.

**Table 2.** Result after improving the precision on keyword selection (%)

Testing data	MacroF1	MicroF1
863 corpus (2004)	75.7	74.8

Although the precision and recall value of some categories decline slightly, Table 2 shows that the overall precise and recall value increase. Especially those whose precision and recall value are low before our changing increase dramatically, and the most prominent increase reaches about 28%. Also, MacroF1 and MicroF1 both increase about 3%, which almost reach the best evaluation result in previous years. From this result, we can see that if keywords are chosen properly, they can have a boosting effect on the efficacy of our transfer learning method.

## 5.3 Test from Different Source in Homogeneous Feature Space

In order to test the effectiveness of transfer knowledge and the proposed method in this paper, we use the corpus that are collected and organized by Dr. Songbo Tan in China Academy of Science Institute of Computing. Five categories in Tan corpus are the same or similar to the 863 corpus. Table 3 shows these five categories and the result of directly transferring the knowledge acquired in the 863 corpus (2003) to Tan corpus.

**Table 3.** Result from different source in homogeneous feature space (%)

Categories in the 863 corpus	Categories in the Tan Corpus	Precision	Recall	F1
Economy	Finance	84.7	99.3	91.4
Art	Aesthetics and Art	65.5	85.7	74.3
Astronomy, Earth Science	Astronomy	88.1	87.6	87.8
Literature	Literature and Art	94.2	74.5	83.2
Medicine, Health	Medicine	99	86.9	91.5
Macro-F1		86.5		
Micro-F1		88.1		

Table 3 shows that the MacroF1 and MicroF1 values of directly transferring knowledge acquired from 863 corpus to Tan corpus closely approximate the categorization result given by Dr. Songbo Tan, who conducted categorization tasks over Tan corpus in several traditional ways, which are shown in Table 4. However, the result of “Art” in 863 corpus (“Aesthetics and Art” in Tan corpus) and “Literature” in 863 corpus (“Literature and Art” in Tan corpus) is comparatively low. This is caused by the difference in the source of collecting documents in two corpuses. For example, “Aesthetics and Art” can be divided into “Aesthetics” and “Art”; and “Literature and Art” can be divided into “Literature” and “Art”. However, the overall result is satisfying.

#### 5.4 Test of the Effectiveness of Transfer Knowledge Itself

To testify the effectiveness of the method of acquiring transfer knowledge, we choose Tan corpus to do cross validation. First, we still choose the previous 5 categories to do 3-cross validation, aiming to compare the result with the result shown in Table 3 in the previous section. Then, we randomly choose 20 categories in Tan corpus and do 3-cross validation, aiming to examine the effectiveness of our method at the macro level.

Firstly, we present the classification results done by Dr. Songbo Tan. He measured his corpus in five different ways including Centroid-Based, KNN, Winnow, Bayes and SVMTorch. The results are shown in Table 4.

**Table 4.** Baseline result provided by Dr. Songbo Tan (%)

Items	Centroid Based	KNN	Winnow	Bayes	SVM Torch
MacroF1	0.8632	0.8478	0.7587	0.8688	0.9172
MicroF1	0.9053	0.9035	0.8645	0.9157	0.9483

Then, we do 3-cross validation over previous 5 categories in Tan corpus. The result is as shown in Table 5.

**Table 5.** Test of 3-cross validation over 5 categories in Tan corpus (%)

Corpus	Macro-F1	Micro-F1
Tan corpus (5 categories)	96.1	97.5

By comparing the result shown in Table 3 and Table 5, we can see that the result of 3-cross validation over Tan corpus is 9~10% higher than transferring knowledge from 863 corpus to Tan corpus. And there are mainly two reasons for this discrepancy. One is caused by the difference in collecting documents in two corpuses. And the other is caused by the difference in the number of test documents.

Next, to further prove the effectiveness of our text categorization method, we randomly choose 20 categories in Tan corpus to do 3-cross validation. And the result is as shown in Table 6.

**Table 6.** Test of 3-cross validation over 20 categories in Tan corpus (%)

Corpus	Macro-F1	Micro-F1
Tan corpus (5 categories)	89.6	91.0

Table 6 shows that the Micro-F1 and Macro-F1 value of our method of acquiring transfer knowledge both reaches about 90% when it is applied to categorization tasks over one corpus. By comparison between Table 4 and Table 6, we can see that our result is satisfying and thus testify the efficacy of the transfer knowledge acquisition strategy proposed in this paper.

## 5.5 Weakly-Supervised Transfer Learning

To challenge the conventional assumption in machine learning that training data set and testing data set being in homogeneous feature space, and to testify the effectiveness of Algorithm 2, we acquire transfer knowledge from the 863 corpus (2003) and apply it to the 20 categories in Tan corpus but not in the 863 corpus. This is to construct a transfer knowledge base from unrelated domains so that we can meet the classification requirement in new text domains.

The 863 corpus and Tan corpus we use in this paper differ greatly in contents and publishing time, so we can regard the knowledge base we construct from the 863 corpus as an outdated knowledge base or cross-domain knowledge base. We firstly acquire transfer knowledge from the 863 corpus (2003) and construct a knowledge base according to Algorithm 1, and then extract 50 keywords from each category in Tan corpus by using methods like TF-IDF, and considering these keywords as the pre-knowledge for each category. Then, we form a new transfer knowledge base based on the previously constructed knowledge base. Finally, we build up our text classifier by extracting 50 keywords and their co-occurrence information from the testing data set according to Algorithm 4. Table 7 shows the result of transfer learning among heterogeneous feature spaces.

**Table 7.** Test of Transfer knowledge among heterogeneous feature spaces (%)

Testing data	MacroF1	MicroF1
Tan corpus	80.1	87.6

We can see from Table 7 that MicroF1 is comparatively higher. This means that the precision of our experiment is high on the whole. In contrast, the value of MacroF1 is comparatively low, which indicates that there is much difference in the value of precision and the value of recall. Also, we find that the precision of several categories, such as “Psychology”, “Publication”, “Job hunting”, are much lower than others. After review their transfer knowledge base, we find that the keywords extracted by TF-IDF could not well represent their categories respectively; and that these three categories differ so greatly from categories in the 863 corpus that it is too difficult to extract transfer knowledge for them.

To further test the effectiveness of our transfer knowledge base, we increase the number of keywords extracted in Algorithm 2 from 50 to 100. Accordingly, when building up our text classifier in Algorithm 4, we also increase the number of keywords from 50 to 100, so that we can add more information into our transfer knowledge base. The result of transfer learning after the expansion of transfer knowledge base is shown as in Table 8.

**Table 8.** Test of transfer learning after increasing transfer knowledge base (%)

Testing data	MacroF1	MicroF1
Tan corpus	81.6	88.5

By comparing the result of Table 7 and Table 8, we can see that the value of precision of “Fortune” and “Publication” increases by approximately 10%, and other categories also increase in some degree. Although the values of MacroF1 and MicroF1 do not increase much, we can still draw the conclusion that the expansion of our transfer knowledge base has positive effect on the result of text categorization.

In order to further test the effect of expansion transfer knowledge base, we acquire transfer knowledge from both the 863 corpus (2003) and the 863 corpus (2004). The result of transfer learning after the expansion of transfer knowledge base is shown as in Table 9.

**Table 9.** Test of transfer learning after expanding transfer knowledge base (%)

Testing data	MacroF1	MicroF1
Tan corpus	81.4%	88.6%

By comparing TABLE 8 and TABLE 9, we find that although the value of precision and recall of each category changes in some degree, the value of MacroF1

and MicroF1 remains unchanged. This is because, firstly, there is not much difference between 863 corpus in the year of 2003 and 2004, so adding the 863 corpus (2004) does not add more useful features into our transfer knowledge base; and secondly, the knowledge extracted from the 863 corpus (2003) is already enough for us to obtain the linguistic information for our pre-knowledge, so adding the 863 corpus in the year of 2004 does not help much to our classification result. Still, the result of our experiment is satisfying. And this indicates that the transfer learning method proposed in this paper which aims at solving the difficulty in constructing cross-domain knowledge base is very effective.

## 6 Conclusions

In this paper, we present a novel strategy for acquiring transfer knowledge and apply it to automatic text categorization tasks among homogeneous and heterogeneous feature spaces. By conducting experiments across different corpuses and different domains, we get a satisfying outcome, which testifies the effectiveness of our method. However, in our methods, we only extract some basic linguistic information. So our future work may involve: (1) try to add more linguistic information to our method of acquiring transfer knowledge; (2) apply key technique in our method to public test set to further examine its efficacy. And actively explore other strategies for acquiring transfer knowledge as well as transfer learning methods.

**Acknowledgments.** This work is supported by the national natural science foundation of China (No. 61073130) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207).

## References

- [1] Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10), 1345–1359 (2010)
- [2] Dai, W., Xue, G.-R., Yang, Q., Yu, Y.: Co-clustering based Classification for Out-of-domain Documents. In: *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, San Jose, California, USA, August 12-15, pp. 210–219 (2007)
- [3] Xue, G.-R., Dai, W., Yang, Q., Yu, Y.: Topic-bridged PLSA for Cross-Domain Text Classification. In: *Proceedings of the Thirty-first International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR 2008)*, Singapore, July 20-24, pp. 627–634 (2008)
- [4] Ling, X., Dai, W., Xue, G.-R., Yang, Q., Yu, Y.: Spectral Domain-Transfer Learning. In: *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, Las Vegas, Nevada, USA, August 24-27, pp. 488–496 (2008)
- [5] Dai, W., Yang, Q., Xue, G.-R., Yu, Y.: Self-taught Clustering. In: *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML 2008)*, Helsinki, Finland, July 5-9, pp. 200–207 (2008)

- [6] Dai, W., Chen, Y., Xue, G.-R., Yang, Q., Yu, Y.: Translated Learning: Transfer Learning across Different Feature Spaces. *Advances in Neural Information Processing*
- [7] Ling, X., Xue, G.-R., Dai, W., Jiang, Y., Yang, Q., Yu, Y.: Can Chinese Web Pages be Classified with English Data Source? In: *Proceedings the Seventeenth International World Wide Web Conference (WWW 2008)*, Beijing, China, April 21-25, pp. 969–978 (2008)
- [8] Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pp. 120–128 (2006)
- [9] Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* 24(5), 513–523 (1988)
- [10] Lewis, D.D.: Naïve(Bayes) at forty: The Independence Assumption in Information Retrieval. In: Nédellec, C., Rouveirol, C. (eds.) *ECML 1998. LNCS*, vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
- [11] Yang, Y.M., Liu, X.: A Re-examination of Text Categorization Methods. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, pp. 42–49 (August 1999)
- [12] Han, E., Karypis, G.: Centroid-Based Document Classification Analysis & Experimental Result. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) *PKDD 2000. LNCS (LNAI)*, vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
- [13] Yang, Y.M.: An evaluation of statistical approaches to text categorization. *Information Retrieval* 1(1), 76–88 (1999)
- [14] He, J., Tan, A.H., Tan, C.L.: A Comparative Study on Chinese Text Categorization Methods. In: *PRICAL 2000 Workshop on Text and Web Mining*, Melbourne, pp. 24–35 (August 2000)
- [15] Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: *Proceedings of the IJCAI 1999 Workshop on Information Filtering*, Stockholm, Sweden (1999)
- [16] Wiener, E.: A neural network approach to topic spotting. In: *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR 1995)*, Las Vegas, NV (1995)
- [17] Apte, C., Damerau, P., Weiss, S.: Text mining with decision rules and decision trees. In: *Proceedings of the Conference on Automated Learning and Discovery Workshop 6: Learning from Text and the Web* (1998)
- [18] Lent, B., Swami, A., Widom, J.: Clustering association rules. In: *Proceedings of the Thirteenth International Conference on Data Engineering (ICDE 1997)*, Birmingham, England (1997)
- [19] Tan, S., Wang, Y.: Chinese Text Categorization Corpus-TanCorpV1.0., <http://www.searchforum.org.cn/tansongbo/corpus.html>
- [20] Tan, S., et al.: A Novel Refinement Approach for Text Categorization. In: *ACM CIKM* (2005)

# Fuzzy Combinations of Criteria: An Application to Web Page Representation for Clustering\*

Alberto Pérez García-Plaza, Víctor Fresno, and Raquel Martínez

NLP & IR Group, UNED, Madrid, Spain

{alpgarcia,vfresno,raquel}@lsi.uned.es

**Abstract.** Document representation is an essential step in web page clustering. Web pages are usually written in HTML, offering useful information to select the most important features to represent them. In this paper we investigate the use of nonlinear combinations of criteria by means of a fuzzy system to find those important features. We start our research from a term weighting function called Fuzzy Combination of Criteria (*fcc*) that relies on term frequency, document title, emphasis and term positions in the text. Next, we analyze its drawbacks and explore the possibility of adding contextual information extracted from inlinks anchor texts, proposing an alternative way of combining criteria based on our experimental results. Finally, we apply a statistical test of significance to compare the original representation with our proposal.

**Keywords:** web page, representation, fuzzy logic, clustering.

## 1 Motivation

Document representation is an essential step in web page clustering. The most common approach consists in trying to capture the importance of the words in the document by means of term weighting functions. Most of these functions work following the Vector Space Model (VSM) [11] and among them, tf-idf is one of the most widely used. This function works with plain text, but does not exploit other additional information that some kind of documents contain.

In order to determine the words that better represent document contents, one of the initial hypothesis of present work is that a good representation should be based on how humans read documents. We usually search for visual clues used by authors to capture our attention as readers.

The HTML tags provide additional information about those visual clues that can be employed to evaluate the importance of document terms in addition to term frequency. Regarding the way of combining different criteria within the VSM, probably the most straightforward way is a linear combination of heuristic criteria like the Analytical Combination of Criteria (*acc*) [4]. These criteria

---

\* The authors would like to thank the financial support for this research to the Spanish research projects MA2VICMR (S2009/TIC-1542) and Holopedia: the automatic encyclopedia of people and organizations (TIN2010-21128-C02).

are extracted from both text reading and writing processes, allowing to set different weights for each criterion. Its main drawback comes from the problem of nonlinearity in the combination of criteria, in other words, the fact that the contribution of one criterion can depend on the rest of the criteria: when a term is important in a single criterion, e.g. in title, the corresponding component will have a value which will always be added to the importance of the term in the document, regardless of the importance of the rest of the components.

To solve this issue we need a system that allows to define related conditions to establish term importance, e.g., a term should appear in the title and emphasized in the document to be considered important, in order to avoid rhetoric titles where words are not representative for the document topic. Because of this, we are interested in nonlinear combinations of criteria. In this context, [3] and [10] presented a document representation based on a fuzzy combination of heuristic criteria (*fcc*). The framework they presented is used here as a starting point to explore the possibilities of these systems to help apply expert knowledge and combine criteria in a nonlinear fashion. As a result, we present a representation resulting of our findings, showing significant improvements over *fcc*.

The remainder of this paper is organized as follows. In Section 2 we summarize related works. Section 3 presents *fcc* web page representation. Experiments to study how to improve *fcc* and to add contextual information to our representation are performed in Section 4. Finally, empirical evaluation is performed in Section 5, concluding the paper in Section 6.

## 2 Background

Most of document representation approaches are based on the VSM, where each document is represented as a vector, and each vector component corresponds to a value which tries to express the importance of the term in the document. These components are also called features, and their value is called feature or term weight. One of the most widely used functions to calculate term weight is tf-idf, that combines term frequency in a document with document frequency of the same term. On one hand, this representation does not take into account the additional information one can find in web pages, just plain text, and, on the other hand, it is worth to notice the fact that, in IR, field where tf-idf was defined, the goal is to find differences instead of similarities.

Some researchers have presented new representations based on variations of tf-idf. In [5] the authors propose to employ keyphrases instead of words, introducing some changes like rewarding instead of penalizing keyphrases that appears in many documents and having into account whether or not they appear in titles or headers by means of a linear combination, but they neither specify the exact weights for each component nor the way of calculating it. In [8] the authors consider that document title, textual content and anchor texts have different importance levels and decide to represent each one with a separate tf-idf feature vector. This requires a particular clustering algorithm so it was not compared to other representations, but with other algorithms. This model does not allow to include new criteria to the representation without changing the whole system

(input format and algorithm). Their results show only average precision, but including recall could lead to different conclusions.

With the same objective, [3] presents a self-content representation for web pages. It is called Fuzzy Combination of Criteria (*fcc*) and has been successfully applied in clustering and classification, where it has been compared with different state of the art alternatives, like *acc* or *tf-idf*, obtaining good results. The main difference with the above mentioned work is that *fcc* keeps the VSM as it is. To do this, the proposed weighting function uses a fuzzy system to heuristically combine criteria. Concretely, four criteria are used: term frequency, term frequency in title, term frequency in emphasis and term positions in the document. Besides, the fuzzy logic engine provides the possibility of adding new criteria and modify the rules easily, which allows to study the contribution of each criterion. For these reasons, in addition to the discussion about linear and nonlinear combinations detailed in Section 1, we have chosen the framework offered by *fcc* to develop our work. Among the fields explored by previous works on *fcc*, the most promising results were achieved in clustering tasks, reason why in this work we will focus our research in this field, using *fcc* as a baseline.

Another alternative to enrich web page representation is adding some kind of links information. In [15] a study about how to combine textual content and link analysis is performed. They use inlinks and outlinks in order to improve clustering applied to search results. Their empirical results suggest that the combination of both, textual content and links can improve web page clustering. About using anchor texts, [9] gives some interesting ideas. They state that anchor texts contribute meaningful information for IR tasks, but this information is not as good to capture the aboutness of web documents. They agree with [2] that anchor text terms are similar to terms used in search queries. Besides, these terms are not often in web page contents, concretely in [9] they found only 51% of the cases, while in [2] they found 66.4% of terms appearing on both. Anchor texts are a lightweight and efficient alternative compared to other more complex methods of anchor context extraction.

In present work we study web page representation by means of fuzzy combinations of heuristic criteria, analyzing the contribution of each criterion to improve clustering results. We also explore not self-content information like anchor texts to extend the combination.

### 3 Fuzzy Combination of Criteria

For a human reader, title and emphasized words in a text document have a bigger role than the rest of the document in understanding its main topic. Moreover, the beginning and the end of the body text usually contain overviews, summaries or conclusions with essential vocabulary. The goal of *fcc* [10] is to define the importance level of each word in a document by using a set of heuristic criteria: word frequency counts in titles, emphasized text segments, in the beginning and the end of the document, and in the whole document. As titles and other special texts are encoded with HTML tags, a subset of those tags are used in *fcc* in order to collect “the most important” words in a document.

The fuzzy system is built over the concept of linguistic variable. Each variable describes the membership degree of a word to a particular class. The variables are defined by human experts. The fuzzy system knowledge base is defined by a set of IF-THEN rules that combine the variables. The aim of the rules is to combine one or more input fuzzy sets (antecedents) and to associate them with an output fuzzy set (consequent). Once the consequents of each rule have been calculated, and after an aggregation stage, the final set is obtained.

The *fcc* IF-THEN rules are based on the following ideas: (1) If a word appeared in the title or the word was emphasized, that word should also appear in one of the other criteria in order to be considered important. (2) Words appearing in the beginning or at the end of a document may be more important than the other words, because documents usually contain overviews and summaries in order to attract the interest of the reader. (3) If a word is not emphasized, it is possible that there are no emphasized words in the document at all. (4) If a word does not appear in the title, it is possible that the document does not have a title at all, or the title does not contain important words. (5) If the previous criteria were not able to choose the most important words, the frequency counts may help to find them. The knowledge base for *fcc* is shown in Table 1. Each row has the values of different criteria and the resulting output, called 'Importance'. The inference engine that evaluates the fired rules is based on the *center of mass* (COM) algorithm that weights the output of every fired rule, taking into account the truth degree of its antecedent. The output is a linguistic label (e.g., 'Low', 'Medium', 'Very High') with an associated number related to the importance of a word in the document, and it is calculated by scaling the membership functions by product and combining them by summation. These kind of systems are called

**Table 1.** Rule base for *fcc*. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
	High		High		High			$\Rightarrow$	Very High
	High		Medium		High			$\Rightarrow$	Very High
	High		High		Medium			$\Rightarrow$	Very High
	High		Medium		Medium			$\Rightarrow$	High
	Low		Low		Low			$\Rightarrow$	No
	High		High		Low	Preferential		$\Rightarrow$	Very High
	High		High		Low	Standard		$\Rightarrow$	High
	High		Medium		Low	Preferential		$\Rightarrow$	Medium
	High		Medium		Low	Standard		$\Rightarrow$	Low
	High		Low		High	Preferential		$\Rightarrow$	Very High
	High		Low		High	Standard		$\Rightarrow$	High
	High		Low		Medium	Preferential		$\Rightarrow$	High
	High		Low		Medium	Standard		$\Rightarrow$	Medium
	High		Low		Low	Preferential		$\Rightarrow$	Medium
	High		Low		Low	Standard		$\Rightarrow$	Low
	Low		High		High	Preferential		$\Rightarrow$	Very High
	Low		High		High	Standard		$\Rightarrow$	High
	Low		High		Medium	Preferential		$\Rightarrow$	High
	Low		High		Medium	Standard		$\Rightarrow$	Medium
	Low		High		Low	Preferential		$\Rightarrow$	Medium
	Low		High		Low	Standard		$\Rightarrow$	Low
	Low		Medium		High	Preferential		$\Rightarrow$	Very High
	Low		Medium		High	Standard		$\Rightarrow$	High
	Low		Medium		Medium	Preferential		$\Rightarrow$	Medium
	Low		Medium		Medium	Standard		$\Rightarrow$	Low
	Low		Medium		Low	Preferential		$\Rightarrow$	Low
	Low		Medium		Low	Standard		$\Rightarrow$	No
	Low		Low		High	Preferential		$\Rightarrow$	High
	Low		Low		High	Standard		$\Rightarrow$	Medium
	Low		Low		Medium	Preferential		$\Rightarrow$	Medium
	Low		Low		Medium	Standard		$\Rightarrow$	Low

additive [7] and their main advantage is the efficiency of the computation. A more detailed explanation of the fuzzy system can be found in [3,10].

## 4 Proposing a New Combination

In this section we will use the framework offered by *fcc* to further investigate about how information extracted from HTML documents can improve document clustering. Each subsection is based on the previous ones in order to build our representation proposal step by step. Some experimental settings will be the same for all the experiments, so they are outlined hereafter.

**Experimental Settings.** In preprocessing, a stop word list was used to remove common words. The punctuation was also removed. Suffixes were removed using a standard implementation of the Porter's algorithm for English. Regarding the clustering process, we chose Cluto rbr (k-way repeated bisections globally optimized) as a state of the art algorithm [6]. It is a widely used algorithm with good results in the literature [1,3,13]. Algorithm parameters were set by default.

After weighting terms, we reduced document vectors to 100, 500, 1000, 2000, and 5000 dimensions using two methods: Most Frequent Terms until  $n$  level (*mft*) and Latent Semantic Indexing (*lsi*). The *mft* method works as follows: first ranks the terms in each document based on the term weighting function values. Then, terms on the first position in the document rankings are put in order according to how many times they have appeared in the rankings. If two or more terms appear the same number of times in different rankings, we put them in order based on the maximum weight found for each of them. Next we take the terms appearing in the second position in the rankings, and so forth. The process stops when the desired number of terms is reached. Notice that by following this algorithm the resulting list may be larger than the required size, because there are as many rankings as documents in the dataset. Nevertheless, as we put the list in order, we can get the exact number of terms just taking the first  $n$  terms. Regarding *lsi*, as suggested by [14], it was applied after a previous reducing step to alleviate its computational complexity. We reduced vector dimension using *mft* from the original size to 5000 features before applying *lsi*.

To evaluate the clustering quality for clustering algorithms, typically the F-Measure (equation 1) is used [12], which is equal to the harmonic mean of recall and precision. The overall F-measure is the weighted average of the F-measure for each category:

$$F(i, j) = \frac{2 \cdot \text{Recall}(i, j) \cdot \text{Precision}(i, j)}{\text{Recall}(i, j) + \text{Precision}(i, j)} \quad ; \quad F = \sum_i \frac{n_i}{n} \cdot \max_j \{F(i, j)\} \quad (1)$$

where  $i$  is the category,  $j$  the cluster and  $n$  the number of documents. The F-measure values are in the interval (0,1) and larger values correspond to higher clustering quality.

**Datasets.** In previous work [3] two different datasets were used: Banksearch and Webkb. Because of this, we decided to use these datasets to obtain comparable

results. Banksearch contains 11,000 documents divided in 11 categories of equal size, divided in two hierarchy levels: 10 main categories at the same level and another one parent of two of them. Our experiments are not oriented to hierarchical clustering, so we use the 10 main categories, corresponding to 10,000 documents. In Webkb we removed 'others' category because introduced noise, resulting in 6 categories for a total of 4,518 documents. Webkb categories are unbalanced with respect to the number of documents in each category (3% of documents in the smallest category, 35% of documents in the biggest one).

#### 4.1 How Does Dimension Reduction Affect Weighting Function?

In order to explore the effect of dimension reduction techniques over the term weighting function we decided to use tf-idf, because it is a standard in clustering, and *fcc*, which will be our baseline.

**Table 2.** F-measure results for dimension reduction experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
<b>Banksearch</b>							
tf-idf mft	0,703	0,737	0,768	<b>0,772</b>	0,758	0,748	0,028
tf-idf lsi	0,750	0,755	0,756	0,757	0,763	0,756	<b>0,005</b>
fcc mft	0,723	0,757	0,768	0,765	<b>0,768</b>	0,756	0,019
fcc lsi	<b>0,775</b>	<b>0,763</b>	<b>0,785</b>	0,763	0,758	<b>0,769</b>	0,011
<b>Webkb</b>							
tf-idf mft	0,385	0,438	0,466	0,498	<b>0,513</b>	0,460	0,051
tf-idf lsi	<b>0,516</b>	<b>0,507</b>	<b>0,505</b>	<b>0,506</b>	0,501	<b>0,507</b>	<b>0,006</b>
fcc mft	0,453	0,472	0,475	0,468	0,475	0,469	0,009
fcc lsi	0,449	0,460	0,473	0,474	0,475	0,466	0,011

Table 2 shows the F-measure results for all the combinations of weighting functions and dimension reduction techniques for both datasets. Each table row contains F-measure values corresponding to the clustering solution obtained by using the representation specified in the first column with the number of features per vector detailed on top of the remaining columns, being Avg. and S.D. the average and the standard deviation for that row.

Between *lsi* and *mft*, results are different depending on the term weighting function. For tf-idf, *lsi* always improves *mft* in Banksearch when vector size is small (100 and 500 features). However, with 1,000 and 2,000 features *mft* obtained higher results, and with 5,000 the difference is < 1%. In Webkb occurs something similar, but the difference also appears with 1,000 features. As *mft* strongly depends on the term weighting function to select the most important terms, an improvement of *lsi* over *mft* implies that the weighting function is not working as well as it could. Looking at *fcc*, *lsi* does not improve its results in Webkb, except in one case, but the difference is < 1%. In Banksearch *lsi* improves *mft* when reducing to 1,000 or less features, and only in 2 cases the difference between them is > 1%. Comparing both functions, while *fcc* outperforms tf-idf in Banksearch, in Webkb the best results correspond to tf-idf helped by *lsi*.

Our hypothesis is that the improvement obtained by *lsi* over *mft* is a consequence of the term weighting function, because *lsi* is a feature transformation technique that could allow to discover relations among features, removing those

less representative. Therefore, if we are able to choose the most representative features of each document, *mft* should work, at least, similar to *lsi*.

## 4.2 Analysis of the Combination of Criteria

Section 4.1 left two open issues: to improve the bad performance of *fcc* in Webkb dataset, and to validate our hypothesis. Both of them are clearly related because if we improve the weighting function, our hypothesis says that the new results should be more similar to those obtained using *lsi*. In this section we perform a comprehensive study about how to improve *fcc* for document clustering.

**Study of Individual Criteria.** The first step is to analyze the contribution of each criteria in order to find any clue about why the combination does not perform in Webkb as well as in Banksearch. To do this, we repeat the clustering process modifying the combination of criteria proposed by *fcc*. We did four variations of this function, one per each criterion, in such a way that the output of the system will correspond only to one criterion at a time. We used *mft* reduction because it does not transform features, allowing us to study the effectiveness of each alternative to give more importance to the most representative terms.

**Table 3.** F-measure results for criteria analysis experiments

Rep.\Dim.	100	500	1000	2000	5000
<b>Banksearch</b>					
<b>fcc mft</b>	<b>0,723</b>	<b>0,757</b>	<b>0,768</b>	<b>0,765</b>	<b>0,768</b>
<b>title</b>	0,626	0,646	0,632	0,634	0,639
<b>emphasis</b>	0,586	0,671	0,674	0,685	0,693
<b>frequency</b>	0,689	0,715	0,720	0,724	0,731
<b>position</b>	0,310	0,525	0,538	0,599	0,608
<b>Webkb</b>					
<b>fcc mft</b>	<b>0,453</b>	<b>0,472</b>	<b>0,475</b>	0,468	0,475
<b>title</b>	0,432	0,433	0,404	<b>0,488</b>	0,479
<b>emphasis</b>	0,415	0,431	0,433	0,465	<b>0,489</b>
<b>frequency</b>	0,441	0,460	0,460	0,468	0,446
<b>position</b>	0,301	0,283	0,317	0,281	0,286

Table 3 shows the results of each individual criterion compared to *fcc*. Focusing on Banksearch results, values corresponding to *fcc* are always higher than individual ones. This means that the combination contributes to improve the results over individual criteria in all cases. Besides, frequency obtains the best values, while position obtains the worst ones. Webkb results are quite different. On one hand, frequency is not always the best among individual criteria and, on the other hand, *fcc* does not always outperform individual criteria, concretely title, emphasis or frequency have higher F-measure values in some cases when reducing vector dimension to 2000 and 5000 features. It seems that frequency strongly affects results, and going further, when title and emphasis could lead to a better clustering, their combination with frequency makes results worse. Therefore, while frequency gets higher results than the other criteria the combination works fine, but when titles or emphasis outperforms frequency, the combination does not work as good as it could. Thus, frequency is very important for a good grouping, as well as title and emphasis, and all of them should be very important

in the combination. However, position is the criterion with the worst results in all cases, so we have to take care using it to establish the importance of a term.

**Improving the Fuzzy Combination of Criteria.** In *fcc* rules (Table 1), when frequency is 'low' output can be 'very high' (the maximum) depending on the position, if title and emphasis are high. As we saw before, frequency contributes to a good clustering much more than position, so the output should reflect that fact, but in this case frequency is totally ignored. This occurs again when title is 'low' and frequency 'medium'. Both criteria are important for a good grouping, but the output is 'very high' based in term position, the same as the previous case. In these cases we are clearly underestimating the discrimination power of frequency and title. The same happens when frequency is 'medium', being title and emphasis 'low': position decides again that importance can be the minimum or not, but frequency should count more than position, as we saw before. Summarizing, *fcc* overestimates the contribution of position, underestimating at the same time the discriminative power of title, emphasis and frequency.

On the other hand, the high number of rules in *fcc* makes the possible combinations more difficult to understand. As the fuzzy system is able to combine the conclusions of the rules, another possibility for the knowledge base is the use of a set of single-input rules for each criterion and let the system calculate the output (*addfcc*, Table 4). This approach represents the knowledge in a simple way, reducing the number of cases that is needed to specify.

**Table 4.** Rule base for *addfcc*. Inputs are related to normalized term frequencies

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
	High						⇒	Very High	
	Low						⇒	No	
		High					⇒	Very High	
		Medium					⇒	Medium	
		Low					⇒	No	
			High				⇒	Very High	
			Medium				⇒	Medium	
			Low				⇒	No	
				Preferential			⇒	Very High	
				Standard			⇒	No	

Nevertheless, if we are looking for very specific definitions for each criterion, we may miss part of the knowledge expressed in the *fcc* system, especially when dealing with dependencies among criteria and not all of them contribute equally to the combination, as occurs in our case. In order to avoid this problem, an intermediate approach is proposed. We refer to it as Extended Fuzzy Combination of Criteria (*efcc*, Table 5). The main idea is to have two sets of rules: one for frequencies and another for the rest of the criteria, in such a way that we have always at least one rule of each set fired by the system, which will combine the outputs. Thus, we simplify the problem of underestimating frequency, because both subsets are always evaluated and combined. We have also reduced the discriminative power of position criterion, that is considered the least important.

For tf-idf and *fcc* we only show the best results for each dataset from Section 4.1 in order to simplify the comparison. Results on Table 6 show how *efcc* clearly improves clustering results in Webkb, while in Banksearch *addfcc* outperforms

**Table 5.** Rule base for *efcc*. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
High			High					$\Rightarrow$	Very High
High			Medium				Preferential	$\Rightarrow$	High
High			Medium				Standard	$\Rightarrow$	Medium
High			Low				Preferential	$\Rightarrow$	Medium
High			Low				Standard	$\Rightarrow$	Low
Low			High				Preferential	$\Rightarrow$	High
Low			High				Standard	$\Rightarrow$	Medium
Low			Medium				Preferential	$\Rightarrow$	Medium
Low			Medium				Standard	$\Rightarrow$	Low
Low			Low				Preferential	$\Rightarrow$	Low
Low			Low				Standard	$\Rightarrow$	No
	High							$\Rightarrow$	Very High
	Medium							$\Rightarrow$	Medium
	Low							$\Rightarrow$	No

**Table 6.** F-measure results for *addfcc* and *efcc* experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
<b>Banksearch</b>							
tf-idf lsi	0,750	0,755	0,756	0,757	0,763	0,756	<b>0,005</b>
fcc lsi	0,775	0,763	<b>0,785</b>	0,763	0,758	0,769	0,011
efcc mft	0,768	0,778	0,758	0,740	0,759	0,760	0,014
efcc lsi	<b>0,780</b>	0,756	0,744	0,755	0,757	0,758	0,013
addfcc mft	0,775	<b>0,788</b>	0,777	<b>0,784</b>	<b>0,779</b>	<b>0,781</b>	<b>0,005</b>
<b>Webkb</b>							
tf-idf lsi	<b>0,516</b>	0,507	0,505	0,506	<b>0,501</b>	0,507	0,006
fcc mft	0,453	0,472	0,475	0,468	0,475	0,469	0,009
efcc mft	<b>0,516</b>	<b>0,546</b>	<b>0,545</b>	<b>0,566</b>	0,484	<b>0,532</b>	0,032
efcc lsi	0,483	0,483	0,483	0,483	0,484	0,483	<b>0,000</b>
addfcc mft	0,459	0,493	0,494	0,491	0,471	0,482	0,016

the rest. Thus, *efcc* solves the first open issue stated at the beginning of Section 4.2: improving the bad performance of *fcc* in Webkb, with good results in Banksearch too. Besides, *addfcc* leads to worse results than *efcc* in Webkb, but obtains the best results in Banksearch in almost all cases.

These experiments for *efcc* also corroborates our hypothesis about the improvement obtained by *lsi* over *mft*, stated in Section 4.1: we have improved our weighting function and, as a result, *mft* has achieved clustering results as good as, or even better than *lsi*, with a much lower computational cost.

### 4.3 Anchor Texts

For this experiment we needed to employ a recently crawled collection, in such a way that it was easy to find other web pages with hyperlinks to the collection documents. We decided to use the dataset Social ODP 2k9 (SODP) [16] consisting of 12,616 documents retrieved from social bookmarking sites and classified by extracting the category for each URL from the first classification level of Open Directory Project. Thus, the entire collection is divided in 17 unbalanced categories, having from 39 to 3,289 documents each. In addition to the documents themselves, we collected the anchor texts corresponding to a maximum of 300 unique inlinks per each document in the collection (2,704 web pages have less than 50 inlinks, 4,717 have less than 100, so the rest, approximately 60%, have more than 100 inlinks).

We decided to add anchor texts to *efcc* in two different ways: (a) in addition to each document textual content, and (b) in addition to each document title,

i.e., giving them the same importance than title terms. Besides, we did three experiments for each case: (1) just adding anchor texts, (2) adding anchor texts and removing text corresponding to outlinks, and (3) removing a set of stop words based on a study over collection anchor text terms, containing words like click, link, homepage, etc. As we introduce here a new collection, we decided to add *fcc* as baseline to validate our results. We also include *addfcc* in these experiments due to its good performance in Banksearch (Section 4.2).

**Table 7.** F-measure results for anchor text experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
<b>SODP</b>							
fcc mft	0,195	0,237	0,254	0,256	0,266	0,242	0,028
addfcc mft	0,208	0,267	0,276	0,279	0,282	0,262	0,031
efcc mft	0,233	0,273	<b>0,287</b>	0,283	<b>0,296</b>	0,275	0,025
efcc a-1 mft	0,225	0,262	0,279	0,286	0,290	0,268	0,027
efcc a-2 mft	0,245	0,246	0,285	0,289	0,269	0,267	0,024
efcc a-3 mft	0,248	0,260	0,285	<b>0,294</b>	0,293	0,276	0,022
efcc b-1 mft	<b>0,254</b>	<b>0,287</b>	0,275	0,282	0,285	<b>0,277</b>	0,015
efcc b-2 mft	<b>0,254</b>	0,249	0,276	0,279	0,291	0,270	0,016
efcc b-3 mft	0,249	0,261	0,263	0,278	0,285	0,267	<b>0,012</b>

Table 7 shows that *efcc* based approaches outperforms *fcc* and *addfcc* in all cases. This corroborates our findings of Section 4.2 about the drawbacks of *fcc* and confirms our belief about the need of a system where not all criteria contribute the same to the combination, in contrast to *addfcc*. Regarding the contribution of anchor texts, there is no clear alternative to improve *efcc*. Anchor texts help improve clustering results with small vector sizes, particularly when anchor texts terms are considered as page titles. However, when we increase vector size, they seem to introduce noise, because clustering results get worse. About using anchor texts as titles, the best option is just adding anchor texts as title terms (named b-1). Although it is interesting to have found an improvement for smaller vector sizes, this improvement is always about 1%, and clearly does not compensate for all the process needed to obtain anchor texts.

These results might be due to poor link density or bad anchor text quality, or just to the nature of clustering problems, where the aim is to capture the aboutness of documents and not just concrete keywords. This conclusions coincide with other works like [2,9] (see Section 2), where authors conclude that anchor text terms are similar to terms used in search queries and these terms are not often in web page contents. Because of this, we believe that anchor texts are more suitable for IR tasks than for clustering problems.

## 5 Empirical Evaluation

In this section we perform a robust evaluation of *efcc* to be sure about whether or not exists a real improvement over *fcc*. As we are using a deterministic algorithm, we want to avoid the possible bias introduced by feeding the algorithm with a single set of vectors for each dataset. The solution presented here consists in dividing each dataset in 100 different sub-datasets 50% smaller than the original,

where the categories are in proportion to the original ones. We performed 100 experiments per each vector size and each sub-dataset, resulting a total of 3,000 different clustering experiments. Due to computational reasons, we chose *mft* reduction for all the experiments. This decision was also made to compare both term weighting functions in the exactly same conditions. Finally, we calculated the statistical significance between F-measure results of both representations. To this end, we employed a paired two-tailed t-test over the results obtained by both representations for each concrete vector size in the 100 sub-datasets.

**Table 8.** F-measure results for t-test experiments

Rep.\Dim.	100	500	1000	2000	5000
<b>Banksearch</b>					
efcc mft	<b>0,764</b>	<b>0,774</b>	<b>0,770</b>	0,760	0,753
fcc mft	0,718	0,760	0,765	<b>0,768</b>	<b>0,768</b>
Difference	0,047	0,014	0,006	-0,008	-0,015
<i>p</i> -value	0,000	0,000	0,002	0,000	0,000
<b>Webkb</b>					
efcc mft	<b>0,487</b>	<b>0,514</b>	<b>0,528</b>	<b>0,534</b>	0,483
fcc mft	0,446	0,462	0,470	0,485	<b>0,490</b>
Difference	0,041	0,051	0,059	0,049	-0,007
<i>p</i> -value	0,000	0,000	0,000	0,000	0,016
<b>SODP</b>					
efcc mft	<b>0,230</b>	<b>0,271</b>	<b>0,279</b>	<b>0,282</b>	<b>0,289</b>
fcc mft	0,200	0,233	0,246	0,251	0,266
Difference	0,030	0,037	0,033	0,031	0,023
<i>p</i> -value	0,000	0,000	0,000	0,000	0,000

In Table 8, for each vector size and representation we show the average F-measure values corresponding to the 100 clustering experiments (one per each sub-dataset), the difference between the corresponding averages, and the *p*-value resulting of applying the statistical t-test between both representations. Attending to *p*-values, in all cases except one, we can say that values are from different populations with likelihood > 99%. Besides, looking at the averages, in most of the cases *efcc* outperforms *fcc*. Regarding differences between representations, just in three cases *fcc* performs better than *efcc*, being the difference lower than 1% in two cases and lower than 2% in the other. In the rest of the experiments *efcc* gets an improvement over *fcc*, higher than 3% in SODP, and greater than 4% in Webkb and even with the smallest vector size in Banksearch.

## 6 Conclusions

Our experiments showed that *efcc* worked better than *fcc* by means of a better combination of criteria, where term frequency is considered as discriminant as title and emphasis, and position is taken into account as the least important criterion. This approach makes also possible to reduce the number of rules needed to specify the knowledge base taking advantage of the additive properties of the fuzzy system, and thus makes the system easier to understand. Moreover, we have shown that with a good weighting function we can use lightweight dimension reduction techniques, as the proposed *mft*, instead of using *lsi*, which implies an important reduction in computational cost. In order to continue exploring new criteria for the combination, we have evaluated the use of anchor

texts to enrich document representation. Although results were not bad, the cost of preprocessing anchor texts and their dependence on link density limit the applicability of this alternative. For this reasons we believe that it could be an interesting option when a collection fulfills these requirements and time complexity is not a problem, but in most of the cases this will not happen and we will have to carry out document representation only with document contents. Finally we performed statistical significance tests to ensure that the application of our findings has a real effect compared to previous work.

Future work could be oriented to find a way of automatically adjusting the representation to specific datasets and analyzing whether or not improves clustering results. Moreover, present work could be applied to different fields, not only the representations by themselves, but the underlying ideas.

## References

1. Dredze, M., Jansen, A., Coppersmith, G., Church, K.: Nlp on spoken documents without asr. In: EMNLP, pp. 460–470 (2010)
2. Eiron, N., McCurley, K.S.: Analysis of anchor text for web search. In: Proceedings of the 26th SIGIR, pp. 459–460 (2003)
3. Fresno, V.: Representacion autocontenido de documentos HTML: una propuesta basada en combinaciones heurísticas de criterios. PhD thesis (2006)
4. Fresno, V., Ribeiro, A.: An analytical approach to concept extraction in html environments. *J. Intell. Inf. Syst.* 22(3), 215–235 (2004)
5. Hammouda, K., Kamel, M.: Distributed collaborative web document clustering using cluster keyphrase summaries. *Information Fusion* 9(4), 465–480 (2008)
6. Karypis, G.: CLUTO - a clustering toolkit. Technical Report #02-017 (November 2003)
7. Kosko, B.: Global stability of generalized additive fuzzy systems. *IEEE Transactions on Systems, Man, and Cybernetics - C* 28, 441–452 (1998)
8. Liu, Y., Liu, Z.: An improved hierarchical k-means algorithm for web document clustering. In: ICCSIT, September 2-29, pp. 606–610 (2008)
9. Noll, M.G., Meinel, C.: The metadata triumvirate: Social annotations, anchor texts and search queries. In: Proceedings of the WI-IAT, vol. 1, pp. 640–647 (2008)
10. Ribeiro, A., Fresno, V., Garcia-Alegre, M.C., Guinea, D.: A fuzzy system for the web page representation (2003)
11. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Commun. ACM* (1975)
12. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining (2000)
13. Tan, Q., Mitra, P.: Clustering-based incremental web crawling. *ACM Trans. Inf. Syst.* 28, 17:1–17:27 (2010)
14. Tang, B., Shepherd, M., Milios, E., Heywood, M.I.: Comparing and combining dimension reduction techniques for efficient text clustering. In: Proceedings of the Workshop on Feature Selection for Data Mining, SDM (2005)
15. Wang, Y., Kitsuregawa, M.: Evaluating contents-link coupled web page clustering for web search results. In: CIKM, pp. 499–506 (2002)
16. Zubiaga, A., Martínez, R., Fresno, V.: Getting the most out of social annotations for web page classification. In: ACM DocEng, pp. 74–83 (2009)

# Clustering Short Text and Its Evaluation

Prajol Shrestha, Christine Jacquin, and Béatrice Daille

Laboratoire d’Informatique de Nantes-Atlantique (LINA),

Université de Nantes, 44322 Nantes Cedex 3, France

{Prajol.Shrestha,Christine.Jacquin,Beatrice.Daille}@univ-nantes.fr

**Abstract.** Recently there has been an increase in interest towards clustering short text because it could be used in many NLP applications. According to the application, a variety of short text could be defined mainly in terms of their length (e.g. sentence, paragraphs) and type (e.g. scientific papers, newspapers). Finding a clustering method that is able to cluster short text in general is difficult. In this paper, we cluster 4 different corpora with different types of text with varying length and evaluate them against the gold standard. Based on these clustering experiments, we show how different similarity measures, clustering algorithms, and cluster evaluation methods effect the resulting clusters. We discuss four existing corpus based similarity methods, Cosine similarity, Latent Semantic Analysis, Short text Vector Space Model, and Kullback-Leibler distance, four well known clustering methods, Complete Link, Single Link, Average Link hierarchical clustering and Spectral clustering, and three evaluation methods, clustering F-measure, adjusted Rand Index, and V. Our experiments show that corpus based similarity measures do not significantly affect the clusters and that the performance of spectral clustering is better than hierarchical clustering. We also show that the values given by the evaluation methods do not always represent the usability of the clusters.

## 1 Introduction

Clustering short text is an emerging field of research and is useful in many NLP tasks such as summarization, information extraction/retrieval, and text categorization. In general, clustering consists of two main parts, the first part is to find a score for similarity between short text and then cluster them according to these similarity scores. Short text pose a challenge while clustering because they have few words which is used to determine the similarity between short text in contrast to text documents. Existing methods use a portion of the terms empirically from a frequency list and find similarity between short text based on these terms using different text similarity methods [1] [2]. The clusters are then evaluated using mapping based measures (e.g. Purity, clustering F-measure) which have drawbacks. One of them is that these methods may not be able to evaluate the entire membership of a cluster and do not evaluate every cluster [3]. Due to this drawback of the mapping based measures, the usefulness of the existing short text clustering methods cannot be judged [4].

Here in this paper, we use four short text corpus, three created from abstracts of scientific papers and the other created from newspaper paragraphs, described in Sect. 3.1, to give an idea of the different variation present in short text and how different clustering methods behave on them. We use Single link (SHC), Complete link (CHC), and Average link (AHC) hierarchical clustering methods. Along these methods, we use spectral clustering (SPEC) which has not been used in the scope of short text. This clustering method has been very successful in the field of machine learning such as image segmentation [5]. Clustering methods depend on similarity values and to see its effect we use four existing similarity measures. The clusters are then evaluated using clustering F-measure (F), adjusted Rand Index (ARI) and V. We demonstrate that none of these measures relate to the usability aspect of the clusters so they are not always able to properly evaluate the quality of clusters. We start by describing the clustering methods.

## 2 Clustering Methods

Clustering short text is the task of grouping short text together into groups in such a way that short text related to a category are found in a unique group. It consists of two steps: the first step is to find the similarity or dissimilarity matrix and then clustering the short text with the help of this matrix. In this paper we consider dissimilarity between two text to be one minus the similarity between them. We used four different corpus based similarity methods to create the matrix namely cosine similarity (CS) measure using tf-idf weights, Latent Semantic Analysis (LSA) using log(tf)-idf weights [6], Short text Vector Space Model (SVSM) [7], and Kullback-leibler distance (KLD) [8]. These measures are used by each of the clustering methods. In this section, we give a brief description of all the similarity and clustering methods.

### 2.1 Short Text Similarity Methods

**Cosine Similarity Measure (CS) :** This measure has been extensively used in NLP to find similarities between text where the text is represented as a weighted vector [9]. Here we use  $tf * idf$  weights where  $tf$  and  $idf$  stands for term frequency and inverse document frequency respectively. For us documents are short text. Given two short text  $\vec{t}_a$  and  $\vec{t}_b$ , their cosine similarity is computed with (1).

$$CS(\vec{t}_a, \vec{t}_b) = \frac{\vec{t}_a \cdot \vec{t}_b}{|\vec{t}_a| \times |\vec{t}_b|} \quad (1)$$

where,  $\vec{t}_a$  and  $\vec{t}_b$  are m-dimensional vectors of short text  $a$  and  $b$  over the term set of  $T = \{t_1, t_2, \dots, t_m\}$ . Each vector dimension represents a term with its weight corresponding to the short text, which is a non-negative value. As a result, the cosine similarity is non-negative and bounded between [0,1] where 1 indicates the two text are identical.

**Latent Semantic Analysis (LSA)** : LSA is a method which is used to find similarity between text using singular value decomposition (SVD), which is a form of factor analysis and is well-known in linear algebra [10]. SVD decomposes the rectangular term-by-short text matrix,  $\mathbf{M}$ , into three other matrices  $\mathbf{M} = \mathbf{U}\Sigma_k\mathbf{V}^T$  where  $\mathbf{U}$  and  $\mathbf{V}$  are column-orthogonal matrices and  $\Sigma_k$  is a diagonal  $k \times k$  matrix which contains  $k$  singular values of  $\mathbf{M}$  such that the singular values are in the descending order,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k$ . We choose  $k' \ll k$  and multiply the three matrices to get  $\mathbf{M} \simeq \mathbf{U}\Sigma_{k'}\mathbf{V}^T$  which is a re-composed matrix of the original matrix  $\mathbf{M}$ . The similarity between short text is then computed using the cosine similarity measure between the columns of the new matrix  $\mathbf{M}$ .

**Kullback-Leibler Distance (KLD)** : KLD is used in [8] to cluster narrow domain abstracts and is based on Kullback-Leibler (KL) divergence which is used to give a value to the difference between two distributions. For two distributions  $P$  and  $Q$  the KL divergence on a finite set  $X$  is shown in (2).

$$D_{KL}(P\|K) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \quad (2)$$

This measure is not symmetric but there exists symmetric versions. In [8] they have used some and shown that there is not much difference between them. We implemented the max of the KL distance as in (3).

$$D_{KLD} = \max(D_{KL}(P\|K), D_{KL}(K\|P)) \quad (3)$$

To use  $D_{KLD}$  as a distance measure for short text we compute the probabilities as shown in (4) and they are based on the distribution of the terms in the vocabulary,  $V$ .

$$P(t_k, d_i) = \begin{cases} \beta * P(t_k|d_i), & \text{if term } t_k \text{ occurs in the document } d_i \\ \epsilon, & \text{otherwise} \end{cases} \quad (4)$$

where,

$$P(t_k|d_i) = \frac{tf(t_k, d_j)}{\sum_{t_k \in d_i} tf(t_k, d_i)}$$

and

$$\beta = 1 - \sum_{t_k \in V, t_k \notin d_i} \epsilon \quad \text{such that,} \quad \sum_{t_k \in d_i} \beta * P(t_k|d_i) + \sum_{t_k \in V, t_k \notin d_i} \epsilon = 1$$

In [11] [8], KLD relies only on terms of a certain portion of the vocabulary. This selection of the terms were done using three methods and among them we choose *Document Frequency* (DF) technique because no parameter has to be estimated and it gives a stable result. The concept of this term selection is that, the lower frequency terms in the collection of text do not play a role in predicting the class for the text. In order to implement KLD we select the top 70% of the vocabulary which was sorted in a descending order according to the term frequency.

**Short Text Vector Space Model (SVSM) :** This method is used in [7] to find similarity between short text. For each text, a text vector is created from term vectors. Given a corpus  $C$  of  $n$  short text and  $m$  unique terms, the term vector,  $\vec{t}_j$ , for term  $t_j$  is a vector created with  $n$  number of possible dimensions where each dimension represents a unique short text. The presence of the term in a short text is indicated by its sentence id and the term's inverse document frequency,  $idf$ , here a document is a short text, as shown below:

$$\vec{t}_j = [(S_1, idf_j), (S_5, idf_j), \dots, (S_i, idf_j)]$$

where  $S_i$  is the short text id where  $t_j$  is present,  $i \in 1, \dots, n$  and  $idf_j$  is the idf value of term  $t_j$ . This term vector is a reduced vector space representation where short text that do not contain the term is absent which saves space. The dimension of the matrix formed by term vectors can be further reduced using LSA [12] or Principle Component Analysis [13] but are not used here. Once we have the term vectors we can create a short text vector by adding the term vectors of the terms present in that short text. For a short text consisting of terms  $t_1, t_2, \dots, t_k$ , the dimension,  $d_i$ , of the sentence vector corresponding to the short text  $S_i$ , will be  $d_i = \sum_{j=1; t_j \in S_i}^k idf_j$ , where  $idf_j$  is the idf value of the term  $j$  and  $i \in 1, \dots, n$ . The similarity between short text is calculated using the cosine similarity between the text vectors.

## 2.2 Clustering Algorithms

The hierarchical agglomerative clustering (HC) and SPEC clustering methods are described in this section. HC are bottom up algorithms in which elements are merged together to form dendograms and are used extensively in the field of NLP. Different HC algorithms are present but have the same underlying approach and can be formally written as these steps:

1. Compute the dissimilarity matrix with one of the approach given in Sect. 2.1
2. Start with each short text in one cluster and repeat the following steps until a single cluster is formed :
  - (a) Merge the closest two clusters.
  - (b) Update the dissimilarity matrix to reflect the dissimilarities between the new cluster and the original clusters.
3. Cut the dendrogram in a way we find the required number of clusters.

The three hierarchical clustering, SHC, CHC and AHC used here differ in step 2a where the closest clusters are determined. Below, we state how the closeness are determined for each algorithm.

**Single Link HC (SHC) :** This clustering method considers two clusters to be close in terms of the minimum dissimilarities between any two elements in the two clusters.

**Complete Link HC (CHC) :** This clustering method considers two clusters to be close in terms of the maximum dissimilarities between any two elements in the two clusters.

**Average Link HC (CHC) :** This clustering methods considers two clusters to be close in terms of the average pairwise dissimilarities of all the pairs of elements in the two clusters.

**Spectral Clustering (SPEC) :** Along with the HC algorithms we also use Spectral Clustering which has been recently used in the community of machine learning [5]. K-means clustering algorithm is the underlying clustering algorithm of SPEC which is applied on the normalized eigenvectors of the similarity matrix. The algorithm for spectral clustering is given below from [14] :

1. Given a set of short text,  $S = \{s_1, \dots, s_n\}$ , the similarity matrix,  $M \in \mathbb{R}^{n \times n}$ , is generated using some similarity measures mentioned in Sect. 2.1.
2. Create the affinity matrix  $A \in \mathbb{R}^{n \times n}$  defined by the Gaussian Similarity function,  $A_{ij} = \exp(-\|r_i - r_j\|^2/2\sigma^2)$  with  $\sigma = 0.5$ , if  $i \neq j$ , and  $A_{ii} = 0$ , where  $r_i, \dots, r_j$  are rows of  $M$ .
3. Construct the normalized graph Laplacian matrix  $L = D^{-1/2}AD^{-1/2}$  where,  $D$  is a diagonal matrix whose  $(i, i)$ -element is the sum of  $A$ 's  $i$ -th row.
4. Compute the eigenvectors of  $L$  and select the  $k$  largest eigenvectors and stacking them in columns to form  $X = [x_1, x_2, \dots, x_k] \in \mathbb{R}^{n \times k}$ .
5. Normalize the row's of  $X$  to have unit length to form the matrix  $Y$  (i.e.  $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$ ).
6. Using K-means, cluster the rows of matrix  $Y$  into  $k$  clusters by treating the row of  $Y$  as points in  $\mathbb{R}^k$ .

### 3 Experiments and Results

In this section, we analyse the behaviour of the clustering methods, the effect of similarity measures on clustering methods, and the evaluation methods. We start by describing the corpus and the evaluation methods so that we can explain the results of the experiments.

#### 3.1 Corpus

We use 4 different types of corpora with regards to the size, type, and the distribution of short text among the clusters. These corpora consist of paragraphs of text from newspapers as well as narrow domain abstracts which make them representative of short text that are normally dealt in the field of written NLP. We use three corpora namely CICLing-2002, hep-ex, and KnCr, created from scientific abstracts, which have been used previously for short text clustering [8] and will serve as a reference corpus. We also use a new short text corpus collected from newspapers. Here, we give a short description of each of these corpora.

**The LDC Corpus :** This corpus is a collection of 12 newspaper articles concerning the *Death of Diana*. The articles were taken from the Linguistic Data Consortium's (LDC) North American News Text Corpus<sup>1</sup>. We consider

---

<sup>1</sup> LDC Catalog number: LDC95T21

each paragraph a short text and each paragraph was manually annotated<sup>2</sup> with one of the 13 categories it is related to. The annotations were done by two annotators independently and the reliability of agreement on the annotation of these categories according to Fleiss' kappa [15] is 0.91. Which is an almost perfect agreement. The small error that arose was due to the fact that some paragraphs could be related to more than one categories but were assigned to one category. The disagreements were resolved between the annotators by discussing the main idea of the paragraph. Table 1 gives the distribution of the paragraphs according to the categories and some other properties of the corpus.

**Table 1.** Properties of the LDC corpus

(a) Number of paragraphs in each category		(b) Features	
Categories	Paragraphs	Feature	Value
Diana's life before accident	22	Number of categories	13
Driver's life before accident	5	Number of paragraphs	242
Other's life before accident	9	Total number of terms	5,351
Just before accident	18	Vocabulary size (terms)	1,761
Accident	10	Term average per paragraph	22.45
Just after accident	22		
Accident aftermath	8		
Expression of grief	32		
Funeral	46		
Accusations	13		
Cause	17		
Investigation	20		
Media	20		

**The CICLing-2002 Corpus :** This is a small corpus consisting of 48 abstracts in the domain of computational linguistics collected from the CICLing 2002 conference. This corpus has 4 classes of 48 abstracts and the abstracts are evenly distributed among the 4 classes which is as follows : {11, 15, 11, 11}.

**The hep-ex Corpus of CERN :** This corpus contains 2,922 abstracts collected by the University of Jaén, Spain on the domain of Physics from the data server of the CERN. These abstracts are related to 9 categories. The distribution of the abstracts among the 9 classes is highly uneven and is as follows: {2623, 271, 18, 3, 1, 1, 1, 1, 1}

**The KnCr Corpus of MEDLINE :** This corpus contains abstracts from the cancer domain of the medical field and collected from the MEDLINE documents [1]. It contains 900 abstracts and they are related to 16 categories. The abstracts are distributed among the 16 classes as follows :{169, 160, 119, 99, 66, 64, 51, 31, 30, 29, 22, 20, 14, 12, 8, 6}

<sup>2</sup> Annotations at URL: <http://www.projet-depart.org/public/LINA-PCL-1.0.tar.gz>

### 3.2 Evaluation Methods

For the purpose of evaluating the quality of the clusters, we use 3 existing measures. These measures will determine which clustering methods produce the best clusters. The details of these measures are based on the initial setting where  $S$  number of short text are naturally grouped into classes denoted by  $C = \{c_1, c_2, \dots, c_n\}$  and are clustered by the clustering algorithms into groups denoted by  $K = \{k_1, k_2, \dots, k_3\}$ .

**Clustering F-measure (F) :** F is a mapping based measure where evaluation is done by mapping each cluster to a class [16] and is based on precision and recall as follows:

$$F(C) = \sum_{C_i \in C} \frac{|C_i|}{S} \max_{K_j \in K} \{F(C_i, K_j)\} \quad (5)$$

where,

$$\text{Recall}(C_i, K_j) = \frac{n_{ij}}{|C_i|} \quad \text{Precision}(C_i, K_j) = \frac{n_{ij}}{|K_j|}$$

and

$$F(C_i, K_j) = \frac{2 \times \text{Recall}(C_i, K_j) * \text{Precision}(C_i, K_j)}{\text{Recall}(C_i, K_j) + \text{Precision}(C_i, K_j)}$$

where  $n_{ij}$  is the number of short text of class  $C_i$  present in clusters  $K_j$ . The F value will be in the range of [0,1], where 1 being the best score. A slight variation of this method has also been used in clustering short text [8] which computes the F according to the clusters rather than the class and is computed as  $F(K) = \sum_{K_j \in K} \frac{|K_j|}{S} \max_{C_i \in C} \{F(C_i, K_j)\}$  which we do not use in this paper.

**Adjusted Rand Index (ARI) :** This measure is an improvement of the Rand Index [17] which is based on counting pairs of elements that are clustered similarly in the classes and clusters. With the initial setting the ARI can be computed as below:

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{S}{2}}{1/2 [\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}] / \binom{S}{2}} \quad (6)$$

where  $n_{ij}$  is the number of short text of class  $C_i$  present in cluster  $K_j$ ,  $n_i$  is the number of short text in class  $C_i$ ,  $n_j$  is the number of short text in the cluster  $K_j$ . The upper bound of this measure is 1 and corresponds to the best score and the expected value of this measure is zero.

**V :** V is based on information theory and uses entropy and conditional entropy to evaluate the cluster [18]. The value of V are computed as in (7).

$$V = \frac{2hc}{h+c} \quad \text{where,} \quad h = \begin{cases} 1 & H(C) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \quad c = \begin{cases} 1 & H(K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases} \quad (7)$$

with,

$$\begin{aligned}
 H(C) &= - \sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{ck}}{S} \log \frac{\sum_{k=1}^{|K|} a_{ck}}{S} \\
 H(K) &= - \sum_{k=1}^{|K|} \frac{\sum_{c=1}^{|C|} a_{ck}}{S} \log \frac{\sum_{c=1}^{|C|} a_{ck}}{S} \\
 H(C|K) &= - \sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{a_{ck}}{S} \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}} \\
 H(K|C) &= - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{a_{ck}}{S} \log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_{ck}}
 \end{aligned}$$

where,  $a_{ck}$  is the number of short text in  $C_i$  which is present in  $K_j$ . V gives an evaluation score in a range of [0,1], 1 being the best score.

### 3.3 Clustering

We used all three hierarchical clustering and the spectral clustering methods to cluster the short text present in the corpora. The clustering was performed in R<sup>3</sup> which is an environment for statistical computing and graphics. After clustering, the distribution of text among clusters shows that each clustering method has its own characteristics that defines the type of clusters that are created in terms of the distribution of text. Table 2 shows the distributions of the elements in the clusters created by all four clustering methods, which used cosine similarity, on the Cicling-2002 corpus and the hep-ex corpus.

**Table 2.** Distribution of the short text among the clusters created by SHC, CHC, AHC, and SPEC which uses cosine similarity

(a) Cicling-2002 corpus

Clustering	Cluster Index			
	1	2	3	4
SHC	45	1	1	1
CHC	11	24	7	6
AHC	33	12	1	2
SPEC	13	4	9	22

(b) hep-ex corpus

Cluster	Cluster Index								
	1	2	3	4	5	6	7	8	9
SHC	2912	1	1	1	1	1	1	1	1
CHC	2879	5	11	5	2	4	5	5	4
AHC	2879	13	11	1	5	3	5	2	1
SPEC	298	248	396	337	243	328	371	303	396

From Table 2, we can see that SHC creates many clusters with only one element in it which indicates that for our purpose, single link hierarchical clustering may not be a good choice. The characteristics of SPEC shows that it distributes the text evenly throughout the clusters. CHC and AHC have similar characteristics which lie between SHC and SPEC. These characteristics of the clustering method remain the same irrespective of the corpora but this characteristic alone cannot be used to decide upon the appropriate method for clustering.

There are evaluation methods that give scores on the quality of the clusters and based on these scores we tend to decide on the appropriate clustering

<sup>3</sup> <http://www.r-project.org/>

method. Different evaluation methods have different properties [3], so before we decide on the clustering method we first have to decide on which evaluation method would be appropriate.

We compare the evaluation methods using a direct method where we assign each cluster generated by the clustering method to a unique class in such a way that the average F-score (AF) for each pair of cluster and class is maximized. F-score is defined as  $F(C_i, K_j)$  in (5). As we maximize the AF, the resulting pairs of cluster and class could be considered as the best practical solution. Tables 3(a) 3(b) 3(c) 3(d) show the F-score confusion matrix of class against clusters generated by 4 clustering methods, using CS, on the Cicling-2002 corpus. The bold-faced values in each matrix makes the AF maximum. This optimal assignment is done automatically using the Hungarian Algorithm [19]. Table 3(e) shows the scores given to each clustering method by the 3 evaluation methods and maximum AF (MAF). We consider an evaluation method to be good if it resembles the MAF scores because a high value for MAF generally indicates a high level of agreement between the classes and the clusters.

**Table 3.** In (a),(b),(c), and (d) the F-score confusion matrices for SHC, CHC, AHC, and SPEC applied on the CICLing-2002 corpus are shown and the elements which make the MAF are bold-faced. The classes and clusters are represented by the rows and columns respectively. In (e) the clusters generated by the clustering methods are evaluated using F, ARI, V, and MAF.

(a) SHC				(b) CHC				(c) AHC				
<b>0.17</b>	0	0.36	0	0.11	0.29	0.36	<b>0.12</b>	0	0.17	0.36	<b>0.15</b>	
0	0	0.5	<b>0</b>	0	<b>0.56</b>	0.15	0.19	0	0.15	<b>0.54</b>	0	
0	0	<b>0.39</b>	0	0	0.29	<b>0.45</b>	0.12	<b>0</b>	0	0.5	0	
0	<b>0.17</b>	0.32	0.17	<b>0.67</b>	0.17	0	0.24	0.17	<b>0.70</b>	0.05	0.15	
(d) SPEC				(e) Cicling-2002								
0.1	<b>0.27</b>	0.25	0.3					F	ARI	V	MAF	
0	0.11	0.14	<b>0.65</b>					SHC	0.40	0.01	0.11	0.21
<b>0.80</b>	0	0	0.18					CHC	0.52	0.10	0.21	0.45
0	0.13	<b>0.67</b>	0.12					AHC	0.53	0.17	0.29	0.35
								SPEC	<b>0.61</b>	<b>0.25</b>	<b>0.34</b>	<b>0.60</b>

Table 3(e) does not help us find the best evaluation method because no evaluation method represents the MAF value, but it certainly gives an insight on the performance of the clustering methods. All of the evaluation methods do point towards spectral clustering to be the best clustering method for our case. Table 4 gives the complete results of the experiments. It shows that for all the corpus, excluding hep-ex corpus, spectral clustering performs better than the rest. In the case of hep-ex, the short text are unevenly distributed among the clusters as shown in Sect. 3.1 and as the characteristics of the spectral clustering tends to make evenly distributed clusters the performance decreases.

**Table 4.** F, ARI, V, and MAF values for four clustering methods SHC, CHC, AHC and SPEC on four corpus KnCr, hep-ex, Cicling-2002, and LDC. The best score achieved by each evaluation method on every corpus are bold-faced.

Corpus		<i>KnCr</i>				<i>Cycling-2002</i>			
Cluster	Similarity	F	ARI	V	MAF	F	ARI	V	MAF
SHC	Cosine	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
	KLD	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
	LSA	0.21	0.00	0.04	0.05	0.40	0.00	0.11	0.17
	SVSM	0.20	0.00	0.03	0.04	0.40	0.01	0.11	0.21
CHC	Cosine	0.21	0.01	0.12	0.14	0.52	0.10	0.21	0.45
	KLD	0.20	-0.01	0.11	0.16	0.45	0.06	0.18	0.33
	LSA	0.21	0.03	0.12	0.09	0.52	0.11	0.23	0.52
	SVSM	0.22	0.01	0.09	0.10	0.46	0.07	0.19	0.40
AHC	Cosine	0.25	0.04	0.12	0.13	0.53	0.17	0.29	0.35
	KLD	0.21	0.00	0.04	0.05	0.40	0.02	0.15	0.25
	LSA	0.21	0.00	0.06	0.07	0.40	0.00	0.10	0.21
	SVSM	0.20	0.00	0.04	0.04	0.40	0.02	0.15	0.25
SPEC	Cosine	<b>0.30 0.09 0.19 0.19</b>				0.61	0.25	<b>0.34</b>	0.60
	KLD	0.23	0.04	0.11	0.14	0.51	0.15	0.26	0.51
	LSA	0.24	0.04	0.15	0.17	0.55	0.19	0.27	0.52
	SVSM	0.22	0.03	0.13	0.16	<b>0.64 0.26 0.34 0.64</b>			
Corpus		<i>hep-ex</i>				<i>LDC</i>			
Cluster	Similarity	F	ARI	V	MAF	F	ARI	V	MAF
SHC	Cosine	<b>0.86</b>	0.01	0.01	0.10	0.19	0.00	0.09	0.08
	KLD	<b>0.86</b>	-0.02	0.01	0.10	0.19	0.00	0.09	0.07
	LSA	<b>0.86</b>	0.01	0.01	0.11	0.19	0.00	0.09	0.07
	SVSM	<b>0.86</b>	0.01	0.01	0.11	0.19	0.00	0.09	0.08
CHC	Cosine	<b>0.86</b>	-0.01	0.01	0.11	0.21	0.10	0.15	0.13
	KLD	0.81	-0.02	0.00	0.10	0.29	0.02	0.26	0.21
	LSA	0.41	0.01	0.02	0.08	0.29	0.08	0.28	0.25
	SVSM	0.56	0.03	0.07	0.11	0.41	0.24	0.42	0.24
AHC	Cosine	<b>0.86</b>	0.03	0.01	0.11	0.38	0.18	0.38	0.21
	KLD	<b>0.86</b>	-0.01	0.00	0.10	0.35	0.14	0.36	0.18
	LSA	<b>0.86</b>	<b>0.10</b>	0.05	<b>0.13</b>	0.43	0.22	0.42	0.28
	SVSM	<b>0.86</b>	0.00	0.01	<b>0.13</b>	0.31	0.14	0.32	0.14
SPEC	Cosine	0.28	0.01	<b>0.08</b>	0.09	<b>0.50</b>	<b>0.29</b>	<b>0.50</b>	0.41
	KLD	0.47	0.00	0.03	0.08	0.26	0.05	0.24	0.21
	LSA	0.28	0.01	<b>0.08</b>	0.09	<b>0.51</b>	0.27	0.49	<b>0.43</b>
	SVSM	0.29	0.01	<b>0.08</b>	0.09	0.45	0.23	0.45	0.36

For the hep-ex corpus, F evaluation method gives a good result for SHC even though the distribution of the short text in the clusters are clearly undesirable for other clusters as seen in Table 2. This is due to the drawback of F as it may not take into account the membership of the clusters and may not evaluate the

clusters. From this table we can also see that none of the evaluation measure resembles the MAF values. But if required, we would select V as the best out of the three evaluation methods. The reason behind this selection is that, V resembles the variation in the range of MAF more than the other evaluation measures. Among the 16 possible range of MAF, present in each box in Table 4, V resembles MAF 9 times where as ARI 7 times.

It is also difficult to comment on the similarity measures because the clusters formed are highly affected by the different characteristics of the corpora which overshadows the effect of the similarity measures. But as we consider spectral clustering to be a good clustering method, according to the number of best evaluation scores achieved shown in Table 4, we analyse the similarity measures based on these best scores. By doing so, we see that in most of the cases the spectral clustering which uses KLD similarity measure produces clusters whose evaluation score have the highest difference with the best evaluation score. This could indicate that the performance of KLD is the least among the other similarity measures. The other three similarity measures do not differ much in most of the cases comparing the evaluation measures.

## 4 Conclusions

In this paper, we cluster short text from four different corpora containing different type, size, and distribution of short text. This difference in the corpora is important to present a generalized solution for the clustering of short text. Among the corpora, three of them have been used in previous research on clustering abstracts. We present a new annotated corpus containing newspaper paragraphs to analyse the clustering of short text. The cluster list for this corpus can be freely downloaded for further research on this field.

To analyse the clustering of short text, we used three hierarchical clustering algorithms which is famous in the field of NLP and spectral clustering which is based on k-means clustering algorithm to show that the latter seems to be a good choice over hierarchical clustering especially when the text are evenly distributed among the clusters. We also show that the performance of KLD method, which uses term selection, is the least compared to the other three measures and the performance of CS, LSA, SVSM do not differ much from each other.

Using the Hungarian algorithm, we assigned each cluster to a class so that the average F-score, AF, is maximized. The maximized AF method can also be considered as an evaluation method if the number of class is the same as the clusters. This optimized assignment was the basis of choosing the best evaluation method. Unfortunately, none of the evaluation method closely resembled the MAF but taking into account the number of times a method shows resemblance to the MAF measure, V has an upper hand. Existing work of short text clustering evaluate the clusters using mapping based methods such as clustering F-measure or Purity. We show that these measures are not able to evaluate the entire membership of the clusters which is a huge drawback. This implies that results from previous work which use these mapping based evaluation methods have to be analysed carefully.

## References

1. Pinto, D., Rosso, P.: Kncr: A short-text narrow-domain sub-corpus of medline. In: Proceedings of the TLH 2006 Conference. Advances in Computer Science, pp. 266–269 (2006)
2. Makagonov, P., Alexandrov, M., Gelbukh, A.: Clustering Abstracts Instead of Full Texts. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 129–135. Springer, Heidelberg (2004)
3. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval* 12, 461–486 (2009)
4. Reichart, R., Rappoport, A.: The nvi clustering evaluation measure. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pp. 165–173 (2009)
5. von Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* 17, 395–416 (2007)
6. Nakov, P., Popova, A., Mateev, P.: Weight functions impact on lsa performance. In: EuroConference RANLP 2001, Recent Advances in NLP, pp. 187–193 (2001)
7. Shrestha, P., Jacquin, C., Daille, B.: Reduction of search space to annotate monolingual corpora. In: Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011) (2011)
8. Pinto, D., Benedí, J.-M., Rosso, P.: Clustering Narrow-Domain Short Texts by Using the Kullback-Leibler Distance. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 611–622. Springer, Heidelberg (2007)
9. Manning, C.D., Raghavan, P., Schütze, H.: Clustering Narrow-Domain Short Texts by using the Kullback-Leibler Distance. Cambridge University Press (2008)
10. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to latent semantic analysis. In: Discourse Processes (1998)
11. Pinto, D., Jiménez-Salazar, H., Rosso, P.: Clustering Abstracts of Scientific Texts Using the Transition Point Technique. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 536–546. Springer, Heidelberg (2006)
12. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 391–407 (1990)
13. Jolliffe, I.T.: Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2, 37–52 (1986)
14. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856. MIT Press (2001)
15. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 378–382 (1971)
16. Fung, B.C., Wang, K., Ester, M.: Hierarchical document clustering using frequent itemsets. In: Proceedings of SIAM International Conference on Data Mining, SDM 2003 (2003)
17. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* 2, 193–218 (1985)
18. Rosenberg, A., Hirschberg, J.: V-measure: a conditional entropy-based external cluster evaluation measure. In: EMNLP 2007 (2007)
19. Harold, K.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97 (1955)

# Information Extraction from Webpages Based on DOM Distances\*

Carlos Castillo<sup>1</sup>, Héctor Valero<sup>1</sup>, José Guadalupe Ramos<sup>2</sup>, and Josep Silva<sup>1</sup>

<sup>1</sup> Universidad Politécnica de Valencia, Camino de Vera s/n, E-46022 Valencia, Spain  
 [{carcasg1,hecvalli}@upv.es](mailto:{carcasg1,hecvalli}@upv.es), [jsilva@dsic.upv.es](mailto:jsilva@dsic.upv.es)

<sup>2</sup> Instituto Tecnológico de La Piedad, La Piedad, México  
[guadalupe@dsic.upv.es](mailto:guadalupe@dsic.upv.es)

**Abstract.** Retrieving information from Internet is a difficult task as it is demonstrated by the lack of real-time tools able to extract information from webpages. The main cause is that most webpages in Internet are implemented using plain (X)HTML which is a language that lacks structured semantic information. For this reason much of the efforts in this area have been directed to the development of techniques for URLs extraction. This field has produced good results implemented by modern search engines. But, contrarily, extracting information from a single webpage has produced poor results or very limited tools. In this work we define a novel technique for information extraction from single webpages or collections of interconnected webpages. This technique is based on DOM distances to retrieve information. This allows the technique to work with any webpage and, thus, to retrieve information online. Our implementation and experiments demonstrate the usefulness of the technique.

## 1 Introduction

*Information Extraction* (IE) is one of the major areas of interest in both the web and the semantic web. The lack of real-time online applications able to automatically extract information from the web shows the difficulty of the problem. Current techniques for IE from Internet are mainly based on the recovering of webpages that are related to a specified query (see [7] for a survey). In this area, search engines such as Google or Bing implement very efficient and precise algorithms for the recovering of related webpages. However, for many purposes, the granularity level of the produced information is too big: a whole webpage.

In this work we try to reduce the granularity level of the information obtained. In particular we introduce a technique that given a collection of webpages, it extract from them all the information relevant for a given query and shows to the user in a new recomposed webpage.

---

\* This work has been partially supported by the Spanish *Ministerio de Ciencia e Innovación* under grant TIN2008-06622-C03-02 and by the *Generalitat Valenciana* under grant PROMETEO/2011/052.

In the semantic web setting, it is often possible to produce similar results composed of texts that answer a given question. However, these techniques often need to pre-process the webpages that are going to be queried. An ontological model is constructed and the knowledge is modeled and queried using languages such as RDF [8] or OWL [9]. This imposes important restrictions on the webpages that can be processed, and thus the implemented tools are usually offline tools. One reason is that most Internet pages have been implemented with plain (X)HTML. A similar problem is faced by the related techniques and tools that use microformats [10,11,12] to represent knowledge.

In this work we introduce a novel technique for IE that is based on DOM distances. Roughly speaking the technique looks for a term specified by the user, and it extracts from the initial webpage and some linked webpages those elements that are close to this term in the DOM trees of the webpages. Therefore, the technique relies on the idea that syntactically close means semantically related. This idea is also extended to distances between pages and domains using hyperlink distances. The main advantages of the technique are that it does not need to use proxies (as in [13]), it can work online (with any webpage) without any pre-compilation or pre-parsing phases (as in [14]); and that it can retrieve information at a very low level of granularity: a single word.

We are not aware of any tool that performs the kind of filtering that our system does. Other related approaches and tools [5] for web content filtering focus on the detection of one particular kind of content (such as porn or violence) in order to filter the whole webpage from a list of webpage results. Therefore, they do not decompose a webpage and filter a part of it as we do. Similar approaches are based on the use of neural networks [4] and application ontologies [6].

There are some works specialized for a particular content (such as tables) that are somehow related to our work. They do not focus on filtering but in content extraction from tables [1], or in wrappers induction [2,3]. In general, they detect a particular content in tables and extract it to be used as a database.

## 2 Information Extraction Based on DOM Distances

Our technique is based on the Document Object Model (DOM) [15] which is an API that provides programmers with a standard set of objects for the representation of HTML and XML documents. Our technique is based on the use of DOM as the model for representing webpages. For the sake of concreteness, in the following we will assume that a DOM tree is a data structure that represents each single element of a webpage with a node labelled with a text. This simplification assumes that all nodes have a single attribute, and it allows us to avoid in our formalization and algorithms low-level details such as the distinction between different kinds of HTML elements' attributes. For instance, in our implementation we have to distinguish and query different properties depending on the element that we are analyzing, e.g., image nodes are queried by using their *alt*, *longdesc* and *src* attributes.

**Definition 1 (DOM Tree).** *A DOM tree  $t = (V, E)$  is a tree whose vertices  $V$  are nodes labelled with HTML elements connected by a set of edges  $E$ .*

We often refer to the label of a DOM node  $n$  with  $l(n)$ ; and we refer to the root of a DOM tree  $t$  with  $root(t)$ . We also use the notation  $n -^x n'$  to denote that there exists a path of size less or equal to  $x$  between nodes  $n$  and  $n'$ . If the path is of size  $x$ , then we say that the DOM distance between  $n$  and  $n'$  is  $x$ . Edges are represented as  $(n \rightarrow n')$  with  $n, n' \in V$ . We use  $\rightarrow^*$  to denote the reflexive and transitive closure of  $\rightarrow$ .

**Definition 2 (Webpage).** *A webpage is a pair  $(u, t)$  where  $u$  is a URL and  $t$  is a DOM tree.*

For simplicity, we assume that queries are composed of a single word. The extension of the technique for multiple words is trivial and it only requires the iteration of the method over the words of the query. This has been already done in our implementation, and thus, the interested reader is referred to its (open) source code for implementation details.

**Definition 3 (Query).** *A query is a pair  $(w, d)$  where  $w$  is a word that is associated with the information which is relevant for the user; and  $d$  is an integer that represents the tolerance required in the search.*

In our setting, the *tolerance* represents the maximum DOM distance allowed. The tolerance is used to decide what elements of the DOM tree are related to the user specified word. With tolerance=0, only elements that contain the specified word should be retrieved. With tolerance=1, only elements that contain the word and those that are in a distance of 1 to them should be retrieved, and so on.

Algorithm 1 implements our method for information extraction of single webpages. Clearly, this algorithm has a cost linear with the size of the DOM tree. In essence, it finds the *key\_nodes* that are those whose label contains the searching word. From these nodes, the *relevant\_nodes* are computed which are those whose DOM distance to the key nodes is equal or lower than the tolerance specified by the user. This idea is an important contribution of this technique because it is a novel method to retrieve semantically related information. Our experiments and implementation together with massive use of anonymous users demonstrate the practical utility of this DOM distance. All the *ancestors* and *successors* of the relevant nodes form the final nodes of the filtered DOM tree. The final edges are those induced by the final set of nodes. Therefore, the final webpage (that we will call in the following *slice*) is always a portion of the original webpage, and this portion keeps the original structure of information because the paths between retrieved elements are maintained.

In order to extend our algorithm for information extraction of interconnected webpages, in the following we will assume that the user has loaded a webpage (that we call *initial webpage*) and she specifies a query to extract information from this webpage, the webpages that are linked to it (either as incoming or outgoing links), the webpages included in it (e.g., as frames or iframes) and the webpages to which it belongs (e.g., as a frame or an iframe). We call all these pages the *interconnected webpages*; and observe that they are not necessarily in the same domain.

**Algorithm 1.** Information extraction from single webpages**Input:** A webpage  $P = (u, t)$  and a query  $q = (w, d)$ **Output:** A webpage  $P' = (\perp, t')$ **Initialization:**  $t = (v, e), t' = (\emptyset, \emptyset)$ 

- (1)  $key\_nodes = \{n \in v \mid l(n) \text{ contains } w\}$
  - (2)  $relevant\_nodes = \{n \in v \mid n - d \wedge n' \in key\_nodes\}$
  - (3)  $ancestors = \{n \in v \mid n_0 \rightarrow^* n \rightarrow^* n_1 \wedge n_0 = \text{root}(t) \wedge n_1 \in relevant\_nodes\}$
  - (4)  $successors = \{n \in v \mid n_0 \rightarrow^* n \wedge n_0 \in relevant\_nodes\}$
  - (5)  $edges = \{(n, n') \in e \mid n, n' \in (successors \cup ancestors)\}$
- 
- return**  $P' = (\perp, (successors \cup ancestors, edges))$
- 

Frames and iframes can be modeled by considering that their DOM trees are subtrees of the webpage that contains them. Therefore, Algorithm 1 is able to extract relevant information from composite webpages structured with frames. For hyperlinks, we can assume that the label of some nodes in a DOM tree is a link pointing to a webpage. This is enough to define the notions of reachable webpages and search hyperspace used in our information extraction algorithm.

**Definition 4 (Reachability).** *Given a webpage  $P_0$  we say that webpage  $P_n$  is reachable from  $P_0$  if and only if  $\exists P_0, P_1, \dots, P_n \mid \forall P_i = (u, (V, E)), 0 \leq i \leq n - 1, \exists v \in V . l(v) \text{ contains } u' \wedge P_{i+1} = (u', t)$ .*

Roughly speaking, a webpage is reachable from another webpage if it is possible to follow a sequence of hyperlinks that connect both pages from the later to the former.

**Definition 5 (Search Hyperspace).** *Given a webpage  $P = (u, t)$  the search hyperspace of  $P$  is the set of webpages that either are reachable following hyperlinks from nodes of  $P$ , or that can reach  $P$  from their hyperlinks.*

The search hyperspace is the collection of webpages that are related to the initial webpage, and that should (ideally) be inspected by our information extraction algorithm. However, the search hyperspace of a webpage is potentially infinite (specially when we surf dynamic webpages [16]), and it is often huge. Therefore we need to reduce it by discarding some of the hyperlinks. In addition, we want our technique to be online. This implies that time response is a critical factor, but the analysis of a webpage implies loading it, which is a time-consuming task. Therefore, reducing the number of webpages that are analyzed is a major objective of the technique.

With this aim, we define an hyperDOM distance between nodes of the search hyperspace. This distance is used to decide what hyperlink nodes are more related to the query specified by the user and should be explored. The others are discarded. Using syntax distances to approximate semantic relations is an idea that is supported by experimental evaluation of different works. For instance, Micarelli and Gasparetti [17] obtained empirical results demonstrating that webpages pointed by closer hyperlinks are more related semantically than webpages

pointed by hyperlinks that are syntactically separated. In order to define an hyperDOM distance, we use the following concepts:

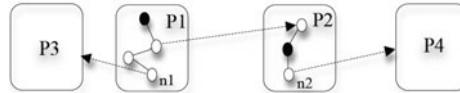
- **DOM distance ( $d_T$ ):** It is the length of the path between two nodes of a DOM tree.
- **Page distance ( $d_P$ ):** It is the lower number of hyperlinks that must be traversed to reach one webpage from another webpage.
- **Domain distance ( $d_D$ ):** It is the lower number of domains that must be traversed to reach one webpage from another webpage following a path of hyperlinks.

We use the initial webpage and the key nodes as the reference to compute distances. Therefore, for a given node, its DOM distance is the length of the path between this node and the closest key node in its DOM tree; and the page and domain distances are taken with respect to the initial webpage.

**Definition 6 (HyperDOM Distance).** *Given a DOM node  $n$ , the hyperDOM distance of  $n$  is  $D = d_T + K_P \cdot d_P + K_D \cdot d_D$  where  $K_P$  and  $K_D$  are numeric constants used to weight the equation. The significance  $S$  of a DOM node is the inverse of its hyperDOM distance  $S = 1/D$ .*

Constants  $K_P$  and  $K_D$  determine the importance that we give to the fact that the word specified by the user is in another page, or in another domain.

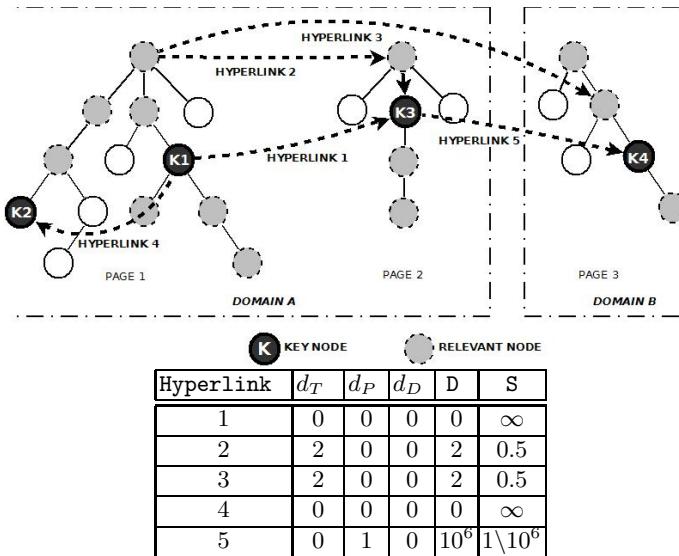
*Example 1.* Consider the following search hyperspace:



where two nodes contain the word specified by the user (those in black); the first node is in the initial webpage (P1), and the second node is in webpage P2 and thus it has a page distance of 1. Now, observe that nodes n1 and n2 are hyperlinks to other webpages. The question is: *which hyperlink is more related to the query of the user and should be explored first by the algorithm?* The answer is clear: the most relevant node and thus with a smaller hyperDOM distance. According to Definition 6, significance strongly depends on the values of the constants  $K_P$  and  $K_D$ . Assuming that all the webpages are in the same domain and if  $K_P = 1$ , then  $D(n1)=3$  and  $D(n2)=2$ , thus n2 is more significant. In contrast, if  $K_P = 10$ , then  $D(n1)=3$  and  $D(n2)=11$ , thus n1 is more significant.

After several experiments and intensive testing we took the following design decisions:

- 1 Those hyperlinks that are in the initial webpage are more important than those in another webpage. And the same happens as the page distance is increased. Hence, the DOM distance is more important than the page distance.



**Fig. 1.** Relevant information hyperlinked through different pages and domains

- 2 Those hyperlinks that are in the same domain as the initial webpage are more important than those in another domain. And the same happens as the domain distance is increased. Hence, the page distance is more important than the domain distance.
- 3 The algorithm should never analyze a webpage with a page distance greater than 5. This is also supported by previous studies (see, e.g., Baeza and Castillo's discussion in [16]) that demonstrate that, in general, three to five levels of navigation are enough to get 90% of the information which is contextually related with the webpage selected by the user for the web search.

Therefore, considering the amount of nodes in a webpage, we take the following values:  $K_P = 10^6$  and  $K_D = 10^9$ . The amount of DOM nodes in a webpage is usually lower than  $10^3$ , thus,  $10^6$  ensures that the distance of two different pages is always greater than the distance of two nodes in the same webpage. Similarly, the amount of webpages analyzed in our method is usually lower than  $10^2$ , thus,  $10^9$  ensures that the distance of two different pages analyzed in different domains is always greater than the distance of two different pages analyzed in the same domain. Hence,  $D = d_T + 10^6 \cdot d_P + 10^9 \cdot d_D$

*Example 2.* Consider an initial webpage P1 and its search hyperspace shown in Fig. 1. Assume that Algorithm 1 has analyzed the three webpages and thus, dark nodes are relevant (key nodes are black) and white nodes are discarded. In order to determine what hyperlinks are more relevant, we compute the significance of their DOM nodes (see the table). This information is used to decide what hyperlinks must be analyzed first. Observe in the example that the hyperDOM

distance of node  $k_4$  is  $0 + 1 * 10^6 + 1 * 10^9$ . This node has a lower significance because it is in another domain. Note also that the significance of hyperlinks is computed from the source node (even though a hyperlink relates two DOM nodes, the HTML element that represents the hyperlink is in the source).

In a DOM tree we can distinguish between hyperlinks that belong to the slice and hyperlinks that do not belong to the slice. Those hyperlinks that do not belong to the slice are often related to webpages of none interest for the user. Therefore, to ensure the quality of the retrieved information we take a fourth design decision:

#### 4 Hyperlinks that do not belong to the slice are discarded.

One important problem of extracting information from webpages happens in presence of dynamic webpages: A dynamic webpage could generate another dynamic webpage that contains the word specified by the user. This new dynamic webpage could do the same, and so on infinitely. This situation is known as black hole because robots searching in these webpages have an infinite search space where they always find what they are looking for. Therefore they are trapped forever if no limit is specified in the search [16]. Observe that the combination of design decisions 3 and 4 avoids this problem because the search is stopped when a webpage does not contain key nodes, or when its page distance is greater than 5. In addition, there is a fifth design decision related to the time response of the technique. Usability rules [18] establish that 10 seconds is (as an average) the time limit that users spend waiting for a webpage to be loaded. Therefore,

#### 5 The maximum time spent to retrieve and show the information is 10 seconds.

The time used to show the retrieved information is constant, but the time used to load a webpage is variable. Therefore, the technique uses a mechanism to iteratively load webpages in significance order and extract information from them. When the time spent is close to the limit, the technique must stop the analysis.

Algorithm 2 summarizes the technique for information extraction of interconnected webpages. It uses the following functions that implement the ideas and equations explained in this section: *timeout()* controls that the algorithm never runs more than 10 seconds<sup>1</sup>. When the time is over, it returns *True*. *getSlice()* computes a slice of a webpage with Algorithm 1. *show()* shows in the browser a collection of DOM nodes. It should be implemented in a way that visualization is incremental. *getLinks()* extracts the link nodes of a set of nodes. *getMostRelevantLink()* computes the hyperDOM distance of a set of nodes to determine what is the most relevant node. *load()* loads a webpage.

---

<sup>1</sup> 10 seconds is the default time used in our implementation, but it can be set to any value (e.g., hours). In this case, constants  $K_P$  and  $K_D$  are redefined to ensure that pages in different domains are farther (with the hyperDOM distance) than pages in the same domain.

**Algorithm 2.** Information extraction from multiple webpages**Input:** A set of interconnected webpages with an initial webpage  $P$ , and a query  $q$ **Output:** A webpage  $P'$ **Initialization:**  $currentPage = P$ ,  $pendingLinks = \emptyset$ 

```

while not(timeout())
  (1)  $relevantNodes = getSlice(currentPage, q)$ 
  (2)  $show(relevantNodes)$ 
  (3)  $pendingLinks = pendingLinks \cup getLinks(relevantNodes)$ 
  (4)  $link = getMostRelevantLink(pendingLinks)$ 
  (5)  $pendingLinks = pendingLinks / link$ 
  (6)  $currentPage = load(link)$ 

return  $P'$  (it is incrementally shown by the show function)

```

**2.1 Visualization of the Relevant Information**

Algorithm 2 is able to collect all the relevant DOM nodes of a set of webpages. Moreover, for each page, we know that the slice extracted is a valid webpage according to Algorithm 1. In addition, the information extracted is semantically related via hyperlinks and the semantic relation is weighted with the computed significance for each DOM node. Therefore, it is possible to use standard techniques for hierarchical visualization of the retrieved information. In our implementation reconstructing DOM trees is possible thanks to the DOM API's command:

```
documentNew.appendChild(documentOld.getElementById('myID'))
```

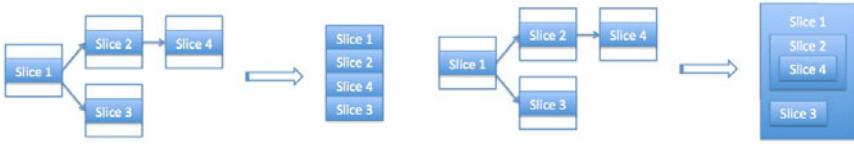
The command *documentOld.getElementById* allows us to extract from a DOM tree a specific element with a particular identifier *myID*. Then, the properties of this node can be queried, and if necessary, it can be inserted into another DOM tree with the command *documentNew.appendChild*. According to Algorithm 2, the visualization of the final webpage is done incrementally. For each analyzed webpage, we extract the slice with Algorithm 1, and then, this slice is inserted into the current webpage. Next, the webpage is refreshed, and thus, the technique produces results from the very beginning and, while the user inspects them, new results are added to the results webpage.

We have implemented two different algorithms to show the reconstructed webpage. The first one presents the information tabularly, the second one uses a hierarchical representation. Both algorithms retrieve information from different webpages and show it incrementally while it is being recovered. The main difference between them is the way in which the information is visualized in the browser.

**Tabular Visualization.** The lowest granularity level in this representation is a page. Basically, the final webpage is a linear succession of the filtered webpages. Each filtered webpage is considered as a whole, and thus, all the information that appeared together in the filtered webpage, is also together in the final webpage. The filtered webpages are ordered according to their navigational structure using a depth-first order.

**Hierarchical Visualization.** The lowest granularity level in this representation is a word. In this representation, the final webpage is a tree where the filtered webpages are organized. In contrast to the tabular representation, the filtered webpages can be mixed because each filtered webpage is placed next to the hyperlink that references it.

*Example 3.* Fig. 7 (left) shows a set of linked webpages where the dark part represents the relevant information, and its tabular representation of this relevant information. Fig. 7 (right) shows the hierarchical representation of the same set of webpages.



**Fig. 2.** Tabular visualization (left) and hierarchical visualization (right)

In Example 4 we show the complete process of information extraction.

*Example 4.* Consider again the initial webpage P1 and its search hyperspace of Fig. 1. Initially, Algorithm 2 extracts the slice of webpage P1. This slice is constructed from two key nodes (K1 and K2). Then, this information is shown to the user in a new webpage. Next, the algorithm tries to find the most relevant link to retrieve information from related webpages. In the table we see that the most relevant hyperlinks are H1 and H4. But H4 is discarded because it points to the initial webpage that has been already processed. Therefore, hyperlink 1 is selected and webpage P2 is loaded, processed and its slice shown to the user.

The information of webpage P2 is shown immediately after the information of K1, because, when this information is added to the webpage, it is placed close to the nodes that pointed to it. Hyperlink 2 is then discarded because it points to a webpage that has been already processed (P2). Because hyperlink 3 is more relevant than hyperlink 5, hyperlink 3 is selected first and webpage P3 is loaded, processed and shown to the user. Finally, hyperlink 5 is discarded because webpage P3 has been already processed. Hence, in the final webpage the slices are shown in order K1 K3 K4 K2.

Other models of visualization are possible and should be investigated. The presented models are designed to work in real-time because they work well when the amount of information shown is not too much (e.g., less than 20 slices). However, if the tool is used in batch mode (e.g., without time limitation), many webpages are filtered and the amount of information to be shown can be too much as to be shown in a single webpage; thus, it should be organized and probably indexed. For this, other models based on tiles [20] or clusters [19] would be more appropriate. Regarding the visualization of many slices, we are currently working

on a third visualization model called *site map*. Roughly, it produces an initial webpage with a site map with links that point to all the slices retrieved with the tool, and these slices are organized according to their original navigational map.

## 2.2 Implementation and Experiments

We have implemented the technique as an official plugin integrated in Firefox. The implementation allows the programmer to parameterize the technique in order to adjust the amount of information retrieved, the number of webpages explored, the visualization model and other functionalities. In order to determine the default configuration, it was initially tested with a collection of real webpages producing good results that allowed us to tune the parameters. Then, we conducted several experiments with real webpages. Concretely, we selected domains with different layouts and page structures in order to study the performance of the technique in different contexts (e.g., company's websites, news articles, forums, etc.).

For each domain, we performed two independent experiments. The first experiment provides a measure of the average performance of the technique regarding recall, precision and the F1 measure. The goal of this experiment was to identify the information in a given domain that is related to a particular query of the user. Firstly, for each domain, we determined the actual relevant content of each webpage by downloading it and manually selecting the relevant content (both text and multimedia objects). This task was performed by three different people without any help of the tool. The selected relevant content included all the information that each user thought was relevant for her. The DOM tree of the selected text was then built for each webpage. In a second stage, we used the tool to explore the webpages using Algorithm 2 and it extracted from them the relevant parts (according to the tool). Finally, we compared the DOM trees produced manually with those produced automatically by the tool. Table 1 summarizes the results obtained.

**Table 1.** Benchmark results

Domain	Query	Pages	Retrieved	Correct	Missing	Recall	Precision	F1
www.ieee.org	student	10	4615	4594	68	98.54 %	99.54 %	99.03 %
www.upv.es	student	19	8618	8616	232	97.37 %	99.97 %	98.65 %
www.un.org/en	Haiti	8	6344	6344	2191	74.32 %	100 %	85.26 %
www.esa.int	launch	14	4860	4860	417	92.09 %	100 %	95.88 %
www.nasa.gov	space	16	12043	12008	730	94.26 %	99.70 %	96.90 %
www.mityc.es	turismo	14	12521	12381	124	99 %	98.88 %	98.93 %
www.mozilla.org	firefox	7	6791	6791	14	99.79 %	100 %	99.89 %
www.edu.gva.es	universitat	28	10881	10856	995	91.60 %	99.79 %	95.51 %
www.unicef.es	Pakistán	9	5415	5415	260	95.41 %	100 %	97.65 %
www.ilo.org	projects	14	1269	1269	544	69.99 %	100 %	82.34 %
www.mec.es	beca	24	5527	5513	286	95.06 %	99.74 %	97.34 %
www.who.int	medicines	14	8605	8605	276	96.89 %	100 %	98.42 %
www.si.edu	asian	18	26301	26269	144	99.45 %	99.87 %	99.65 %
www.sigmaxi.org	scientist	8	26482	26359	241	99.08 %	99.54 %	99.30 %
www.scientificamerican.com	sun	7	5795	5737	97	98.33 %	98.99 %	98.65 %
ecir2011.dcu.ie	news	8	1659	1503	18	98.81 %	90.59 %	94.52 %
dsc.discovery.com	arctic	9	29097	29043	114	99.60 %	99.81 %	99.70 %
www.nationalgeographic.com	energy	12	41624	33830	428	98.75 %	81.27 %	89.16 %
physicsworld.com	nuclear	15	10249	10240	151	98.54 %	99.91 %	99.22 %

For each domain, the first column contains the URL of the initial webpage. Column **Pages** shows the number of pages explored by the tool in each experiment (the analysis time was limited to 10 seconds). Column **Query** shows the query used as the slicing criterion. Column **Retrieved** shows the number of DOM nodes retrieved by the tool; in the DOM model, the amount of words contained in a DOM node depends on the HTML source code of the webpage. It usually contains between one sentence and one paragraph. Column **Correct** shows the number of retrieved nodes that were relevant. Column **Missing** shows the number of relevant nodes not retrieved by the tool. Column **Recall** shows the number of relevant nodes retrieved divided by the total number of relevant nodes (in all the analyzed webpages of each domain). Column **Precision** shows the number of relevant nodes retrieved divided by the total number of retrieved nodes. Finally, column **F1** shows the F1 metric that is computed as  $(2 * P * R) / (P + R)$  being  $P$  the precision and  $R$  the recall.

The first important conclusion of the experiments is that, in 10 seconds, the tool is able to analyze 13,3 pages as an average for each domain. Therefore, because the visualization algorithms are incremental, the first result is shown to the user in less than 1 second (10/13,3 seconds).

Results show that the tool produces a very high recall and precision. We were not surprised by the high precision of the tool because the syntactic matches with the DOM nodes ensures that the information retrieved is often very related to the user's query. But we were very excited with the recall being so high. Only in a few cases the recall was below 75%. The cause was the occurrence of synonyms that the tool is currently ignoring. Our next release will include a lexicon to solve this problem. In ten seconds results are very good because the tool explores webpages that are close to the initial webpage, and, in this search space, it is able to accurately detect semantic relations between pages.

After these good results, we were wondering whether this tool could be also used to retrieve information in a batch process (i.e., without a time limit, analyzing as many pages as possible). In this context, we wanted to know what is the page coverage of the tool. For this, we conducted a second experiment in which we retrieved information from the domains allowing the tool to explore as much as possible (i.e., restrictions 3 and 5 were ignored). Then, we collected the amount of webpages analyzed by the tool and compared it with the amount of (reachable) webpages in the whole domain. The later was computed with the Apache crawler Nutch [22]: the whole domain was indexed starting from the initial webpage and the amount of indexed documents was counted. The result was that the tool explored, as an average, 30% of the webpages in the search space of all the domains in Table 1. The cause is that the technique automatically discards many hyperlinks and concentrates on the most relevant search space; this is due to restriction 4, that prevents the tool to explore those webpages pointed by other webpages without relevant nodes. Relaxing restriction 4 would allow the tool to explore the whole search space, but precision would (probably) decrease significantly, because it would retrieve information from different contexts.

All the information related to the experiments, including the source code of the tool and other material can be found at: <http://www.dsic.upv.es/~jsilva/webfiltering>

The official webpage of the tool at Firefox where the last stable release can be downloaded and where several comments and feedback from real users can be found at: <https://addons.mozilla.org/en-US/firefox/addon/web-filtering-toolbar>

### 3 Conclusions

This work introduces a novel information extraction technique based on syntax distances. The technique is able to work online and extract information from websites without any pre-compilation, labeling, or indexing of the webpages to be analyzed. Our experiments produced an F1 measure of 96%, demonstrating the usefulness of the technique. The analysis of the experimental results revealed that synonyms can cause a loss of recall. We are currently analyzing the impact of a lexicon. Using synonyms and semantic relations will allow us to increase the precision of our algorithms, but the efficiency of the technique will be affected. Empirical experimentation is needed to decide whether it is better to analyze many webpages without the use of a lexicon or few webpages with a lexicon. A balance between amount of information retrieved and the quality of this information must be studied. Our current implementation has been integrated in version 1.5 of the *Firefox WebFiltering Toolbar*. This tool is an official extension of the Firefox web browser that has been tested and approved by Firefox developers experts area, and that has more than 11.000 downloads at the time of writing these lines.

### References

1. Dalvi, B., Cohen, W.W., Callan, J.: Websets: Extracting sets of entities from the web using unsupervised information extraction. Technical report, Carnegie Mellon School of computer Science (2011)
2. Kushmerick, N., Weld, D.S., Doorenbos, R.: Wrapper induction for information extraction. In: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 1997) (1997)
3. Cohen, W.W., Hurst, M., Jensen, L.S.: A flexible learning system for wrapping tables and lists in html documents. In: Proceedings of the international World Wide Web conference (WWW 2002), pp. 232–241 (2002)
4. Lee, P.Y., Hui, S.C., Fong, A.C.M.: Neural networks for web content filtering. IEEE Intelligent Systems 17(5), 48–57 (2002)
5. Anti-Porn Parental Controls Software. Porn Filtering (March 2010), <http://www.tueagles.com/anti-porn/>
6. Kang, B.-Y., Kim, H.-G.: Web page filtering for domain ontology with the context of concept. IEICE - Trans. Inf. Syst. E90, D859–D862 (2007)
7. Henzinger, M.: The Past, Present and Future of Web Information Retrieval. In: Proceedings of the 23th ACM Symposium on Principles of Database Systems (2004)

8. W3C Consortium. Resource Description Framework (RDF), [www.w3.org/RDF](http://www.w3.org/RDF)
9. W3C Consortium. Web Ontology Language (OWL), [www.w3.org/2004/OWL](http://www.w3.org/2004/OWL)
10. Microformats.org. The Official Microformats Site (2009),  
<http://microformats.org>
11. Khare, R., Çelik, T.: Microformats: a Pragmatic Path to the Semantic Web. In: Proceedings of the 15th International Conference on World Wide Web, pp. 865–866 (2006)
12. Khare, R.: Microformats: The Next (Small) Thing on the Semantic Web? IEEE Internet Computing 10(1), 68–75 (2006)
13. Gupta, S., et al.: Automating Content Extraction of HTML Documents. World Wide Archive 8(2), 179–224 (2005)
14. Li, P., Liu, M., Lin, Y., Lai, Y.: Accelerating Web Content Filtering by the Early Decision Algorithm. IEICE Transactions on Information and Systems E91-D, 251–257 (2008)
15. W3C Consortium, Document Object Model (DOM), [www.w3.org/DOM](http://www.w3.org/DOM)
16. Baeza-Yates, R., Castillo, C.: Crawling the Infinite Web: Five Levels Are Enough. In: Leonardi, S. (ed.) WAW 2004. LNCS, vol. 3243, pp. 156–167. Springer, Heidelberg (2004)
17. Micarelli, A., Gasparetti, F.: Adaptative Focused Crawling. In: The Adaptative Web, pp. 231–262 (2007)
18. Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis (2010) ISBN 1-56205-810-X
19. Zhang, J.: Visualization for Information Retrieval. The Information Retrieval Series. Springer, Heidelberg (2007) ISBN 3-54075-1475
20. Hearst, M.A.: TileBars: Visualization of Term Distribution Information. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Denver, CO, pp. 59–66 (May 1995)
21. Gottron, T.: Evaluating Content Extraction on HTML Documents. In: Proceedings of the 2nd International Conference on Internet Technologies and Applications, pp. 123–132 (2007)
22. Apache Foundation. The Apache crawler Nutch (2010), <http://nutch.apache.org>

# Combining Flat and Structured Approaches for Temporal Slot Filling or: How Much to Compress?

Qi Li, Javier Artiles, Taylor Cassidy, and Heng Ji

Computer Science Department and Linguistics Department,  
Queens College and Graduate Center,  
City University of New York,  
New York, NY 11367, USA

{liqiearth, javart, taylorcassidy64, hengjicuny}@gmail.com

**Abstract.** In this paper, we present a hybrid approach to Temporal Slot Filling (TSF) task. Our method decomposes the task into two steps: temporal classification and temporal aggregation. As in many other NLP tasks, a key challenge lies in capturing relations between text elements separated by a long context. We have observed that features derived from a structured text representation can help compressing the context and reducing ambiguity. On the other hand, surface lexical features are more robust and work better in some cases. Experiments on the KBP2011 temporal training data set show that both surface and structured approaches outperform a baseline bag-of-word based classifier and the proposed hybrid method can further improve the performance significantly. Our system achieved the top performance in KBP2011 evaluation.

## 1 Introduction

There are many relations between named entities that may change over time (e.g. a person’s residence, an organization’s top employees, etc.), and these changes are expressed in the usage of temporal expressions in text. The TempEval evaluation campaigns [15] studied Temporal Information Extraction (TIE) concentrating on the identification of temporal expressions, events and temporal relations, but these tasks did not tackle the problem of finding the specific start and end dates for a given relation or event. In order to solve this problem a TIE system should detect whether a temporal expression actually concerns a certain relation, and in that case, the kind of role this temporal expression plays (i.e. whether it expresses the beginning of the relation, its end, or a time in between). Temporal information about a single relation can be scattered among different sentences and documents and presented with varying degrees of precision (e.g. a specific date, a range of dates such as a month, season, or year). To address these problems a system needs to detect coreferential mentions of the entities involved in a relation and aggregate the collected temporal information into a single answer. The NIST TAC Knowledge Base Population (KBP) 2011 track [9] included a pilot Temporal Slot Filling (TSF) task. In this task systems extract the start and end dates of a relation from a collection of documents. A relation involves an entity, a type of slot and a particular fill for that slot (e.g. “Nicole Kidman” is the slot fill for the entity “Tom Cruise” in the relation *spouse*).

As is the case for many NLP tasks, one of the main challenges of TSF lies in capturing relationships within long contexts. The context surrounding the entity, slot fills and temporal expressions is often too long and diverse to be generalized with surface features. By using syntactic parsing we can *compress* long contexts based on their underlying structure and capture common syntactic patterns. However, NLP tools that try to provide a deeper representation of the text can introduce many errors. Furthermore, in cases where there is a short context, surface features tend to provide a more appropriate generalization. One of the earliest works in IE asked “where is the syntax?” [8] and concluded that, although incorporating structure into an Information Extraction process would be necessary to overcome performance plateaus, only a conservative approach to parsing would be accurate enough to improve IE results without introducing too much noise. Our work re-visits the general hypothesis that using just the right amount of structure can improve IE results by evaluating the impact of features defined in terms of multiple levels of structure.

Our general approach to the TSF task is to decompose it into two problems: the classification of a temporal expression in the context of a query and its slot fill(s); and temporal information aggregation, in which the classified temporal expressions are combined to produce the start/end dates of a relation expressed by a given query and its slot fill. To this end, we have developed and tested two approaches to the temporal classification problem: a structured approach and a flat approach. The structured approach captures long syntactic contexts surrounding the query entity, slot fill and temporal expression using a dependency path kernel tailored to this task. The flat approach exploits information such as the lexical context and shallow dependency features. We show that these two approaches are complementary and thus can be combined through a simple temporal information aggregation process. We also show that the performance of each approach is highly correlated with the length of the example contexts. Our proposed approaches outperform a number of baselines, and achieved the top performance in the KBP2011 evaluation.

## 2 Related Work

The need for structural representations is acknowledged in many Natural Language Processing fields. For example, the shortest path between two vertices in a dependency parsed graph has been used to capture the syntactic and semantic relation between two words [2,16,18].

Temporal IE has recently attracted intensive research interests. For example, the TempEval [15,17] shared tasks has aimed at the extraction of relations between events and temporal expressions from single documents. Various approaches have been developed for this task, which can be roughly categorized into flat or structured approaches: (1) **Flat approaches:** [3] built a supervised learning model to classify a pair of event triggers in a sentence based on syntactic and semantic features at the lexical level based on tense and aspect. [19] used similar features, using a Markov Logic based joint inference framework for temporal relations. [10] also exploited cross-event joint inference, but they used shallow dependency features to build local formulas without considering the deeper syntactic structure. (2) **Structured approaches:** [1] designed syntactic

and semantic features based on syntactic treelets and verbal government for temporal relation classification. [14] used sentence level syntactic trees to propagate temporal relations between syntactic constituents. [5] introduced a type of feature called *tree position* that classifies nodes on a syntactic dependency tree based on their position in the tree relative to that of a target node.

Our work advances temporal IE in the following three main aspects: (1) it extends the notion of temporal relation from that of a pair  $\langle \text{event}, \text{time expression} \rangle$ , or  $\langle \text{event} \rangle$ , to a 3-tuple  $\langle \text{entity}, \text{relation/event}, \text{time expression} \rangle$ , allowing us to capture temporal information of varying degrees of uncertainty; (2) We represent the contexts surrounding the tuple elements, using both flat and structural features; (3) it extends from single-document extraction to cross-document extraction so that we are able to effectively combine the advantages from flat and structured approaches through cross-document information aggregation.

### 3 Experimental Setup

#### 3.1 Task Definition

The goal of KBP2011 temporal slot filling task is to add temporal information to selected slots for a given entity query from a large collection of documents. The slot types considered on this task are: spouse, title, employee\_of, member\_of, cities\_of\_residence, state or provinces\_of\_residence and countries\_of\_residence for people, and the top-employees/members slot for organizations. There are two subtasks: full and diagnostic. For the full temporal task, the system is given an entity name and a document where this entity is mentioned and is expected to find the relevant slots in the document collection, augmented with temporal information as described below. For the diagnostic temporal task, the system is given the entity and a set of slot values with their types and the documents that support them. For this task the system should determine the temporal information for each slot value, based only on the information in the provided support document. In order to investigate the capability of various approaches to temporal information extraction, we conduct experiments in the diagnostic setting.

#### 3.2 Temporal Representation

The KBP2011 temporal representation model consists of a 4-tuple whose elements are dates (day, month and year),  $\langle t_1, t_2, t_3, t_4 \rangle$ . A tuple represents the set of possible beginnings and endings of an event.  $t_1$  and  $t_2$  represent the lower and upper bounds, respectively, for the beginning of the event, while  $t_3$  and  $t_4$  represent the lower and upper bounds for end of the event.

Given a slot-filling query name *Jose Padilha*, its slot fill *Film Maker* for the slot type *per:title*, a diagnostic temporal slot filling system may discover a temporal tuple  $\langle -\infty, 2007-12-26, 2007-12-26, +\infty \rangle$  to represent the temporal boundaries.

### 3.3 Scoring Metric

We use the official scoring metric  $Q(S)$  for the task. This metric compares a system's output  $S = \langle t_1, t_2, t_3, t_4 \rangle$  against a gold standard tuple  $S_g = \langle g_1, g_2, g_3, g_4 \rangle$ , based on the absolute distances between  $t_i$  and  $g_i$ :

$$Q(S) = \frac{1}{4} \sum_i \frac{1}{1 + |t_i - g_i|}$$

When there is no constraint on  $t_1$  or  $t_3$  a value of  $-\infty$  is assigned; similarly a value of  $+\infty$  is assigned to an unconstrained  $t_3$  or  $t_4$ .

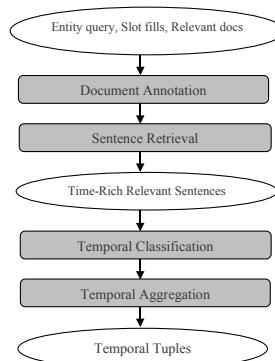
Let  $\{G^1, G^2, \dots, G^N\}$  be the set of gold standard tuples,  $\{S^1, S^2, \dots, S^M\}$  the set of system output tuples, where for each unique slot fill  $i$ ,  $G^i$  there is the 4-tuple  $\langle g_1, g_2, g_3, g_4 \rangle$ , and  $S^j$  is the 4-tuple  $\langle t_1, t_2, t_3, t_4 \rangle$ . Then Precision, Recall and  $F_1$ -measure scores are calculated as follows:

$$\text{Precision} = \frac{\sum_{S^i \in C(S)} Q(S^i)}{M} \quad \text{Recall} = \frac{\sum_{S^i \in C(S)} Q(S^i)}{N} \quad F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where  $C(S)$  is the set of all instances in system output which have correct slot filling answers, and  $Q(S)$  is the quality value of  $S$ . In the diagnostic task, precision, recall, and  $F_1$  values are the same since we are provided with correct slot filling values as part of the system input.

## 4 Approach Overview

Our approach to the TSF problem consists of three main steps: (i) find all the contexts where the entity and the slot value are mentioned; (ii) classify each temporal expression in those contexts according to its relation with the entity/slot value pair; (iii) aggregate all the classified temporal information in a coherent 4-tuple. In Figure 1 we summarize the system pipeline. Our system takes as input an entity, slot type and slot value as well as the source documents where the slot value was found.



**Fig. 1.** General Temporal Slot Filling System Architecture

Each source document is fully processed using the Stanford NLP Core toolkit [7] to tokenize, detect sentence boundaries, detect named entities (including temporal expressions), build coreference chains and analyze the syntactic dependencies within sentences. The annotated output is used to find sentences that mention both the entity and the slot value. Finding these sentences by string matching provides only very limited coverage, so we use named entity recognition and coreference results to expand this set of relevant sentences. We look at the coreference chains that contain the provided slot value or entity name and we select sentences that mention both, according to the coreference chains.

Each temporal expression in these sentences is then represented as a classification instance and labeled as belonging to one of the following classes: start, end, hold, range and none. Finally, for each particular entity/slot value, all of its classified temporal expressions are aggregated in a single 4-tuple.

#### 4.1 Temporal Classification

Classification is applied to label temporal expressions that appear in the context of a particular entity and the slot value as “start”, “end”, “hold”, “range” or “none”. Suppose our query entity is *Smith*, the slot type is *per:title*, and the slot-fill is *Chairman*. Table 1 shows a description of each class along with its corresponding 4-tuple representation:

**Table 1.** Description of Temporal Classes

Class	Temporal Role	Four tuple
Start	beginning of the slot fill	$< t_a, t_b, t_a, \infty >$
End	end of the slot fill	$< -\infty, t_b, t_a, t_b >$
Hold	a time at which the slot fill is valid	$< -\infty, t_b, t_a, \infty >$
Range	a range in which the slot fill is valid	$< t_a, t_a, t_b, t_b >$
None	unrelated to the slot fill	$< -\infty, \infty, -\infty, \infty >$

The next two subsections describe the two classification approaches we have tested.

#### 4.2 Flat Approach

The flat approach uses two types of features: window features and dependency features. A window feature value for the query entity, slot value, and a target temporal expression is extracted from each example. This value is a set containing all tokens that occur in the normalized sentence within 4 tokens in either direction of any instance of the normalized token in question.

Two dependency feature values for the query entity, slot value, and a target temporal expression are extracted from each example, resulting in two sets of tokens for each normalized token  $T$ . One set contains all tokens that any instance of  $T$  governs, the other set contains all tokens governed by any instance of  $T$ . Before a feature value set for a normalized token  $T$  is created, punctuation marks, duplicate consecutive normalized tokens, and instances of  $T$  itself are removed.

Example (1) is from the evaluation set, for the query, attribute = *per:title*, entity = *Makoni*, slot fill = *minister of industry and energy development*. (1') is its normalized version.

(1) In 1981, Makoni was moved to the position of minister of industry and energy development, where he remained until 1983.

(1') In DATE, TE was moved to the position of TA, where he remained until TD.

Table 2 shows the feature values extracted from (1').

**Table 2.** Feature Values for (1)

Feature	Value
TE Win	be, move, to, in, DATE, the
TA Win	of, to, remain, position, the, where, until, he
TD Win	remain, where, until, he
TE Governs	-
TA Governs	-
TD Governs	-
TE Governed by	move
TA Governed by	position
TD Governed by	remain

For two feature values  $U, V$ , let  $K_T$  be the normalized size of their intersection

$$K_T(U, V) = \frac{|U \cap V|}{\sqrt{|U|^2 + |V|^2}} \quad (1)$$

Let  $F$  denote the flat features. Then for any  $G \subseteq F$ , let  $K_S$  be the kernel function for a pair of examples, and  $x.i$  the feature value for the  $i^{th}$  feature value type for example  $x$ :

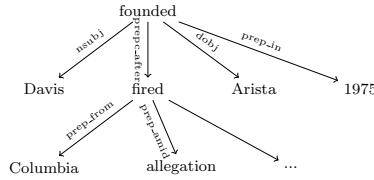
$$K_S(x, y) = \sum_{i \in G} K_T(x.i, y.i) \quad (2)$$

With these features we trained a classifier using Support Vector Machines (SVM) [6].

### 4.3 Structured Approach

**Dependency Path Representation.** In the structured approach, we exploit collapsed dependency parsed graphs generated from the Stanford dependency parser [12] to capture relevant grammatical relations and discover syntactic patterns. Figure 2 shows a part of the dependency graph obtained from the sentence, “*In 1975*<sub>[time expression]</sub>, *after being fired from Columbia amid allegations that he*<sub>[query entity]</sub> *used company funds to pay for his*<sub>[query entity]</sub> *son’s bar mitzvah, Davis*<sub>[query entity]</sub> *founded Arista*<sub>[slot fill]</sub>” . In this example, *Davis* is the query entity, the slot type is *per:employee\_of*, *Arista* is the slot fill, and *1975* is the time expression.

We extend the idea of shortest path on a dependency graph (see Section 2) to include three items: query entity, slot fill and time expression. Each instance is represented by

**Fig. 2.** Dependency parsed graph of the sample sentence

three paths: (i) the path between query entity and temporal expression ( $P_1$ ), (ii) the path between slot fill and temporal expression ( $P_2$ ); and (iii) the path between query entity and slot fill ( $P_3$ ).

We found that two modifications in the graph allow us to obtain more informative paths. To capture phrasal verbs, take for example *took over* as one node in the path instead of only using *took*, we change two vertices linked by *prt* dependency into one vertex, where *prt* indicates a phrasal verb relation. Second, consider dependency path between *he* and *president*, sentence “*he was president of ...*” and “*he is president of ...*” produce same path, because *he* and *president* are linked by *nsubj*, where *nsubj* indicates subject relation. To address this issue, we reshaped the dependencies around copula verb such as *is* and *become* to those of common verbs.

Each shortest path  $P_i$  is represented as a vector  $\langle t_1, t_2, \dots, t_n \rangle$ , where  $t_i$  can be either a vertex or a typed edge in the dependency graph. Each edge is represented by one attribute, which is formed by combining the corresponding dependency type and direction. More formally, attribute  $a \in \mathcal{D} \times \{\leftarrow, \rightarrow\}$ , where  $\mathcal{D}$  is the set of dependency types, and the arrow is directed from the governor to the dependent word. Vertices, on the other hand, may contain different levels of features, which can be found in Table 3.

**Table 3.** Features of Vertices

Feature	Description
Word	The original word token from the sentence. E.g., “Davis <i>founded</i> <sub>[<i>founded</i>]</sub> Arista”
Stem	Stemmed form of the word token. E.g., “Davis <i>founded</i> <sub>[<i>found</i>]</sub> Arista”
Entity type	Person, Location, Organization. E.g., “fired from Columbia <sub>[Organization]</sub> ”
Semantic class of trigger words	Each class contains trigger words of event subtype in Automatic Content Extraction 2005 corpus <sup>1</sup> , and some manually collected slot type sensitive key words. E.g. if slot type is <i>per:spouse</i> , then the word <i>marry</i> belongs to one semantic class while <i>divorce</i> belongs to another semantic class.
Part-of-speech	Part-of-speech tag of original word

<sup>1</sup> <http://projects.ldc.upenn.edu/ace/>

For example, in the sentence of Figure 2, there exists *prep\_in* dependency from *founded* to *1975*. *prep\_in* represents prepositional relation between these two words, meaning that the action *founded* happened at *1975*.

When we search the shortest path between two nodes, we consider all mentions of the query entity and the slot fill in a sentence. For this reason there could be more than one candidate for each  $P_i$ . We define the following simple but effective strategy to choose one path among all candidate paths. If some candidate paths contain predefined trigger words, we choose the shortest path with trigger words. Otherwise, we choose the shortest path among all candidates.

Figure 3 shows three shortest paths that result from the sentence of Figure 2. These paths not only contain lexical features such as words, but also syntactic relations. In the resulting representations, informative patterns are distilled while some irrelevant information, as well as misleading words such as *fire*, are discarded.

The next step in our system is to use a kernel function to generalize these paths and represent them in a high dimensional feature space implicitly.



**Fig. 3.** Three Shortest Paths from Figure 2

**Kernel Function.** Following previous work [11] and [2], we present a string kernel function based on dependency paths. The main idea is to use the kernel trick to deal with common-substring similarity between dependency paths, and to extract syntax-rich patterns from dependency paths.

Let  $x, y$  be two instances. We use  $l(P)$  to denote the length of a dependency path  $P$ ,  $P[k]$  to denote the set of all substrings of  $P$  which have length  $k$ , and a substring  $a \in P[k]$  is a substring of  $P$  with length  $k$ . For example, if  $P$  is “ABC”, then  $P[2] = \{“AB”, “BC”\}$ . The kernel function of  $x$  and  $y$  is defined as follows:

$$K_s(x, y) = \sum_{i=1}^3 K_p(x.P_i, y.P_i) \quad (3)$$

$$K_p(P_x, P_y) = \sum_{k=1}^{\min(l(P_x), l(P_y))} \sum_{a \in P_x[k], b \in P_y[k]} \prod_{i=1}^k c(a_i, b_i) \quad (4)$$

Where  $K_p$  is a kernel function on two dependency paths  $P_x$  and  $P_y$  which sums the number of common substrings of feature paths in  $P_x$  and  $P_y$  with length from 1 to the maximum length. In  $c(a_i, b_i)$  we calculate the inner product of the attribute vectors of  $a_i$  and  $b_i$ , where  $a_i$  and  $b_i$  are elements of two paths respectively. The final kernel function  $K_s$  does the summation of the partial results of the three dependency paths (query entity-slot fill, query entity-temporal expression, slot fill-temporal expression).

Consider the following example containing two dependency paths  $P_x$  and  $P_y$  between an entity (E) and a temporal expression (T) in two different sentences.

$$\begin{array}{c} E \xrightarrow{\text{nssubj}} \text{founded/} \text{found/VBD/} \text{Start-Position} \xleftarrow{\text{prep,in}} T \\ E \xrightarrow{\text{nssubj}} \text{joined/} \text{join/VBD/} \text{Start-Position} \xleftarrow{\text{prep,in}} T \end{array}$$

For instance, if we consider substrings of length 5 we find the following two matches:

$$\begin{array}{c} E \xrightarrow{\text{nssubj}} \text{VBD} \xleftarrow{\text{prep,in}} T \\ E \xrightarrow{\text{nssubj}} \text{Start-Position} \xleftarrow{\text{prep,in}} T \end{array}$$

By counting the common substrings for the remaining lengths (1 to 4) we can obtain the final result:  $K_p(P_x, P_y) = 26$ .

A problem of Equation (4) is that  $K_p$  has a bias toward longer dependency paths. To avoid this bias, we normalize  $K_p$  as in [11]. This normalization scales the feature vector  $\phi(P)$  in the kernel space to  $\phi'(P) = \frac{\phi(P)}{|\phi(P)|}$ :

$$K'_p(P_x, P_y) = \frac{K_p(P_x, P_y)}{\sqrt{K_p(P_x, P_x) \cdot K_p(P_y, P_y)}} \quad (5)$$

A deviation from related work in [11] and [2] is that we count common substrings from  $m$  to maximum, rather than a fixed length. Furthermore, we only consider contiguous substrings in  $K_p$  because each substring feature in the kernel space is treated as a pattern. Non-contiguous substrings with the same length can be safely discarded as different patterns.

Although it's not easy to enumerate all substrings explicitly, like many other kernel functions,  $K_p$  can be efficiently computed by using dynamic programming in polynomial time complexity. Here, we applied a variant of the Levenshtein Distance algorithm to calculate  $K_p$ . Given the representation and kernel function, SVM model was applied to train a classifier.

#### 4.4 Temporal Aggregation

In order to produce the final 4-tuple for each entity/slot value pair, we sort the set of the corresponding classified temporal expressions according to the classifier's prediction confidence. We initialize a 4-tuple to  $< -\infty, +\infty, -\infty, +\infty >$  and then iterate through that set, aggregating at each point the temporal information as indicated by the predicted label (see Section 4.1). Given two four-tuples  $T$  and  $T'$ , we use the following equation for aggregation.

$$T \wedge T' = < \max(t_1, t'_1), \min(t_2, t'_2), \max(t_3, t'_3), \min(t_4, t'_4) >$$

At each step we modify the tuple only if the result is consistent (i.e.  $t_1 \leq t_2, t_3 \leq t_4$ , and  $t_1 \leq t_4$ ).

Furthermore, we utilize 4-tuple aggregation to combine outputs from the flat classifier, which uses shallow syntactic features, with that of the structured classifier, which uses deep syntactic features. We hypothesize that these two systems are complementary when combined in this way. Given an input, we consider the output from the structured classifier  $T$  as the default output. If one element of the output equals  $-\infty$  or  $\infty$ , then we combine it with output from flat classifier  $T'$  as final output.

## 5 Experiments

### 5.1 Automatic Training Data from Distant Supervision

Given the expensive nature of human-assessed training data for this task, we adapted a distant supervision approach [13] to obtain large amount of training data from the Web without human intervention.

We use Freebase<sup>2</sup> to gather not only instances of relations, but also the start and end dates of those particular relations. We can still follow the usual distant supervision assumption: given a context that mentions the query entity and slot fill it is likely that it will express the relation in the database. But our methods go beyond the usual distant supervision in that we incorporate an additional element, the temporal expression. We assume that we can label a temporal expression occurring in the context of the entity/slot fill pair by comparing it to the start/end temporal information that is stored in our database. We obtained through this method more than 40,000 training instances with no human intervention.

### 5.2 Overall Results

To evaluate the performance of different approaches, we use the KBP 2011 temporal slot filling training data as test set. This data set contains 430 query entity names, and 748 slot fills and corresponding temporal four-tuples. In the experiments, we used LIB-SVM library [4]<sup>3</sup> to train SVM classifiers.

**Table 4.** Overall Performance

System	Overall	Employee_of	City	State	Country	Memebr_of	Title	Top_members	Spouse
<i>BoW</i>	0.638	0.637	0.781	0.525	0.662	0.582	0.702	0.510	0.438
<i>Structured</i>	0.667	0.674	0.844	<b>0.675</b>	<b>0.766</b>	0.627	0.702	0.538	0.556
<i>Flat</i>	0.663	0.657	0.844	0.661	0.707	0.613	0.707	0.544	0.570
<i>Combine</i>	<b>0.678</b>	<b>0.681</b>	<b>0.865</b>	<b>0.673</b>	0.721	<b>0.628</b>	<b>0.720</b>	<b>0.545</b>	<b>0.862</b>

We compared the performance of the proposed combination approach against *Structured*, *Flat*, and *BoW*. The baseline *BoW* uses bag-of-words representation and linear kernel on top of sentence normalization to represent each instance. Table 4 shows overall performance with breakdown scores for each slot type. Compared to other approaches, *BoW* achieves the lowest performance. Although the advantage of the structured approach against the flat approach is subtle, the combined system outperforms both of them, and achieves the highest scores in 7 slot types. We conducted the Wilcoxon Matched-Pairs Signed-Ranks Test on a four-tuple basis. The results show that the improvement of the combined system is significant at the 99.8% confidence level when compared with the structured approach, and at the 99.9% confidence level compared with the flat approach.

<sup>2</sup> <http://www.freebase.com>

<sup>3</sup> <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

## 6 Discussions

For many NLP tasks including this new TSF task, one main challenge lies in capturing long contexts. Semantic analysis such as dependency parsing can make unstructured data more structured by *compressing* long contexts and thus reduce ambiguities. However, current core NLP annotation tools such as dependency parsing and coreference resolution are not yet ideal for real applications. The deeper the representation is, the more risk we have to introduce annotation errors. Furthermore, for certain types of slots such as “title”, since the contexts are relatively short between the query and its slot fill (e.g. “Today[Time] President[Title] Obama [Query]...”), structured representation is not appropriate. Therefore we pursued a more conservative approach combining benefits from both flat approach (local context, short dependency path, etc.) and structured approach (e.g. dependency path kernel). We reported that the structured approach outperforms the flat approach in general except slot types involve shorter contexts. Furthermore, combining them through cross-document temporal aggregation can achieve higher performance than each approach alone.

For example, there is a long context between the query “Mugabe”, the time expression “1980” and its slot fill “ZANU-PF” in the following sentence “ZANU, which was renamed **ZANU-PF** after taking over ZAPU, has been the country’s ruling party and led by **Mugabe** since **1980**.<sup>7</sup>” The structured approach successfully identified “1980” as the starting date based on the short dependency paths among “ZANU-PF”, “Mugabe” and “Mugabe”.

On the other hand, dependency parsing can produce errors. For example, it failed to capture the dependency relation between “September 2005” and “the Brookings Institute” in the following sentence “In **September 2005**, **Dichter** left office and became a research fellow at **the Brookings Institute** in Washington , D.C.”. In contrast the flat approach can easily identify “September 2005” as the starting date for the query “Avi Dichter” to be a member of “the Brookings Institute” based on lexical features such as “became”.

We also found that the gains by the structured approach are highly correlated with the compression rate, which is defined by (1 - the lengths of dependency paths among [query, slot fill, time expression] divided by the number of context words). For example, using structured approach they achieved much higher gains on residence slots (about 0.78 compression rate) than title (about 0.68 compression rate).

## 7 Conclusions and Future Work

In this paper, we presented a hybrid approach to diagnostic temporal slot filling task. We decompose the task into two steps: temporal classification and temporal aggregation. First, two approaches are developed for temporal classification: a flat approach that uses lexical context and shallow dependency features and a structured approach that captures long syntactic contexts by using a dependency path kernel tailored for this task. Following the hypothesis that these two approaches are complementary, we then combine them by aggregation as a hybrid approach. Experiment results show that the individual flat and structured approaches both outperform bag-of-word based classifier,

and the proposed hybrid method can further improve the performance significantly. In the future we are particularly interested in conducting cross-query and cross-slot temporal reasoning to enhance the performance.

## References

1. Bethard, S., Martin, J.H.: Cu-tmp: Temporal relation classification using syntactic and semantic features. In: SemEval 2007: 4th International Workshop on Semantic Evaluations (2007)
2. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proc. of the HLT and EMNLP, pp. 724–731 (2005)
3. Chambers, N., Wang, S., Jurafsky, D.: Classifying temporal relations between events. In: Annual Meeting of the Association for Computational Linguistics (ACL), pp. 173–176 (2007)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001)
5. Cheng, Y., Asahara, M., Matsumoto, Y.: Naist.japan: Temporal relation identification using dependency parsed tree. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007), pp. 245–248 (2007)
6. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning, 273–297 (1995)
7. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by gibbs sampling. In: ACL (2005)
8. Grishman, R.: The NYU System for MUC-6 or Where's the Syntax? In: Proceedings of the MUC-6 workshop (2010)
9. Ji, H., Grishman, R., Dang, H.T., Li, X., Griffitt, K., Ellis, J.: An Overview of the TAC2011 Knowledge Base Population Track. In: Proc. Text Analytics Conference (TAC 2011) (2010)
10. Ling, X., Weld, D.: Temporal information extraction. In: Proceedings of the Twenty Fifth National Conference on Artificial Intelligence (2010)
11. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. The Journal of Machine Learning Research 2, 419–444 (2002)
12. Marneffe, M.C.D., Maccartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: LREC 2006 (2006)
13. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL/AFNLP, pp. 1003–1011 (2009)
14. Puşcaşu, G.: Wvali: Temporal relation identification by syntactico-semantic analysis. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007), pp. 484–487 (2007)
15. Pustejovsky, J., Verhagen, M.: Semeval-2010 task 13: Evaluating events, time expressions, and temporal relations (tempeval-2) (2010)
16. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in Neural Information Processing Systems, vol. 17, pp. 1297–1304 (2005)
17. Verhagen, M., Gaizauskas, R., Schilder, F., Katz, G., Pustejovsky, J.: Semeval2007 task 15: Tempeval temporal relation identification. In: SemEval 2007: 4th International Workshop on Semantic Evaluations (2007)
18. Wu, F., Weld, D.S.: Open Information Extraction using Wikipedia. In: Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (2010)
19. Yoshikawa, K., Riedel, S., Asahara, M., Matsumoto, Y.: Jointly identifying temporal relations with markov logic. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 405–413 (2009)

# Event Annotation Schemes and Event Recognition in Spanish Texts

Dina Wonsever<sup>1</sup>, Aiala Rosá<sup>1</sup>, Marisa Malcuori<sup>2</sup>, Guillermo Moncecchi<sup>1</sup>,  
and Alan Descoins<sup>1</sup>

<sup>1</sup> Instituto de Computación

Facultad de Ingeniería

Universidad de la República, Uruguay

{wonsever, aialar, gmonce}@fing.edu.uy, dekked@gmail.com

<sup>2</sup> Instituto de Lingüística, Uruguay

Facultad de Humanidades y Ciencias de la Educación

Universidad de la República

marisamalcuori@gmail.com

**Abstract.** This paper presents an annotation scheme for events in Spanish texts, based on TimeML for English. This scheme is contrasted with different proposals, all of them based on TimeML, for various Romance languages: Italian, French and Spanish. Two manually annotated corpora for Spanish, under the proposed scheme, are now available. While manual annotation is far from trivial, we obtained a very good event identification agreement (93% of events were identically identified by both annotators). Part of the annotated text was used as a training corpus for the automatic recognition of events. In the experiments conducted so far (SVM and CRF) our best results are in the state of the art for this task (80.3% of F-measure).

## 1 Introduction

The fact of processing texts, no matter the purpose of such task, involves dealing with certain properties of the discourse that need to be grasped. We have chosen to adopt a modular structure to account for these properties, expressing them by means of the analysis of different independent axes, nevertheless able to interact with each other. Even though this structure does not provide, in principle, a holistic view of the discourse, it does allow to work independently in each axis, while it enables others to be added, as they develop.

The proposed analysis axes are: Enunciation, Events-Factivity, Temporality, Rhetorical Structure. Two more axes of structural nature are added to these four: Syntax axis and Textual Structure (paragraph, section, title, etc.) axis. The analysis for each one of the first four modules or axes is expressed in an annotation scheme for corpus annotation. Machine Learning techniques are applied upon these annotated corpora in order to generate a discourse analyzer. In this work we present the results of a set of tasks performed within the *Events-Factivity* module. We propose an event

annotation scheme based on TimeML (called SIBILA), we contrast this scheme with other proposals for Romance languages and we report the results obtained in the automatic recognition of events.

## 2 Event Annotation on Texts

### 2.1 Definition of Event

A core aspect in the computational understanding of a text is the detection of event references, as they constitute the minimal units with propositional content. Events can be actions (carried out voluntarily by an agent), processes (events spontaneously set off or caused by a force external to the process, which can, in both cases, be punctual or have duration), or states (situations maintained along a period or that are permanent). Generic predication will also be considered as events for they refer to states of things, states about which it is asserted that they take place.

Even though the events are in general indicated by verb forms, there also exist nouns that designate events. These event nouns do not designate objects (whether physical or abstract) but occurrences or incidents as in the case of *accidente* [accident], *batalla* [battle], *cena* [dinner], *eclipse* [eclipse], *desfile* [parade], *muerte* [death], *nacimiento* [birth], *temporada* [storm], among many others.

While the verb category, whether in a finite form or not, is a powerful indication for detecting events, clear morphosyntactic indicators are missing for nominal events. Also, under the same form it is possible to interpret a noun as denoting an event or an object: *El concierto empieza a las ocho. / El concierto en si menor para violonchelo* [The concert starts at eight. / Cello concerto in B minor]; *Durante la construcción se presentaron varios problemas. / La construcción data del siglo XIX* [Several problems arose during the construction. / The construction dates from the 19<sup>th</sup> century]. This ambiguity constitutes a difficulty for automatic recognition. Nevertheless, there exists a series of syntactic indications that help to recognize this kind of nouns: co-occurrence with verbs such as *tener lugar* [to take place] or *presenciar* [to witness]; with verbs or expressions indicating duration or aspectual phase such as *empezar* [to start], *comenzar* [to begin], *concluir* [to finish], *terminar* [to end], *durar* [to last], as it is shown in (1):

- (1) *Esto sucedía después de que se mirara con buenos ojos el fin del corte en Gualeguaychú llevado a cabo sobre las 14 horas de la tarde de ayer. [This was happening after the end of the roadblock in Gualeguaychú carried out around 14.00 hours yesterday afternoon was well regarded.]*

Besides, events can be expressed by means of other categories such as adjective, prepositional phrase, given that states can be designated by means of them, and also by the pronoun category when the referent is an event.

## 2.2 Annotation Scheme

The annotation scheme SIBILA, which dates from 2008, is an adaptation to Spanish of the TimeML scheme [12, 17]. Beyond the fact that adaptation is not a trivial task, the SIBILA scheme incorporates some innovative elements, the most important of which is the *factivity* attribute and its values. Starting from the SIBILA scheme a detailed annotation guide with lots of examples was made in order to guide annotators [24] and, likewise, reasons for the study of factivity and its relevant values [25] were established.

There currently exist other adaptations of TimeML for Romance languages such as Italian [7] and French [3], and there is also a Spanish version proposed by the TimeML team [20].

The adaptation for Spanish by means of the SIBILA scheme shares some attributes incorporated in the schemes above mentioned and it also includes, besides the *factivity* attribute, other changes about which we are going to briefly speak about next. Anyway, the SIBILA scheme is consistent with the proposal of TimeML.

Even though the scheme establishes, in addition to events, the annotation of other elements such as different kinds of indexes, aspectual and subordination links between events, temporal expressions and temporal links, in this occasion we will only refer to the events.

### Events and Their Attributes

A complete description of the *event* element is presented next, followed by the analysis of differences and similarities with regard to the rest of the schemes based on TimeML. Table 1 shows the event attributes and their values.

**Table 1.** Event attributes

Attribute	Value
<i>id</i>	unique identifier
<i>class</i>	OCCURRENCE   PERCEPTION   REPORT   ASPECT   STATE   INTENSIONAL_CAUSAL_ACTION   INTENSIONAL_STATE   EXISTENCE
<i>category</i>	VERB   NOUN   ADJECTIVE   PRONOMINAL   OTHER
<i>verb_form</i>	INFINITIVE   GERUND   PARTICIPLE   FINITE_FORM
<i>mood</i>	INDICATIVE   SUBJUNCTIVE   CONDITIONAL   IMPERATIVE

**Table 1.** (continued)

<i>time</i>	PAST   PRESENT   FUTURE
<i>determination</i>	DEFINITE   INDEFINITE   BARE
<i>modality</i>	Lexical item of a modality operator (free text)
<i>polarity</i>	NEG   POS
<i>factivity</i>	YES   NO   PROGRAMMED_FUTURE   NEGATED_FUTURE   POSSIBLE   INDEFINITE
<i>indexes</i>	references to indexes (ids)
<i>lex_item</i>	free text (CDATA)
<i>comments</i>	free text (CDATA)

As in the schemes proposed for French and Italian and in the Spanish version of TimeML, in SIBILA, *mode* and *verb form* attributes are incorporated in order to account for the flexive complexity of Romance languages. However, a significant difference shown by SIBILA relates to the value of the *time* attribute. Beyond the tense value assigned to finite forms by the tagger, the *time* attribute will take the value of PAST, PRESENT or FUTURE accordingly with the meaning that the verb form may have in the text in which it appears. So, it represents the semantic temporal value and not the syntactic tense value. For instance, a verb form like *descubre* [*discovers*] in *Colón descubre América en 1492* [*Colon discovers America in 1492*] will have for the time attribute the PAST value, even if it is a present verb form.

On the other hand, SIBILA incorporates the EXISTENCE value for the *class* attribute. In this way, it treats the copulative, existential and presentative verbs as events that operate predicating others' event existence. That is to say, when an event referred by a noun, an adjective or a prepositional phrase is part of a predicate with copulative verb or when an existential or presentative verb takes an argument that refers to an event, the copulative, existential, presentative or other verb elements that may act as such in the text will take the EXISTENCE value. In (2) and (3) we show in bold the events with existence value and underlined the subordinated events.

- (2) *La estatal brasileña también **está interesada** en estaciones de servicio y otros activos de Esso en el resto del Cono Sur Americano, dijo durante un encuentro con periodistas en Río.* [The Brazilian state-owned company is also interested in gas stations and other assets of Esso in the rest of South America, he/she said during a press conference in Rio.]

- (3) *Tal fue el caso de este lunes, en que se registraron durante 20 minutos fuertes nevadas en Colonia, según informó Canal 10. [That was the case of this Monday, when strong snowfalls during 20 minutes were recorded in Colonia, as reported by Channel 10.]*

It can also occur, as it is shown in (4), that nominalizations of the OCCURRENCE class may behave in a way similar to the predicates mentioned and introduce an event under the form of complement. In this case, they will also take the EXISTENCE value.

- (4) *Se descartó la ocurrencia de nevadas en Montevideo. Sí pueden producirse precipitaciones de "agua nieve". [The occurrence of snowfalls in Montevideo was ruled out. "Sleet" falls may certainly happen.]*

A partially similar change is proposed for French [3] with the introduction of the new class EVENT\_CONTAINER for events. Predicates that take a nominal event as subject (*De nombreuses manifestations se sont produites dans la tournée du dimanche.*) belong to this class.

The scheme for French also introduces for the *class* attribute the CAUSE value to account for verbs that indicate a causal relationship between two events (*causer, provoquer, engendrer*, etc.). A similar change had already been proposed in the SIBILA scheme: the extension of the INTENSIONAL ACTION class under the name of CAUSAL INTENSIONAL ACTION, in order to give place, precisely, to this kind of verbs.

In the TimeML annotation guide[17] the description of the events is presented by means of two differentiated elements: *event* and *makeinstance*<sup>1</sup>, the second of which is an empty element. This information was unified in SIBILA in order to simplify the annotation task, which implied the creation of 2 elements by each registered event. An alternative solution was then proposed, the *lexical item* attribute for the case of elided events that TimeML resolved by means of the creation of another instance with the same reference. The *lexical item* attribute is optional and is used to register an event in the cases of ellipsis, that is to say, to register the instance of an event the mention of which is omitted, because the predicate that names it may be recovered by resorting to another mention in the text. The remaining attributes of the elided event (empty event) collect additional information associated to it, as it is shown in (5) and (6).

- (5) *En el norte del país llovió abundantemente el sábado y <event lex\_item = "llovió"/> el domingo. [It rained heavily on Saturday and Monday in the north of the country.]*

- (6) *El corte de ruta comenzó el día 14 y <event lex\_item = "corte"/> terminó una semana después. [The roadblock began on the 14<sup>th</sup> and ended a week later.]*

---

<sup>1</sup> Reference to the *makeinstance* element has disappeared in the last versions of TimeML [21].

### The *Factivity* Attribute

The *factivity* attribute represents the degree of certainty of the utterer with regard to the occurrence of the referred event. It follows then that any affirmation about the occurrence or not of an event remains circumscribed to an enunciation context.

(7) *Esto **dificulta** aún más el **diálogo** con el gobierno uruguayo quien **confirmó** ayer a través de la cancillería que no se **negociará** mientras permanezca algún **corte**. [This makes the dialogue with the Uruguayan government even more difficult; the Uruguayan government confirmed through the Ministry of Foreign Affairs that they will not negotiate while a roadblock remains in place.]*

In (7) the events<sup>2</sup> are in bold and an aspectual operator (*permanezca* [*remains*]) is underlined. Note that while some events are presented as occurred (*confirmó* [*confirmed*], *dificulta* [*makes difficult*], *corte* [*roadblock*]) others are uncertain (*diálogo* [*dialogue*]) and the eventual negotiation (*negociará* [*will negotiate*]) is presented as future and with negative polarity. This means that the occurrence of some event referring word is not enough to infer that such event has occurred or is occurring. It is also necessary to interpret these terms in their contexts of occurrence, where they can be affected by elements of negative polarity, or by modal operators, or by predicates that affect their veracity value, and by combinations of all of them. The property of an event of having occurred or not or of being occurring is not then an evident piece of information. In fact, it is necessary to make some kind of textual inference in order to determine it.

Annotators must, precisely, make those inferences and annotate the event by attributing to it one of the following values:

YES – performed event

NO – non performed event

PROGRAMMED\_FUTURE – event with high probability of taking place

NEGATED\_FUTURE – highly improbable event

POSSIBLE – event that might take place

INDEFINITE<sup>3</sup> – event about which it is not known whether it has taken place or not

An example for each of these values is offered next:

(8) a. *La ministra Daisy Tourné **anunció** que algunos reclusos del Compen **serán trasladados** al interior del país, para **aliviar** la superpoblación de ese centro carcelario. No se **conocen** más novedades. [Minister Daisy Tourné announced*

<sup>2</sup> The expression of the event usually contains more than a word. A term considered to be the nucleus of the event is annotated (and is shown highlighted).

<sup>3</sup> Note that the difference between the values POSSIBLE and INDEFINITE is that the first one indicates that an event may take place in the future, while the second says that there are no elements to determine if an event took or did not take place: *Están valorando iniciar un paro* [POSSIBLE]. [They are evaluating to begin a strike] / *En ese momento valoraron iniciar un paro* [INDEFINITE] [They evaluated at that time to initiate a strike].

*that some Compen prisoners will be transferred to the provinces, in order to relieve that prison's overpopulation. No further news are known.]*

*anunció* = YES

*serán trasladados* = PROGRAMMED\_FUTURE

*aliviar* = POSSIBLE

*se conocen* = NO

b. *La idea de la exposición "Shoá. Memoria y legado del Holocausto" surgió de tres jóvenes judíos que querían transmitir el legado recibido de los supervivientes del exterminio. [The idea of the "Shoá. Memoria y legado del Holocausto" exhibition came from three Jewish young people who wanted to transmit the legacy received from the Holocaust survivors.]*

*transmitir* = INDEFINITE

c. *El gobierno uruguayo confirmó ayer a través de la cancillería que no se negociará mientras permanezca algún corte. [The Uruguayan government confirmed yesterday through the Ministry of Foreign Affairs that they will not negotiate while a roadblock remains in place.]*

*se negociará* = NEGATED\_FUTURE

Although factivity is closely related to tense, modality and polarity, the association is not automatic. Thus, events with the same values for these attributes may exhibit different factivity values: *Celebro que lleguen* [PROGRAMMED\_FUTURE] mañana [*I am glad they are coming tomorrow*] / *Dudo que lleguen* [POSSIBLE] mañana [*I doubt they are coming tomorrow*]; *Logró cerrar* [YES] la puerta [*He managed to close the door*] / *Olvídó cerrar* [NO] la puerta [*H forgot to close the door*].

In [19] there is a proposal to associate factivity to events, with a definition partially similar to ours. Besides, in this work it is developed a determinist algorithm for the calculus of factivity values, based on the fact that some relevant elements such as markers of polarity or modality, source introducing predicates and events selecting predicates, have been recognized and classified in a previous stage. But, to our knowledge, an attribute for factivity has not been previously included in an annotation scheme. We claim that this attribute will be useful for an effective recognition of this complex phenomenon.

### 3 The Annotated Corpora

#### 3.1 Description of the Corpora

The annotated corpus are constituted by journalistic and historical texts. Journalistic texts come from a corpus in Spanish created for the TempEval2<sup>4</sup> task, annotated on the basis of the TimeML scheme for Spanish. It was decided to annotate these texts in order to obtain a comparative parameter for Spanish.

---

<sup>4</sup> <http://www.timeml.org/tempeval2>

The corpus is formed by 11,986 tokens and 408 sentences. 1,677 events were annotated, most of them being verbs, nouns in second place, and lastly, a few of them being adjectives.

### 3.2 Agreement between Annotators

In order to evaluate the agreement between annotators we used the *agr*<sup>5</sup> measure proposed in [18], defined as follows:

Let  $A$  and  $B$  be the portions of text marked as events by two annotators  $a$  and  $b$  respectively. The *agr* measure tells us which proportion of  $A$  was also marked by  $b$ . To be precise, agreement between  $b$  and  $a$  is computed as:

$$\text{agr}(a \mid \mid b) = \frac{|\text{A agreeing with B}|}{|\text{A}|}$$

The  $\text{agr}(a \mid \mid b)$  measure corresponds to the *recall* if  $a$  is taken as *gold standard* and  $b$  as the labeling system, and to *precision* if  $b$  is the *gold standard* and  $a$  the system.

Agreement values between the annotators obtained are shown in table 2.

**Table 2.** Agreement between annotators

	Precision	Recall	F-measure
Global	91.6 %	93.0 %	92.3 %
Verbal Events	94.2 %	97.1 %	95.6 %
Nominal Events	85.8 %	88.7 %	87.3 %

We can see that the values are significantly lower for nouns than for verbs, as it was to be expected. Agreement values for other categories were not calculated for they are much less frequent in the corpus and, therefore, results would not be representative.

## 4 Machine Learning on the Corpus

### 4.1 Models for Learning

As a first experience of exploitation of the annotated corpus, we have developed a system that uses machine learning techniques for event recognition. The system only determines the text segments corresponding to events, a task that, for the particular case of nouns, is far from trivial. Recognition of segments referring to events was focused as a problem of sequential classification, using the usual system of labels B,I,O.

<sup>5</sup> The widely used Kappa measure was discarded as it suffers from various problems for sequential relatively scattered data [12].

We have used two learning methods radically different to generate classifiers: *Conditional Random Fields* (CRF) and an adaptation of *Support Vector Machine* (SVM) for problems of sequential classification.

CRF [9] is a discriminative model of sequential classification which, given a sequence  $x$  of observations, tries to obtain the sequence  $y$  of output labels that maximizes probability  $P(y|x)$ . This model has certain advantages over other models (of generative type, such as the *Hidden Markov Models*, HMM), for they do not need to calculate probability  $P(x)$  of the input sequence [9].

The SVM [22] model is not in principle a sequential classification method, although it can be adapted for that task. In the non-sequential case, SVM considers instances to be classified as points in a space with a certain dimension (possibly finite) and builds a lineal separator that partitions the space and divides the instances according to their class. In this way, the new instances will obtain their class according to the side of the hyperplane in which they are. Two modifications are necessary in order to apply this model to the sequential classification task. The first one is to be able to classify in more than two classes (SVM is a binary classification method), for which classifiers for each pair of classes are built, making then a pondered voting to determine the class to be assigned. The second one is to incorporate the rest of the elements of the sequence, in addition to the one that is being classified, to the classification. This is made by means of a technique called *forward parsing*, that uses labels assigned so far as attributes for subsequent classifications (proceeding from left to right in the sequence). For more details, consult [8].

70% of the total annotated corpus was used as training corpus in order to train classifiers. The remaining 30% was divided as follows: 15% as development corpus and 15% as testing corpus.

We used the CRFSuite<sup>6</sup> tool in order to train the classifier based on CRF and Yamcha<sup>7</sup> (a sequential classification tool) for the SVM classifier.

In both cases we used morphosyntactic attributes, some of which coming from the *Freeling* [1] tagger (token, lemma, POS-tag, number, mood and tense), and others associated with word structure (capital letters, last four letters). We considered a [-2,2] window centered on the token we wanted to classify.

## 4.2 Results

Results can be observed in table 3. The base line shown there was obtained by marking as an event every contiguous sequence of verbs and the nouns with the most frequent endings (4 final letters) among the nominal events of the training corpus. Results of agreement between annotators were used as top line.

---

<sup>6</sup> <http://www.chokkan.org/software/crfsuite>

<sup>7</sup> <http://chasen.org/~taku/software/yamcha>

**Table 3.** Classifiers' results (%) on the testing corpus

	Precision				Recall				F-measure							
	Base		Top	CRF	SVM	Base		Top	CRF	SVM	Base		Top	CRF	SVM	
	Base	Top	CRF	SVM	Base	Top	CRF	SVM	Base	Top	CRF	SVM	Base	Top	CRF	SVM
Global	67.1	91.6	81.7	<b>84.7</b>	57.3	93	72.4	<b>76.4</b>	61.8	92.3	76.7	<b>80.3</b>				
Verbal Events	65.2	94.2	83.2	<b>84.2</b>	79.3	97.1	91.9	<b>98.5</b>	71.6	95.6	87.3	<b>90.8</b>				
Nominal Events	63.3	85.8	71.8	<b>78.9</b>	27.9	82.7	41.2	<b>44.1</b>	38.8	87.3	52.3	<b>56.6</b>				

As it can be seen in the table, the base line of 61,8% of F-Measure is broadly surpassed by both methods. Contrary to what might be expected, given the fact that CRF is the state of the art in several problems of sequential classification, the SVM model gives higher values than the CRF model in all cases. On the other hand, both classifiers are far from reaching the top line, for which the F-Measure is 92,3%.

The most frequent mistakes made by both classifiers are related to nominal events. In order to improve this result, strategies similar to those used in [15] will be tried for the detection of non-deverbal event nouns. With regard to the precision value of verb events, we think that it is affected by the inclusion in this class of participle forms that many times do not constitute events.

### 4.3 Comparison with Other Works

With regard to automatic recognition, the obtained results are very encouraging, being of the same order that the results produced by similar works applied to English (see table 4). As it is shown by the table, only one system reaches a F-Measure higher than the ours. This work [10] includes among the input attributes information about thematic roles. For the time being, it is not possible to have this kind of information for Spanish for there does not exist an automatic tagger for thematic roles.

An important difference between the works mentioned and ours is the size of the corpus used for learning. In our case, the training corpus contains about 8,500 tokens, while the rest of the systems, all of them based on TimeBank, have a corpus 7 times larger. Even though it is generally accepted that it is necessary to have a larger corpus, differences between sizes of corpora used, on the one hand, and similarity of the results obtained, on the other hand, suggest that it is not the size of the corpus the element that has more bearing on the results.

**Table 4.** Comparison with other systems

System	F-Measure
Our system	76.7% (CRF) / 80.3% (SVM)
Evita [14]	80.1%
Sim-Evita [2]	73.0%
Boguraev, Ando [4]	80.3%
Step [2]	75.9%
March, Baldwin [9]	76.4%
Llorens et al [8]	81.4%

## 5 Conclusions

The SIBILA annotation scheme was defined; it constitutes an adaptation of the TimeML event annotation scheme to Spanish with the addition of elements for event factivity annotation. The basic part of the scheme is maintained, but some changes that we think make SIBILA more suitable for this language are introduced. From a comparative study with works for other Romance languages it comes out that similar modifications were proposed independently. Modifications proposed by SIBILA do not imply a mismatch with TimeML, a SIBILA conversion to TimeML is completely feasible, with some loss of information. This is important because TimeML is becoming a standard in works in this field.

The SIBILA scheme was validated by the effective annotation of a first set of texts with more than 1,500 events. Event manual annotation is not an easy task, there exist several difficult cases for which it is still necessary to clarify the criteria to be followed by annotators. Anyway, the agreement measure between annotators is very good (92.3% of global F-measure), even for nouns, that constitute the most complex case (87.3% of F measure in event nouns).

As a first experience of exploitation of the annotated corpus, a system that uses machine learning techniques for event recognition was developed. The system only determines the text segments corresponding to events, a task that, for the particular case of nouns, is far from trivial. Two learning methods radically different were used to generate classifiers: *Conditional Random Fields* (CRF) and an adaptation of *Support Vector Machine* (SVM) for problems of sequential classification. Results obtained are encouraging, having obtained in the best case 80% of F-measure with SVM. This number improves a lot (90%) if we only consider the verb events; the best F-measure that we have obtained for nominal events is 56.6%.

A larger volume of text is being annotated; it will be used for conducting new experiments, as well as for carrying independent factivity learning experiments. Another future work will be the integration with the enunciation axis, based on [16].

## References

1. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC) ELRA (2006)
2. Bethard, S., Martin, J.H.: Identification of event mentions and their semantic class. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP 2006, pp. 146–154. Association for Computational Linguistics, Stroudsburg (2006)
3. Bittar, A., Amsili, P., Denis, P., Danlos, L.: French TimeBank: An ISO-TimeML Annotated Reference Corpus. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: shortpapers, Portland, Oregon, pp. 130–134 (2011)
4. Boguraev, B., Kubota Ando, R.: TimeML-compliant text analysis for temporal reasoning. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 997–1003. Morgan Kaufmann Publishers Inc., San Francisco (2005)
5. Boguraev, B., Kubota Ando, R.: Effective Use of TimeBank for TimeML Analysis. In: Schilder, F., Katz, G., Pustejovsky, J. (eds.) Annotating, Extracting and Reasoning about Time and Events. LNCS (LNAI), vol. 4795, pp. 41–58. Springer, Heidelberg (2007)
6. Carletta, J.: Assessing agreement on classification tasks: The Kappa statistic. Computational Linguistics 22, 249–254 (1996)
7. Caselli, T., Bartalesi, V., Sprugnoli, R., Pianta, E., Prodanof, I.: Annotating Events, Temporal Expressions and Relations in Italian: the It-TimeML Experience for the Ita-TimeBank. In: Proceedings of the Fifth Law Workshop (LAW V), Portland, Oregon, pp. 143–151 (2011)
8. Kudo, T., Matsumoto, Y.: Chunking with Support Vector Machines. In: NAACL (2001)
9. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
10. Llorens, H., Saquete, E., Navarro-Colorado, B.: TimeML events recognition and classification: learning CRF models with semantic roles. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 725–733. Association for Computational Linguistics, Stroudsburg (2010)
11. March, O., Baldwin, T.: Automatic event reference identification. In: Proceedings of the Australasian Language Technology Association Workshop 2008, páginas, Hobart, Australi, pp. 79–87 (2008)
12. Ben, M.: Investigating Classification for Natural Language Processing Tasks, Ph.D. Thesis, Cambridge University (2007)
13. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: Fifth International Workshop on Computational Semantics, IWCS-5 (2003)

14. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TIMEBANK Corpus. In: Proceedings of Corpus Linguistics, pp. 647–656 (2003)
15. Resnik, G., Bel, N.: Automatic Detection of Non-deverbal Event Nouns in Spanish. In: Proceedings of the 5th International Conference on Generative Approaches to the Lexicon, Istituto di Linguistica Computazionale, Pisa (2009)
16. Rosá, A., Wonsever, D., Minel, J.-L.: Comparación de dos métodos para la extracción de opiniones en textos en español. In: Proceedings of IBERAMIA 2010, Workshop on Natural Language Processing and Web-based Technologies, Bahía Blanca (2010)
17. Saurí, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: a robust event recognizer for QA systems. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 700–707. Association for Computational Linguistics, Stroudsburg (2005)
18. Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML Annotation Guidelines Version 1.2.1 (2006)
19. Saurí, R.: A Factuality Profiler for Eventualities in Text. PhD Dissertation. Brandeis University (2008)
20. Saurí, R., Batiukova, O., Pustejovsky, J.: Annotating Events in Spanish TimeML Annotation Guidelines. Version TempEval-2010 (2009)
21. Saurí, R., Goldberg, L., Verhagen, M., Pustejovsky, J.: Annotating Events in English TimeML Annotation Guidelines. Version TempEval-2010 (2009)
22. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
23. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. In: Language Resources and Evaluation (2005)
24. Wonsever, D., Malcuori, M., Rosá, A.: Sibila: esquema de anotación de eventos. Technical Report 08–11, Biblioteca InCo PEDECIBA (2008) ISSN: 0797–6410
25. Wonsever, D., Malcuori, M., Rosá, A.: Factividad de los eventos referidos en textos. Technical Report 09–12, Biblioteca InCo PEDECIBA (2009) ISSN: 0797–6410

# Automatically Generated Noun Lexicons for Event Extraction

Béatrice Arnulphy, Xavier Tannier, and Anne Vilnat

LIMSI-CNRS, Univ. Paris-Sud  
91403 Orsay, France  
`firstname.surname@limsi.fr`

**Abstract.** In this paper, we propose a method for creating automatically weighted lexicons of event names. Almost all names of events are ambiguous in context (*i.e.*, they can be interpreted in an eventive or non-eventive reading). Therefore, weights representing the relative “eventiveness” of a noun can help for disambiguating event detection in texts.

We applied our method on both French and English corpora. Our method has been applied to both French and English corpora. We performed an evaluation based upon a machine-learning approach that shows that using weighted lexicons can be a good way to improve event extraction. We also propose a study concerning the necessary size of corpus to be used for creating a valuable lexicon.

## 1 Introduction

Information extraction consists in a surface analysis of text dedicated to a specific application. Within this general purpose, detection of event descriptions is often an important clue (*e.g.*, temporal ordering of events on a chronological axis). However, events are, in open-domain information extraction, less studied than general named entities like location and person names.

We focused our study on nominal forms of events<sup>1</sup>. Lexicons provide lists of nouns that can be considered as events in context. These lexicons only contain common nouns, but the events are not only named with common nouns or with words that are in the existing lexicons. Indeed, almost all nouns are highly dependent on context to assign those nouns an event property. In this paper, we propose a method using patterns and shallow parsing to automatically build a lexicon for nouns event extraction. We apply this method on two languages (French and English). Our work is close to Bel et. al [5], which present cues for the disambiguation of non-deverbal event nouns. Contrary to Bel et al. [5], our lexicon provides quantitative information concerning the “eventiveness” of the words. Such a lexicon would help disambiguation of noun class in context.

First, we present our observations about the way we name events and we propose a brief survey of works dealing with nominal forms of events. Then we

<sup>1</sup> This work has been partially funded by OSEO under the Quaero program, as well as French National Research Agency (ANR) under project Chronolines (ANR-10-CORD-010).

present the resources we used in our study, before introducing our method for the automatic creation of the weighted lexicons in order to extract names of events. To conclude, we evaluate the performances of our weighted lexicons in comparison with other classical lexicons, based on annotated corpora.

## 2 The Event

From our point of view, an event is what happens, it corresponds to a change of state. It can be either recurring or unique, predicted or not. It may last a moment or be instantaneous. It can occur in the past, the present or the future.

### 2.1 Construction of Event Names

In the Humanities, studies about events usually deal with single events or only few events (e.g., *Jasmin Revolution* or *H1N1 flu* [8]), and do not offer generalization hints. We do not consider events in the same way. According to those studies and based upon our corpus analysis, we propose a description of the lexical construction of names of events, organized into three types, according to their construction.

1. **Nominalization related to an action verb.** A lot of event names are formed from words morphologically related to action verbs. They can be supported by deverbal nouns, nouns derived from action or event verbs by a process of nominalization. For example:

- the verb *fêter* (“to celebrate”) is morphologically linked to *fête* (“party”, “celebration”): *la fête de la musique* (“the music festival”).
- the verb *to assign* is nominalized into *assignment*.

In all languages, this nominalization is often ambiguous (nominalization could refer, either to the process or to the result of the process, the location or the object). Here, *assignment* can be the act of assigning something, as well as the result of this action.

2. **Nominalization not related to verbs.** Some names of events are introduced by nouns that intrinsically denote events, such as *festival* or *match*. Then a disambiguation is needed: in French, *salon* can be either a lounge or an exhibition show (e.g., *salon de l'automobile* — “motor exhibition”).
3. **Metonymic nominalization.** Other nouns or noun phrases become name of events in specific contexts, often by metonymy: a location name (*Tchernobyl* refers to the 1986 nuclear accident that occurred in this town [18]) or a date (*September-11* stands for the 2001 attacks [7]).

For each of those three classes, we could use resources is a first approach that must be refined in context; context must be used to decide whether nouns or noun phrases are events.

## 2.2 Event Nouns in NLP

In NLP, the definition of events seems to be quite *ad hoc* to the application they are meant to describe. We will focus here on works dealing with events nouns in temporal extraction project and those more specifically oriented towards nominal event extraction.

**Events in Temporal Extraction.** *TimeML* [21] is a specification language for events and temporal expressions, originally developed to improve the performance of question answering systems. In TimeML, an event is defined as “a cover term for situations that happen or occur”. An event is based upon punctuality or duration properties and it can describe states. The TimeML specification language is used for the annotation of numerous corpora in several languages.

In our work, we consider all kinds of events, being proper names or not, taking place in the past, the present or the future. We do not consider states (even if they can be nominalized) and we focus on events based upon a nominalization, not on verbs or predicative clauses, which are the main interest of TimeML.

We must also pay attention to the few Named Entity Recognition campaigns which considered events in their frameworks. Automatic Content Extraction (**ACE**) [11] proposed an event extraction project [1] in which the classification of events is detailed (arguments are related to particular events) and precise, but it only concerns a very limited number of domains (the category “life” is composed of “be-born”, “be-injured” sub-domains, etc.). The objective of ACE is to detect thematic events. We are interested in all mentions of nouns describing events without any thematical predefined class. In the continuation of MUC [15] and ACE, SemEval<sup>2</sup> paid interest to events within the framework of a semantic role labelling approach and detection of eventive verbs in Chinese news. French **ESTER** campaigns [14] provide a very different classification of events as named entities: the aim is to produce an open-domain named entity tagging. For this purpose, event typology is quite simple: *historical and unique* events on the one hand, *repetitive* events on the other hand. Even if this typology is not detailed, it corresponds to our point of view on events.<sup>3</sup>

**Nominal Event Extraction.** Little research has been fully dedicated to automatic extraction of nominal events. We described here some works that follow a comparable approach to ours, using lexicons and linguistic classed-based information. Evita [25] is an application recognizing verbal and nominal events in English texts. This work is based upon the TimeML definition. Disambiguation of nouns that have both eventive and non-eventive interpretations is based on a statistical module, using a lexical lookup in WordNet<sup>4</sup> and the use of a Bayesian Classifier trained on SemCor. Also for English, following the ACE definition of events, Creswell et al. [10] created a classifier that labels NPs as events or

<sup>2</sup> <http://semeval2.fbk.eu/semeval2.php>

<sup>3</sup> In our works, we developed a more detailed typology which takes into account modality (factual, abstract, etc.), frequency (unique, recurring, instantiation), and temporality of the event.

<sup>4</sup> <http://wordnet.princeton.edu/>

non-events. They worked on seed term lexicons from WordNet and the British National Corpus.<sup>5</sup> Eberle et al. [12] present a tool using cues for the disambiguation of readings of German *ung*-nominalizations within their sentential context. Russo et al. [24] focused on the eventive reading of deverbals in Italian, using syntagmatic and collocational cues. In a close approach, Resnik and Bel [23] worked on Spanish and Bel et al. [5] on Spanish and English. They tried to disambiguate result and event, as well as deverbal nouns and non deverbal nouns. In a machine-learning approach, they used cues which are assumed in the linguistic literature (aspectual verbs and prepositions, temporal quantifying expressions, etc.). Dealing with the classification of deverbals (result, event, underspecified or lexicalized nouns), Peris et al. [19] focused on Spanish. Several lexicons, as well as automatically or manually extracted features, are evaluated in a machine learning model.

### 3 Resources

In our study, we use several resources : corpora and existing lexicons. We worked with raw corpora for the lexicon extraction, manually annotated corpora for the evaluation, both type of corpus in French and English. Here is an overview of these resources for English and French.

#### 3.1 Corpora

**For the Lexicon Extraction.** For the creation of our weighted lexicons in French and English, we used a corpus of newswires from the French News Agency *AFP*<sup>6</sup>. The *AFP* corpus is available on a same period in two languages, so we could have similar corpus. The English corpus is composed of 1.3 millions texts over the 2004-2011 period (120 million tokens). The French corpus is of 1 million texts over 2005-2011. In French, we also used a corpus of 120,246 newspaper articles from *Le Monde* (two years, other 2001-2002, 61 million tokens): this corpus is of similar size to the French *AFP* corpus ; these two corpora are also similar according to the realities they deal with, even if they are evoked differently (newspaper articles and short news). We thus created a weighted lexicon from this corpus in order to complete our French weighted lexicon.

#### For the Evaluation

*The two TimeML annotated corpora* we used are based on newswires (cf. 2.2). In English, *TimeBank 1.2* [20] contains 1,722 non-stative nominal events. The annotated texts are extracted from news media (*Wall Street Journal*, *ABC*, *CNN*, *Voice Of America*) over the 1989-1998 period. In French, *FR-TimeBank* [6] contains 663 nominal mentions of event. The annotated texts come from the newspaper *L'Est Républicain* over the period 1999-2003.

<sup>5</sup> <http://www.natcorp.ox.ac.uk/>

<sup>6</sup> We thank the French News Agency (AFP) for providing us with the corpus.

*Our French Manually Annotated Corpus* is composed of 192 French newspaper articles from *Le Monde* and *L'Est Républicain* for a total amount of 48 thousand words. Our corpus contains 1,844 events, which is comparable to TimeBank 1.2, FR-TimeBank, as well as the Italian IT-TimeBank [24] (3,695 event nouns) and the English corpus from [10] (1,579). We defined and followed precise annotation guidelines: they detail a typology of events, as well as instructions for deciding whether a noun or a noun phrase is an event or not. Among these instructions:

- Try to imagine some non-ambiguous valuable substitutes for the noun. This proves to be very effective.
- Take inspiration from examples of eventive and not eventive uses of the same word, that can be found in dictionaries, together with their proper definition.
- Remember that enumeration items are often (not always) of the same class.
- When decision is impossible, choose to annotate as non-event.

Delimiting the event boundaries is also a difficult issue and the guidelines provide instructions for this other problem. Following the guidelines, the two annotators (the authors of the guidelines) obtained a good agreement for the annotation of the heads of noun phrases ( $\kappa=0.808$ ). Among the corpus, the 109 documents from *L'Est Républicain* are common with FR-TimeBank [6]. The two annotations have a different purpose, but seem quite similar according to the good inter-annotator agreement ( $\kappa=0.704$ ).

### 3.2 Lexicons

In French, two lexicons can be useful to find nominal mentions of events: *VerbAction* [26] and Bittar's alternative lexicon [6]. In English, we used nouns of events and actions from *WordNet* [13].

*VerbAction* is a deverbal noun lexicon. It contains a list of French verbs of action (e.g., *fêter* — “to celebrate”) together with the deverbal nouns derived from these verbs (*la fête* — “the feast/celebration”). However, deverbals' eventive reading can be ambiguous, mainly because they can also refer to the result of the action. The *VerbAction* lexicon contains 9,393 noun-verb lemma pairs and 9,200 unique nominal lemmas. It was built by manually validating a list of candidate couples automatically composed from lexicographical resources and from the Web.

*The Alternative Noun Lexicon of Bittar* contains 804 complementary event nouns.<sup>8</sup> These nouns are not deverbals (e.g., *anniversaire* — “birthday” and *grève* — “strike”). They have at least only one eventive reading, and can be ambiguous, as for deverbals: they may denote the event or the object of the process, as it is the case for *apéro* (“apéritif/cocktail”) and *feu* (“fire”). Some of

<sup>7</sup> We used the Carletta's Kappa coefficient [9]. This measure compares the agreement against what might be expected by chance. According to Landis and Koch [17], from 0.6 to 0.8 is what we consider a good agreement. Up to 0.8 is a very good agreement.

<sup>8</sup> We are thankful to André Bittar for providing us this list.

these nouns describe a state and do not match our definition of events (*e.g.*, *absence* — “non-attendance”). Lots of these nouns (like *anticoagulothérapie* — “anticoagulation therapy”) belong to language of speciality, such as the medical one. This lexicon has been used for TimeML manual annotation in French.

*The Action and Eventive Nouns in WordNet* contains 5903 nouns tagged as “act” (for action) or events . This list of English words can be considered as comparable with the French lexicons (VerbAction and Bittar). It contains words describing events in almost all cases (*war, election, show, carnival*), expressions which are very ambiguous (*arts and crafts, bet, coloration*), multi-word expressions (*a cappella singing*), name of events (*Arab-Israeli War, Battle of Britain, laser trabecular surgery*), but also expressions that do not seem to fit with any event definition (*Attorney General, judo, industry*).

## 4 Automatic Lexicon Creation

We showed in a previous work [3] that a lexicon of event nominals can be created by applying extraction rules. These experiments demonstrated that the French automatically generated lexicon (created from *Le Monde*) is as precise as manually-validated lists, and weights can be used to improve the classification of nouns. This work was only conducted on French. In the present study, we extend these experiments to English and evaluate the process. We also generated a new lexicon for French from the *AFP* corpus in order to obtain comparable multilingual results. From the corpora of *AFP* news, we extracted two lexicons of nouns describing events according to our extraction rules, the first one in French and the second one in English. Our extraction rules depend on the use of a syntactic parser. For this purpose, we chose a robust parser, XIP.

XIP [2] is a robust parser for French and English which provides dependency relations and “classical” named entities (like persons or locations). But events are not identified. XIP is a product from XRCE (Xerox Research Centre Europe), distributed with encrypted grammars that cannot be changed by the users. However, it is possible to add resources and grammar rules in order to enrich the representation. It is what we have done. The parser is language-dependent, but the extraction rules are commutable to other languages with a minimal cost. We developed the same type of rules for French and English. We performed a corpus analysis to evaluate the meaningful of those rules for event extraction.

### 4.1 Extraction Rules

**Temporal Rules.** Because events are anchored to time, they are often linked to temporal prepositions and used in temporal context. Using these temporal markers is a good way to extract event noun phrases. In this way, we focused on the more unambiguous prepositions. These prepositions or trigger-words show:

	(FR)	(EN)
the occurrence of an event:	<i>à l'occasion de</i>	<i>at the time/moment of</i>
	<i>au moment de</i>	<i>on the occasion of</i>
a referential use of the event:	<i>avant/après</i>	<i>the morning of</i>
	<i>le lendemain de</i>	<i>the day before</i>
	<i>au matin de</i>	<i>at the morning of</i>
	<i>à la suite de</i>	<i>following (temporal)</i>
	<i>lors de</i>	<i>during</i>
an internal moment of the event:	<i>à l'issue de</i>	<i>the beginning of</i>

However, few of these triggers are unambiguously temporal triggers. Some like *avant* (“before”), *après* (“after”), *au commencement de* (“at the beginning”) can be either temporal or locative, while *à l'occasion de* (“when”) or *la veille* (“the day before”) have only a temporal interpretation.

**Verbal Rules.** A previous study on French [4] shows which verbs are the most meaningful for event extraction and in which configuration (subject and/or object) it would be greatful to use them. We took this information into account in the following rules:

	(FR)	(EN)
in a subject position:	<i>avoir lieu, se tenir</i>	<i>to take place, to come about</i>
in an argument position:	<i>entraîner</i>	<i>to be the result of</i>

We focus on three types of verbs. The first type concerns verbs which explicitly introduce events (occurrence predicates):

(FR) *se produire, avoir lieu*

(EN) *to befall, to occur*

Le **sommet du G8** est organisé à Deauville.

The **G8 Summit** is organized in Deauville.

The second type of verbs introduce a relation of cause and/or effect for events. Indeed as we can see in the following examples, a causal action or event provokes another event.

(FR) *occasionner*

(EN) *to ensure*

La **crise économique** entraînera la **famine** dans les pays sous-développés.

The **economic crisis** will lead to **famine** in underdeveloped countries.

Le **feu** provoqué par l'**attaque-suicide**, n'était pas encore éteint que [...]

The **fire** provoked by the **suicide attack**, was not extinguish yet that [...]

And the last one is for verbs which present a moment of an event (aspectual predicates):

(EN) *to begin, to last*

‘**The Event**’ will *end* like all successfull US TV shows.

Let the **spectacle** *begin*.

We used verbs which are quite always meaningful for event extraction, according to the observation from a corpus analysis. The verbs we selected introduce events in more than 90% of the cases.

## 4.2 Calculating the Eventiveness Relative Weight

The extraction rules based on contextual clues gives precise results ( $P > 0.80$ ) but a low recall ( $R < 0.10$ ). Therefore, to be representative, the lexicon has to be extracted from a large corpus (see Section 5.3). The application of the extraction rules allows the extraction of a list of eventive nouns. From this list and our corpus, we can fetch information about the level of ambiguity (eventive or non-eventive reading) of each word in the corpus. Otherwise, we are able to predict how eventive the word is expected to be. This prediction is achieved by computing the Eventiveness Relative Weight (*ERW*): after applying the rules on the corpus, we calculate a weight for each noun extracted as an event at least twice.  $ERW(w)$  is the number of occurrences  $e(w)$  of the word  $w$  tagged by the rules, divided by the total number of its occurrences  $t(w)$ :

$$ERW(w) = \frac{e(w)}{t(w)} \quad (1)$$

As the recall of the rules is low, the *ERW* is obviously not a rate or a probability of the eventive reading of this word. However, a relative comparison with other weights allows us to estimate how ambiguous the noun is in a given corpus. This value is then interesting for noun classification.

**Table 1.** Examples of trigger words extracted by the extraction rules

Potential triggers <i>French</i>	Nb. detected / total occ	<b>ERW</b>	Potential triggers <i>English</i>	Nb. detected / total occ	<b>ERW</b>	
chute	fall	434 / 2620	<b>0.166</b>	overthrow	383 / 448	<b>0.855</b>
clôture	closing	63 / 470	<b>0.134</b>	intifada	7 / 11	<b>0.636</b>
élection	election	1243 / 9713	<b>0.128</b>	bombardement	6 / 12	<b>0.500</b>
bousculade	jostle	12 / 115	<b>0.104</b>	testimony	426 / 13109	<b>0.032</b>
crise	crisis	286 / 6185	<b>0.046</b>	sleepover	3 / 27	<b>0.111</b>
tension	tension	16 / 1595	<b>0.001</b>	publication	154 / 9337	<b>0.016</b>
subvention	subvention	2 / 867	<b>0.002</b>	marathon	52 / 8070	<b>0.006</b>
Anschluss	Anschluss	3 / 4	<b>0.750</b>	play-off	73 / 75	<b>0.973</b>
méchoui	mechoui	3 / 5	<b>0.600</b>	breastfeeding	3 / 4	<b>0.750</b>
krach	krach	20 / 169	<b>0.118</b>	overheat	3 / 7	<b>0.428</b>
RTT	~ day off	14 / 166	<b>0.084</b>	stopover	372 / 1345	<b>0.276</b>
demi-finale	semifinal	35 / 553	<b>0.063</b>	cross-examination	53 / 416	<b>0.127</b>
cessez-le-feu	cease-fire	15 / 440	<b>0.034</b>	distillery	4 / 126	<b>0.032</b>
accès	access	9 / 2828	<b>0.003</b>	welcome	66 / 3884	<b>0.017</b>
11 septembre	September-11	12 / 4354	<b>0.003</b>	influenza	37 / 6019	<b>0.006</b>

This interest is illustrated by examples given in Table 1: the upper part of the tables presents words which are found in the English or French standard lexicons while the lower part presents words fetched by the extraction rules which are not in the standard lexicons. We created three weighted lexicons: one based on the two years *Le Monde* French corpus, and two from the whole *AFP* corpora (one in English and one in French). The lemmas present in the weighted lexicons must be extracted by our rules at least twice. See Table 2.

**Table 2.** From corpora to weighted lexicons: Size in number of tokens

Corpus used for the lexicon creation	Number of tokens		Number of lemmas in the weighted lexicon	
	total size	extracted differents		
(FR) AFP (2005-2011)	166,077	8,053		3,538
(EN) AFP (2004-2011)	543,394	14,619		3,452
(FR) LM (2001-2002)	61,920,573	19,767	4,843	1,559

## 5 A Machine-Learning Evaluation

We applied the French and English automatically-built weighted lexicons using a machine-learning approach and conducted an evaluation. We added the *ERW* value as a feature in the rule-based classifier J48, an implementation of C4.5 algorithm [22], as implemented in the software Weka [16]. The manually annotated corpus was split into a training set (75% of the annotated corpus) and a test set (the remaining 25% of the annotated corpus). The training set contains the same number of event entries than non event entries (see Table 3).

**Table 3.** Number of tokens in the training and test corpus

	Training Set			Test Set		
	total	YES	NO	total	YES	NO
English	2,182	1,092	1,092	3,246	453	2,793
French	5,226	1,263	1,263	2,700	566	2,134

For each language, we implemented three very basic models, allowing us to show the trade-off introduced by the *ERW*, without any suspicion of side effect due to other features:

- $M_l$  uses only the standard manually validated lexicons:  
(FR) VerbAction and Bittar (EN) WordNet action and event nouns
- $M_r$  uses only the *ERW*, as a real value. As we have two weighted lexicons in French, they are called:
  - $M_r^{LM}$ , based on an extraction of the lexicon from two years of *Le Monde* corpus.
  - $M_r^{AFP}$ , our new weighted lexicon based on the *AFP* corpus.
- $M_{rl}$  uses both existing and weighted lexicons.

Our models are evaluated using the classical measures of precision (P), recall (R) and F-measure (F1)<sup>9</sup>

<sup>9</sup> Precision is defined as the observed probability for a hypothesized element to be correct, recall is the observed probability for a referenced element to have been found and F-measure is the weighted harmonic mean of precision and recall.

### 5.1 ERW Lexicons vs. Standard Lexicons Comparison

Table 4 presents the evaluation of the French *LM* and *AFP* weighted lexicons in comparison to standard lexicons (Bittar’s and VerbAction lexicons) on our annotated corpus. Table 4 presents the evaluation of the English *AFP* weighted lexicon in comparison to the standard lexicon extracted from WordNet on the TimeBank 1.2 corpus.

**Table 4.** Evaluation of the weighted lexicon in French (left) and in English (right)

	<i>Our EWR lexicons</i>		<i>Standard</i>	<i>Mixed</i>			<i>Our EWR lexicons</i>		<i>Standard</i>	<i>Mixed</i>
	$M_r^{LM}$	$M_r^{AFP}$	$M_l$	$M_{lr}^{LM}$	$M_{lr}^{AFP}$		$M_r^{AFP}$	$M_l$	$M_{lr}^{AFP}$	
P	0.49	0.55	0.53	0.54	<b>0.60</b>	P	0.36	0.30	<b>0.36</b>	
R	0.89	0.77	0.88	0.89	0.84	R	0.71	0.64	0.77	
F1	<b>0.63</b>	<b>0.64</b>	<b>0.66</b>	<b>0.67</b>	<b>0.70</b>	F1	0.476	0.414	<b>0.493</b>	

First of all, in both French and English, we notice that:

- Using only our weighted lexicons ( $M_r$ ) leads to similar results than using standard manually validated lexicons ( $M_l$ ).
- Combining all information leads to a small but substantial improvement of precision and recall.

From these observations, we confirm that our automatically created weighted lexicons are as precise as the standard manually validated lexicons in French and in English. In French, we also notice that the weighted lexicon from *AFP* corpus is more precise than both the standard lexicon ( $P=0.53$ ) and that the *LM* one ( $P=0.49$ ). Besides, as a point of comparison, we applied the  $M_r^{AFP}$  model on the FR-TimeBank and our annotated corpus. The performances of the *AFP* weighted lexicon are similar on the two annotated corpora, even if the corpora were not annotated with the same aim or guidelines. Precision reaches 0.56 on FR-TimeBank and 0.55 on our annotations, recall is of 0.77 on both corpora and F1 is 0.648 and 0.642. Moreover, we observe that results for English are much lower than results for French. However, this difference is not due to the lexicons quality. Indeed, the trade-off between standard lexicons (VerbAction + Bittar in French, WordNet in English) and our ratio lexicon is similar. This means that their quality are similar as well. Our initial guess that a direct translation of French rules was enough is then confirmed. The fact that lexicons perform so poorly in English rather tends to prove that the problem is just more difficult in English. Studying this difference is one of our prospectives.

### 5.2 ML-Evaluation vs. Threshold Based Approach Comparison

As a comparison to the ML-Evaluation and in order to observe the evolution of performances, we tested different “slices” of the lexicon in a threshold based

**Table 5.** Results when applying “slices” of *ERW* on the corpus (French LM lexicon)

Words of <i>ERW</i> >	Precision	Recall	F-measure
10%	84.1%	16.6%	0.28
8%	83.6%	24.3%	0.38
6%	79.8%	31.5%	0.45
1%	56.3%	71.0%	0.63
0.5%	43.4%	80.1%	0.56

approach. According to the value of the *ERW*: all words with an *ERW* higher than 10%, then all those with an *ERW* greater than 8%, 6%, etc. The results are presented in the Table 5. Precision and recall evolve in an opposite way: when the lexicon is less selective, the recall increases and the precision decreases. The best F-measure (for 1% *ERW*) is 0.63, a value similar to the F-measure of the VerbAction and Bittar’s lexicons combined (0.61).

### 5.3 Impact of the Size of the Corpus

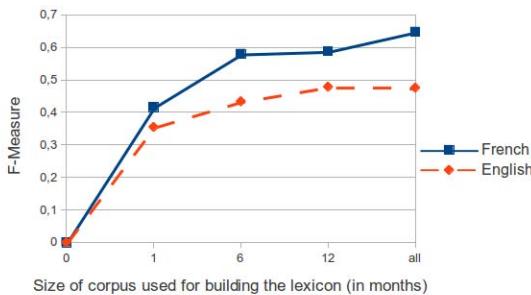
As the precision of our extraction rules is good and the recall is low, we stated that a large corpus was necessary. But how large must the corpus used for the lexicon extraction corpus be? We created several weighted lexicons from parts of our corpus, from one month to one year of news. We studied the performances of  $M_r^{Afp}$  models depending of the size of the corpus it was based on (cf. Table 6).

**Table 6.** Evaluation of the weighted lexicons depending of the size of the corpus

<i>Lexicon created on</i>	1 month 07 2005	6 months 07-12 2005	1 year 2005	all 2004-2011
<i>French</i>	P	0.665	0.539	0.512
	R	0.303	0.628	0.692
	F1	<b>0.416</b>	<b>0.58</b>	<b>0.588</b>
<i>English</i>	P	0.36	0.31	0.35
	R	0.35	0.7	0.76
	F1	<b>0.36</b>	<b>0.43</b>	<b>0.48</b>

Figure 1 shows that, in English and in French, the gain in terms of F-measure of a model trained on a one-year-learned lexicon is as good as for a whole-corpus-learned lexicon. The figures and the shape of the curves seem to show that more corpora would not increase significantly the performances.

However, even if global performances are not improved by adding more and more documents, it is still interesting to extract names of event in a much longer period or during a specific period of time. Indeed, events and their names are anchored to time, and very particular event names will be used only at a precise moment (*e.g. tsunami, Arab Spring*).



**Fig. 1.** Progression of the F-measure depending of the size of the corpus

## 6 Conclusion

We automatically created lexicons of eventive nouns in French and English by using rules based on verbs and temporal clues. A relative weight of eventiveness (*ERW*) is added to the lexicon. The *ERW* a great information in order to help for the disambiguation of the words. In a machine-learning evaluation, we showed that our automatically generated weighted lexicons are competitive to the lexicons which were manually created. These experiments also prove that the transposition of the rules from a language to another one is possible. As well, we observed that a one-year corpus is significant enough to build a lexicon with our method and to obtain comparable result as those of classical lexicons.

According to our experiments on French, we conclude that the performance of the weighted lexicon is dependent on the corpus chosen to generate the lexicon. It would be interesting to apply our method on other domains. In English, as the result with the lexicon from WordNet is low, we plan to study this difference. However, because some words take an eventive meaning at a given moment (*e.g.*, *le nuage islandais* (literally “Icelandic cloud”) refers to the blast of the Eyjafjöll volcano from March to October 2010), we would like to work on a new lexicon which would consider the date of the appearance of an event name.

## References

1. ACE (Automatic Content Extraction) - English Annotation Guidelines for Events, V 5.4.3 2005.07.01. Tech. rep., LDC (2005)
2. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond Shallowness: Incremental Deep Parsing. *Natural Language Engineering* 8 (2002)
3. Arnulphy, B.: A weighted Lexicon of French Names. In: Proc. of RANLP Student Workshop (2011)
4. Arnulphy, B., Tannier, X., Vilnat, A.: Les entités nommées événement et les verbes de cause-conséquence. In: *Actes de TALN* (2010)
5. Bel, N., Coll, M., Resnik, G.: Automatic Detection of Non-deverbal Event Nouns for Quick Lexicon Production. In: Proc. of COLING (2010)

6. Bittar, A.: Building a TimeBank for French: A Reference Corpus Annotated According to the ISO-TimeML Standard. Ph.D. thesis, Univ. Paris Diderot (2010)
7. Calabrese Steinberg, L.: Les héméronymes. Ces événements qui font date, ces dates qui deviennent événements. *Mots. Les langages du politique* 3 (2008)
8. Calabrese Steinberg, L.: La nomination d'événements dans le discours d'information : entre activité collective et déférence épistémiologique. In: *Colloque Langage, discours, événements* (2011)
9. Carletta, J.: Assessing Agreement on Classification Tasks: the Kappa Statistic. *Computational Linguistics* 22 (1996)
10. Creswell, C., Beal, M.J., Chen, J., Cornell, T.L., Nilsson, L., Srihari, R.K.: Automatically Extracting Nominal Mentions of Events with a Bootstrapped Probabilistic Classifier. In: *Proc. of the COLING/ACL* (2006)
11. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R.: The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation. In: *Proc. of LREC* (2004)
12. Eberle, K., Faaf, G., Heid, U.: Corpus-based identification and disambiguation of reading indicators for German nominalizations. In: *Proc. of Corpus Linguistics* (2009)
13. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. Bradford Books (1998)
14. Gravier, G., Bonastre, J.F., Geoffrois, E., Galliano, S., McTait, K., Choukri, K.: Ester, une campagne d'évaluation des systèmes d'indexation automatique d'émissions radiophoniques en français. In: *Proc. of JEP* (2004)
15. Grishman, R., Sundheim, B.: Message Understanding Conference: A Brief History. In: *Proc. of COLING* (1996)
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009)
17. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33 (1977)
18. Lecolle, M.: Éléments pour la caractérisation des toponymes en emploi événementiel. In: Evrard, I., Pierrard, M., Rosier, L., Raemdonck, D.V. (eds.) *Les sens en marge - Représentations linguistiques et observables discursifs: actes du colloque international de Bruxelles, Novembre 3-5, L'Harmattan* (2009)
19. Peris, A., Taulé, M., Boleda, G., Rodriguez, H.: ADN-classifier: Automatically assigning denotation types to nominalizations. In: *Proc. of LREC* (2010)
20. Pustejovsky, J., Verhagen, M., Saurí, R., Littman, J., Gaizauskas, R., Katz, G., Mani, I., Knippen, R., Setzer, A.: TimeBank 1.2. LDC (2006)
21. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: *IWCS-5, Fifth International Workshop on Computational Semantics* (2003)
22. Quinlan, R.: *C4.5: Programs for Machine Learning*. Morgan Kaufman Publishers (1993)
23. Resnik, G., Bel, N.: Automatic detection of non-deverbal event nouns in spanish. In: *di Linguistica Computazionale*, I. (ed.) *Proc. of the 5th Int. Conference on Generative Approaches to the Lexicon* (2009)
24. Russo, I., Caselli, T., Rubino, F.: Recognizing deverbal events in context. In: *Proc. of CICLing*. Springer, Heidelberg (2011)
25. Saurí, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: A Robust Event Recognizer for QA Systems. In: *Proc. of HLT/EMNLP* (2005)
26. Tanguy, L., Hathout, N.: Webaffix: un outil d'acquisition morphologique dérivationnelle à partir du Web. In: Pierrel, J.M. (ed.) *Actes de TALN. ATILF* (2002)

# Lexical Acquisition for Clinical Text Mining Using Distributional Similarity

John Carroll<sup>1</sup>, Rob Koeling<sup>1</sup>, and Shivani Puri<sup>2</sup>

<sup>1</sup> Department of Informatics, University of Sussex, Brighton BN1 9QH, UK

<sup>2</sup> GPRD, 151 Buckingham Palace Road, London SW1W 9SZ, UK

**Abstract.** We describe experiments into the use of distributional similarity for acquiring lexical information from clinical free text, in particular notes typed by primary care physicians (general practitioners). We also present a novel approach to lexical acquisition from ‘sensitive’ text, which does not require the text to be manually anonymised – a very expensive process – and therefore allows much larger datasets to be used than would normally be possible.

## 1 Introduction

In the UK, almost all primary care physicians (general practitioners, or GPs) use one of a small number of computer systems for managing their patients’ medical records. These systems provide facilities for electronic storage, retrieval and modification of records, allowing GPs to enter orders for prescriptions, request tests, view laboratory results, read letters sent by hospital consultants, consult the notes of previous patient consultations and enter new notes. The records contain information in the form of codes, dates, numeric quantities, and free text. From a GP’s perspective, the main purpose of the records is to ensure individual patients obtain high quality medical care; however, records for individual patients are also used for medico-legal and health insurance purposes, and on a collective basis to support healthcare audits and to calculate incentivised payments to GPs for specific quality indicators. In addition, samples of records are collected regionally and nationally for other purposes, including health services research, epidemiological studies, and monitoring the safety of medications.

In these electronic medical records, Read codes [1] (the standard system for clinical terminology in the UK) are used to enter symptoms, test results, diagnoses and procedures (and also personal and administrative information). However, during consultations GPs usually do not input all relevant codes due to pressure of time, lack of incentive or relevant training, unwillingness to code symptoms that at the time seem not to be salient or where there is clinical uncertainty, and reluctance to code conditions that would normally be diagnosed by a specialist. In these situations the GP would be likely to type such information into the computer system in the form of unstructured free text. As well as uncoded symptoms and diagnoses, there is often a considerable amount of clinical information in the free text notes, including information on the severity

of symptoms, observations on examinations made by the GP, and information relating to diagnoses that the GP has ruled out. If all information were entered in a structured fashion it would be more amenable to automatic analysis, but the flexibility of the written language is necessary in order to capture the nuanced nature of much of this information and the variability between patients [9,18].

In recent years there has been a lot of interest in applying natural language processing techniques to clinical text. A further motivation in addition to those mentioned above (particularly in the USA) is to be able to automatically bill medical insurers for medications that have been administered and clinical procedures that have been performed. Some of the research activity has been centered around shared tasks, for example those organised by The Cincinnati Children's Hospital Medical Center involving assignment of clinical codes to radiology reports [16], and by i2b2 ('Informatics for Integrating Biology and the Bedside')<sup>1</sup> including identifying patient smoking status and extraction of medication information from discharge summaries [21,22]. The information retrieval conference TREC 2011<sup>2</sup> also included a shared task of retrieving clinical cases from narrative records [15].

Although these shared tasks and the deployed application systems that inspired them – as well as many other research efforts in clinical text mining (e.g. [19]) – involve documents of a variety of types (including discharge summaries, diagnostic test reports, and letters written by non-clinicians), the documents were produced from transcribed handwriting or dictation followed by post-transcription checking and editing, and consist of fairly standard language.

In contrast, notes typed by GPs do not go through any transcription or editing process, and are usually not carefully written. As mentioned above, the notes are written to support patient care within the GP practice, and not with the intention that they should be shared with other people. The notes often contain segments of informal language (not using medical terminology) summarising what the patient reported about their condition in their own words, and are usually typed by the GP during the consultation under pressure of time and with the competing requirement to attend simultaneously to the patient. Moreover, since these are notes taken in primary care, they potentially relate to very wide range of medical conditions.

These considerations make automatic processing of free text notes written by GPs a challenging task. However it is potentially a useful task, since epidemiological studies have shown that the free text contains important information relating to symptoms that is not available in the coded part of records [6,8]. In recent work, we have been applying natural language processing techniques to free text notes in order to extract certain kinds of uncoded information. We have created a corpus of clinical free text records in which we have annotated all mentions of the main symptoms of ovarian cancer [10]. We have been using this corpus to estimate the quantity of symptom-related information in records that

<sup>1</sup> <http://www.i2b2.org>

<sup>2</sup> <http://trec.nist.gov>

is not coded, and to develop techniques for automatically recognising mentions of these symptoms in free text [12].

Our current approach to recognising mentions of symptoms is based on manually curated word lists and relatively straightforward string matching technique, in which precision is optimised at the expense of recall. In this paper, we investigate methods based on distributional similarity for improving the recall of this and similar approaches, by automatically acquiring variant ways of expressing relevant words. Section 2 characterises the text data that we are using, summarises our approach to automatic symptom recognition, and motivates our use of distributional similarity methods. Section 3 discusses the datasets of free text records which we use in our investigation. We then go on to describe how we compute distributional similarity (Sect. 4) and the experiments we have conducted (Sect. 5). Finally, we conclude and outline directions for future work.

## 2 Background and Motivation

### 2.1 Primary Care Free Text

Our data is drawn from the General Practice Research Database (GPRD),<sup>3</sup> which contains records of about five million currently registered patients from around 630 general practices throughout the UK. GPRD data is used worldwide for research by the pharmaceutical industry, clinical research organisations, regulators, government departments and academic institutions. The free text in these records is of two main types: *notes* written by the GP, and *letters* sent to the practice by other agencies (primarily hospitals and mental health services) that the patient has been referred to. Letters are often entered by clerical staff in the practice, in the form of OCRed versions of the original hardcopy documents.

Notes written by GPs exhibit a very terse, telegraphic style with limited use of full sentence syntax; in particular, sentential subjects are very rare, and even finite verbs are uncommon (see (1a) below). Discourse connectives and conjunctions (e.g. *but*, *however*, *and*, *or*) are often omitted where they would normally be present, as in (1b).

- (1) a *chiropracter seen*
- b *sleeps well, low and tearful*

GPs also make widespread use of abbreviations and acronyms, some of which are conventionalised whereas others have a range of variations. For example, in (2a), it is likely that *bl* has been used to abbreviate ‘bleeding’ and *D* to abbreviate ‘diarrhoea’. In (2c), *cpn* is the standard acronym for ‘community psychiatric nurse’.

- (2) a *no bl no D*
- b *Had TAH and BSO 4/52 ago*
- c *prev h/o depression, ref cpn*

---

<sup>3</sup> <http://www.gprd.com>

Abbreviated words can be ambiguous, for example *occ* may mean any of *occurred*, *occasional* or *occupational* depending on the context. Spelling mistakes and anomalous tokenisation (e.g. missing spaces) are ubiquitous.

- (3) a *exmiane and futher hx needed and conisder mirena or orther form of contracpetion*
- b *Rx amoxicillin today,now feeling worse,burning up but feels cold,no energy*
- c *has 4 children, house needing renovation+working.*

Question marks and other shorthand means of indicating possibility or change are frequently used ambiguously.

- (4) a *Postnatal depression ? sl better on lofepramine*
- b *small ? 2 outer left breast tender*
- c *Shortness of breath +chest tightness*

Punctuation is used inconsistently or idiosyncratically (5a), and is sometimes missing in cases where it would normally be expected (5b).

- (5) a *has passed urine only once throughout whole day,/ since Fri/ yes weds/seen dr this am*
- b *pt requesting result of preg test result in mail box pt told neg*

In contrast, letters received from external agencies almost always use standard grammar and punctuation, and contain few spelling mistakes, idiosyncratic abbreviations, acronyms or shorthand expressions, as in (6a-c) below (unless the OCR software introduces mistakes).

- (6) a *It might be worthwhile continuing with some regular Nasal Steroids but if there is a continued deterioration then she could try an alternative preparation such as Accolate 20mgs bd.*
- b *She remained positive and was willing to discuss her problem and look at change. She is presently medication free and appears to be coping well.*
- c *She does describe some chest tightness and the symptoms are certainly worse first thing in the morning but some of her symptoms would seem to be related to a musculoskeletal element of the anterior chest.*

The large majority of tools for grammatical analysis of text which have been built by the natural language processing research community are based on statistical models and lexicons trained on edited text genres such as newspaper articles or scientific papers. The vocabulary and the way language is used in these genres is very different to GP-written free text notes, and we have observed that standard NLP tools make many errors when applied to these notes. Retraining the tools on GP notes would be very expensive, since we would require access to significant

amounts of text, which would have to be anonymised manually (to conceal all ‘sensitive’ information that could identify a patient) before it could be released by the GPRD and made available for part-of-speech or syntactic annotation. We therefore do not attempt any form of grammatical analysis.

## 2.2 Automatic Symptom Recognition

Koeling et al. [12] describe an investigation into automatic estimation of the incidence of symptoms using coded and free text information in primary care medical records. The experiments were run on records from the General Practice Research Database for 344 patients who were ultimately diagnosed as having ovarian cancer, and used an algorithm based on matching the textual descriptions of Read codes (for example ‘abdominal pain’) for the five most commonly presenting symptoms of the disease. The algorithm consists of three steps, performed in sequence:

1. Locate an occurrence of textual description of Read code in the text
2. Check whether there is evidence of negation
3. Determine whether the located textual description is within the scope of the negation

In the first step, sometimes an exact match of the textual description of the Read code is found in the text, but often the GP used a variant of the textual description. To deal with this, Koeling et al. manually compiled lists of common abbreviations of each word used in the Read code descriptions. Each list was augmented with a small number of semantically similar variants, selected from words which had a very similar distribution in medical record text. The algorithm allowed for spelling mistakes by matching words that are a small edit distance from the original word. For some symptoms the algorithm was able to double the amount of information extracted from the records compared to just considering coded information.

In Koeling et al.’s investigation, precision was optimised at the expense of recall; better results might therefore be obtained by improving the recall of the algorithm by producing more comprehensive lists of ways of expressing symptoms. Unfortunately, standard biomedical ontologies have poor coverage of many important phenomena in GP notes, especially abbreviations, acronyms, common shortenings of words and alternative spellings; they also do not cover informal but still medically-relevant language, as used by patients and reported by GPs (e.g. *tummy*). We are currently working on ways to automatically derive variations of surface realisations of words in order to obviate the need to compile lists of common variants manually, and to improve coverage of non-obvious variants. One approach we are exploring involves improving the process which identifies semantically similar variants of words so that it can find larger numbers of suitable candidates.

### 2.3 Distributional Similarity

It is often the case that semantically similar words are distributed similarly in text – that is, they occur in similar contexts. This idea goes back to observations by J.R. Firth who summarised it as “You shall know a word by the company it keeps” [4]. The *distributional similarity* of a pair of words is computed based on the shared contexts of the two words. Several measures of distributional similarity have been described in the literature. In the experiments described in this paper, we compute distributional similarity scores between words using Lin’s measure [13]. We use the scores to create a *distributional thesaurus*, in which each word is associated with a list of other words with the highest distributional similarity scores.

More formally, to encode context information a word  $w$  is associated with a set of features,  $f$ , each feature having an associated frequency. Each feature is a pair  $\langle r; x \rangle$  consisting of a relation name<sup>4</sup> and a word  $x$  that is related to  $w$  via  $r$ . To create a distributional thesaurus we compute similarity of contexts for every pair of words in the free text, but limited to those words that have a total feature frequency of at least  $N$ .<sup>5</sup> A thesaurus entry of size  $k$  for some word  $w$  consists of the  $k$  most similar words to  $w$ .

Weeds and Weir [24] provide empirical insights into what makes a ‘good’ distributional similarity measure for semantic similarity prediction. They observe that weighting features by pointwise mutual information appears to be beneficial. The intuition behind this is that the occurrence of a less common feature is more important in characterising a word than a more common feature. For example, the verb *to eat* is more selective and tells us more about the meaning of its grammatical arguments than the verb *to be*.

## 3 Datasets

We have access to two datasets of free text records from which we can construct distributional thesauruses. The first of these datasets, described in Sect. 3.1 below, consists of a relatively small amount of manually anonymised data which we have stored within our institution. The other dataset (Sect. 3.2) is rather different, in that it is much larger and is a ‘virtual dataset’ of un-anonymised text which we cannot view in raw form and can only access for running experiments via an intermediary, due to reasons of confidentiality.

### 3.1 Anonymised Dataset

As part of an interdisciplinary research project, PREP<sup>6</sup>, which is exploring the utility of free text in primary care medical records, we have been focussing on

<sup>4</sup> Previous studies have used grammatical relations (such as *subject* or *direct object*), or proximity relationships (such as *next word to the right*).

<sup>5</sup> For the experiments reported in this paper we set the frequency threshold  $N$  to 10 for smaller datasets, and 25 for larger ones.

<sup>6</sup> <http://www.informatics.sussex.ac.uk/research/projects/PREP/>

the records of women diagnosed with ovarian cancer. In this project, standard epidemiological methods were used to select a cohort of 344 patients and obtain from the General Practice Research Database all the records for these patients dating from 12 months before the diagnosis until 2 weeks after [20]. The resulting corpus consists of just over 6100 records, containing about 192K words. This corpus was manually anonymised by staff at the GPRD. Even though this dataset is large enough to answer many epidemiological questions, for most NLP purposes it is very small (especially considering the variety and amount of noise in the data). Fortunately, the GPRD have previously dealt with requests for anonymised free text and were also able to share with us text that had been anonymised for previous research projects. Even though this data is not relevant for studying ovarian cancer, it gives us more data that is representative of language in the database. The complete anonymised dataset contains around 3.5 million words, of which 3 million words are GP-written notes, and 500,000 words are letters.

### 3.2 Un-anonymised Dataset

Previous research into the distributional similarity technique has demonstrated that the quality of a distributional thesaurus improves as the amount of data it is derived from increases [2]. Given that manual anonymisation is expensive, it is unlikely that we will be able to obtain significantly more anonymised text in the near future. However, the full General Practice Research Database is orders of magnitude larger than our anonymised dataset (and is growing all the time, as more records are collected from participating practices). We would therefore like to be able to draw on this much larger source of data in order to create thesauruses.

Another reason for wanting to use more data is that our 3.5 million words of anonymised text is a very small amount in comparison to previous work in distributional thesaurus building. Moreover, it contains less useful information for thesaurus building even than that figure might suggest. As Sect. 2.1 argues, the text is difficult to parse, and so the distributional similarity computation between two words should probably be with respect to the words that are proximate to both rather than the words that are grammatically related to both. McCarthy et al. [14] found that to obtain similar results, ten times more input data was needed when using proximity relationships compared with using grammatical relations.

A further problem with the anonymised dataset is the fact that it is not a random sample of the database. The records from which the dataset is derived were selected for a small set of studies – most of which were concerned with cardiovascular disease – so the dataset is biased.

These problems of size and bias inspired us to devise a better approach. Since thesaurus creation only needs to establish relationships between words in free text records and takes no account of the surface form of sensitive words (i.e. whether these words are anonymised or not), we can expect to obtain similar thesauruses from un-anonymised text as from anonymised text. So if we can arrange for the thesaurus building software to run at the GPRD, on a machine

behind their firewall, the free text records themselves do not need to leave GPRD premises. Instead of anonymising the *input* data, any identifiable information would be removed from the *output* thesaurus before it left the GPRD. The set of words comprising the thesaurus would be much smaller than the input data, so this would be a much cheaper exercise.

This novel approach allows us to use a much larger, balanced sample of text from the General Practice Research Database. We are currently running experiments with a random sample of records comprising 55 million words. We refer to this sample as the ‘*un-anonymised* dataset’.

## 4 Creating Distributional Thesauruses

We compute the distributional similarity of a pair of words based on the extent to which the words occur in the same contexts in some body of text. In order to be able to record the surrounding contexts for each word, we need to define the set of relations used to compute the features.

Many approaches to distributional similarity use sets of grammatical relations. The motivation is that, for example, if the word *codeine* appears in the direct object relation to the verb *prescribe*, it is likely that other words that appear in the direct object relation with *prescribe* are semantically close to the word *codeine*. The fact that two words share one particular grammatical relation might not be of much consequence, but the more grammatical relations two words share, the more likely it is that those two words have a related meaning. However, as discussed in Sect. 2.1, standard parsers perform poorly on GP-written free text notes. We therefore use an alternative, more robust way of establishing a relation between two words in which we define a window around each word and relate it to other words in terms of proximity. In view of the telegraphic style of GP-written notes, we use a small window. We disregard apparent sentence boundaries and consider all the words that appear within the window. The relationships we use are summarized in Table 1.

**Table 1.** Proximity relationships capturing context

---

prev	previous word
prev_window	word within a distance of 2–5 words to the left
next	next word
next_window	word within a distance of 2–5 words to the right

---

## 5 Experiments

### 5.1 Variant Realisations

Section 2.2 summarised our approach to symptom recognition, which requires the ability to recognise variants of words. A key resource for this is an automatically

built distributional thesaurus, which allows us to harvest variant realisations automatically from relevant free text, thus avoiding the time-consuming and error-prone task of manually compiling lists of variants. In future studies, as well as recognising symptoms, we also will want to recognise mentions of specific tests, diagnoses and treatments in a similar way.

In GPRD text records, many commonly occurring words have a large number of distinct realisations. Reasons include the use of abbreviations, spelling errors and idiosyncratic capitalisation and punctuation. For example, we have found 15 variants of the word *patient*, including *pat.*, *pat*, *Pt*, *pt.*, *pateint*, *aptient* and *ptient*. Even though some of these are clearly unintentional misspellings, they are used in sufficiently consistent ways to be identified on the basis of the contexts in which they appear.

An example of how distributional methods can be used to identify candidate variants is the thesaurus entry for *abx* (a common abbreviation for *antibiotics*), shown in Fig. 1. The column to the right of the entry (*abx*) contains a list of the twenty words scored as most similar to *abx*. The numbers in the next column are scores that indicate the degree of similarity. The first thing we note about the related words in Fig. 1(a), is that the most similar word is the word that is being abbreviated (*antibiotics*). Even though the rest of the list does not contain many highly relevant words for this purpose, most of the words are related in one way or another. The list in Fig. 1(b) gives a strong indication that more data results in a better quality thesaurus. Out of the twenty words, more than half are variant realisations of *abx*, and half of the remainder are names of specific antibiotics.

It turns out that *abx* is a very frequently occurring term. Less frequently used terms do not always end up with such accurate results. However, the thesaurus usually gives an acceptable pre-selection of candidate variants. We are working on other methods to distinguish between the full version of an abbreviated word, other surface variations, related words and unrelated words.

## 5.2 Related Words

In addition to identifying variant realisations of a single word, some information extraction tasks might require the ability to recognise sets of *related* words expressing qualities or attributes of a clinically-relevant entity (such as a symptom, part of the body, or mental state). In contrast to symptoms, tests, diagnoses and treatments which are typically nouns, such words would usually be adjectives. In general, adjectives are more polysemous than nouns;<sup>7</sup> we might therefore expect slightly lower quality distributional thesaurus entries for adjectives, since polysemy is acknowledged problem for distributional approaches [5].

For example, we might need words related to the attribute *swollen*, such as *bruised*, *enlarged*, *inflamed*, *painful*, *red* and *sore*. Figure 2 shows thesaurus entries for this word; comparing (a) and (b), it is evident that the larger unanonymised dataset again gives good quality results.

<sup>7</sup> In WordNet 3.0, the average polysemy of nouns is 1.24, whereas that of adjectives is 1.40 (<http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html>)

abx	abx	1.0	abx	abx	1.0
	antibiotics	0.1231		abs	0.1574
	antibiotics.	0.1102		antibiotics	0.1509
	calpol	0.1088		ab	0.1416
	msu	0.1064		abx,	0.1330
	fluids	0.1053		a/b	0.1313
	steroids	0.1042		abx.	0.1303
	diuretic	0.1039		antibiotic	0.1276
	pred	0.1018		amoxil	0.1252
	diuretics	0.1016		amox	0.1248
	treat	0.0992		ab's	0.1241
	rx.	0.0988		amoxicillin	0.1238
	uti	0.0985		erythromycin	0.1196
	meds	0.0981		calpol	0.1181
	infection	0.0978		steroids	0.1178
	analgesia	0.0963		fluclox	0.1159
	observe	0.0958		treat	0.1158
	1/52	0.0949		analgesia	0.1150
	ibuprofen	0.0940		Abx	0.1136
	tomorrow	0.0936		antibiotics.	0.1099
	sos	0.0930		ABs	0.1089

(a)

(b)

**Fig. 1.** Distributional thesaurus entries for *abx*, derived from (a) the 3.5 million word anonymised dataset, and (b) the 55 million word un-anonymised dataset

For other information extraction-type tasks, we might want to relate words expressing similar qualities – for example a system that recognised whether a patient has reported pain (through words such as *discomfort*, *ache* or the word *pain* itself) would also probably need to determine gradation in the level of pain (from slight to severe), since *severe discomfort* might be of interest whereas *minimal pain* might not be. Figure 3 shows the thesaurus entry for *slight* derived from the 3.5 million word anonymised dataset. Since this is a small dataset, it should be expected that some of the words with high similarity scores are not relevant to this type of gradation; however, it is surprising that although in general most words are distributionally very similar to their antonyms, in this entry all of the relevant words are close in meaning to *slight* and there are no antonyms or close antonyms.

## 6 Discussion and Future Work

GP-written free text notes differ in many ways from the types of edited text that are commonly used in the natural language processing research community. Standard tools for grammatical analysis of text in general give poor results when applied to GP notes. However, our experiments into creating distributional similarity thesauruses from this text give promising results, and are able to

swollen	swollen	1.0	swollen	swollen	1.0
	swelling	0.1467		painful	0.1542
	painful	0.1458		inflamed	0.1416
	red	0.1440		red	0.1383
	thigh	0.1406		swelling	0.1365
	ankles	0.1291		inflamed	0.1357
	leg	0.1233		red,	0.1306
	hot	0.1200		swollen,	0.1251
	legs	0.1190		sore	0.1222
	tender	0.1183		redness	0.1179
	sob	0.1144		infected	0.1175
	ankle	0.1130		slightly	0.1171
	swollen,	0.1090		itchy	0.1158
	unwell	0.1087		sl	0.1141
	sore	0.1085		tender	0.1140
	oedema	0.1080		hot	0.1140
	dry	0.1055		swelling,	0.1119
	calf	0.1051		enlarged	0.1105
	crying	0.1046		tonsils	0.1091
	worse	0.1040		swollen.	0.1082

(a)

(b)

**Fig. 2.** Distributional thesaurus entries for *swollen*, derived from (a) the 3.5 million word anonymised dataset, and (b) the 55 million word un-anonymised dataset; in GP notes *sob* is an acronym for ‘shortness of breath’, and *sl* is commonly used to abbreviate ‘slight’ or ‘slightly’

slight	slight	1.0
	some	0.1291
	sl	0.1246
	mild	0.1199
	cough	0.1191
	c/o	0.1096
	slightly	0.1084
	minimal	0.1074
	swelling	0.1072
	ankle	0.1065
	little	0.1047
	tender	0.1027
	dry	0.0993

**Fig. 3.** Distributional thesaurus entry for *slight*, derived from the anonymised dataset; *c/o* is a standard abbreviation in GP notes for ‘complains of’

extract lexical information that is suitable for clinical text mining tasks, and which cannot be obtained from standard resources such as (bio)medical lexicons or ontologies.

Our successful use of distributional similarity for unsupervised lexical acquisition in the medical domain accords with other recent research efforts. A number of these have focussed on organising biomedical terminology with respect to (bio)medical ontologies and encyclopedias. In particular, Weeds et al. [23] apply distributional techniques to determine semantic proximity in order to classify terminology drawn from the GENIA corpus of biomedical research abstracts with respect to an associated ontology of terminological types. Fan and Friedman [3] reclassify UMLS concepts into broader semantic classes, using text from MEDLINE/PubMed titles and abstracts to compute distributional information. Van der Plas and Tiedemann [17] describe a system for identifying variants of Dutch terms in a medical encyclopedia using statistical information extracted from raw text, using word-aligned parallel corpora to establish co-occurrence relationships between terms in Dutch and their translations.

The study that is probably the most similar to ours – in that it is concerned with clinical text rather than edited biomedical text – is that of Henriksson et al. [7]. They describe an approach to assigning ICD-10 codes for diagnoses to uncoded medical records in Swedish which comprise text that is often semi-structured, but still contains many typing errors and non-standard abbreviations. Their approach computes word and code co-occurrences at the document level in order to capture information about the semantic similarity of individual words and codes, which is then used to classify uncoded documents.

The main novel aspect of our work is a new approach for acquiring lexical information from sensitive text which does not require the text to be anonymised before processing. This makes it possible to create thesauruses from un-anonymised text, and thus process much larger quantities of data than would normally be the case. However, even the un-anonymised dataset we are using is much smaller than the corpora of general text used in previous investigations into distributional similarity, which have shown that the larger the corpus the higher the quality of the resulting thesaurus [2,14,24]. We therefore intend to scale up this aspect of our processing – although at a purely practical level it does depend on having access to sufficiently powerful remote computing infrastructure, which in turn relies on purchasing decisions outside our control. In addition, although this approach means that the input text does not need to be anonymised, any identifiable information must be removed from the output thesaurus before it leaves the GPRD. While this should be relatively cheap exercise, we are currently investigating a set of safeguards which might mean that even this step might be unnecessary.

Previous work has found that distributional thesauruses can reflect latent aspects of the text they were built from. In particular, Koeling et al. [11] demonstrate that differences in the most frequent meaning of an ambiguous word between domains (for example the meanings of *bypass* in medical text and in current affairs news articles) can be predicted by creating separate distributional

thesauruses from documents in each domain. One of the reasons why this works is that for many words whose predominant meaning changes between domains, the most similar words in a thesaurus are specific to the domain (for example *catheter* might be similar to *bypass* but would only appear in medical text). This observation is relevant to our processing of clinical free text. As mentioned in Sect. 3.1, the dataset of anonymised text is a by-product of a relatively small number of research projects, and contains a large proportion of text relating to cardiovascular disease and prostate cancer. If we examine the thesaurus that was built using this dataset we can detect a bias towards these diseases. Although this is a weakness in our current experiments, we may be able to take advantage of it start processing larger amounts of GPRD data. We intend to explore how we can specialise our thesauruses for certain disease areas and see if it improves the utility of the resulting thesauruses.

In this paper we have not carried out an objective assessment of the thesauruses we have created, nor have we performed a quantitative evaluation of how they can contribute to a relevant application task. One of our next steps will be to conduct an extrinsic evaluation of thesaurus data applied to a symptom recognition problem. We already have a suitable prototype system (outlined in Sect. 2.2) which we can adapt, and also an annotated corpus [10]. This should constitute a good test of our techniques.

**Acknowledgments.** We are grateful to Jackie Cassell, Clare Laxton and Rosemary Tate for advice and suggestions on the direction of this research. The work was supported by the Wellcome Trust [086105/Z/08/Z]. Access to the GPRD database was funded through the Medical Research Council's licence agreement with MHRA. The authors were independent from the funder and sponsor, who had no role in conduct, analysis or the decision to publish. This study is based in part on data from the Full Feature General Practice Research Database obtained under licence from the UK Medicines and Healthcare Products Regulatory Agency. However, the interpretation and conclusions contained in this study are those of the authors alone.

## References

1. Bentley, T., Price, C., Brown, P.: Structural and lexical features of successive versions of the Read Codes. In: Teasdale, S. (ed.) *Proceedings of the Annual Conference of The Primary Health Care Specialist Group of the British Computer Society*, Worcester, UK, pp. 91–103 (1996),  
<http://www.phcsg.org/main/pastconf/camb96/readcode.htm>
2. Curran, J., Moens, M.: Scaling context space. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, pp. 231–238 (2002)
3. Fan, J.W., Friedman, C.: Semantic classification of biomedical concepts using distributional similarity. *JAMIA* 14(4), 467–477 (2007)
4. Firth, J.R.: *A synopsis of linguistic theory 1930-1955. Studies in Linguistic Analysis*, 1–32 (1957)

5. Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., Wang, Z.: New experiments in distributional representations of synonymy. In: Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL), Ann Arbor, MI, pp. 25–32 (2005)
6. Hamilton, W., Peters, T., Bankhead, C., Sharp, D.: Risk of ovarian cancer in women with symptoms in primary care: population based case-control study. *British Medical Journal* 339, b2998 (2009)
7. Henriksson, A., Hassel, M., Kvist, M.: Diagnosis Code Assignment Support using Random Indexing of Patient Records a Qualitative Feasibility Study. In: Peleg, M., Lavrač, N., Combi, C. (eds.) *AIME 2011. LNCS*, vol. 6747, pp. 348–352. Springer, Heidelberg (2011)
8. Johansen, M., Scholl, J., Hasvold, P., Ellingsen, G., Bellika, J.: “Garbage in, garbage out” – extracting disease surveillance data from EPR systems in primary care. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, San Diego, CA, pp. 525–534 (2008)
9. Kalra, D., Ingram, D.: Electronic health records. In: Zielinski, K., Dupлага, M., Ingram, D. (eds.) *Information Technology Solutions for Healthcare*. Springer, Heidelberg (2006), <http://eprints.ucl.ac.uk/1598/>
10. Koeling, R., Carroll, J., Tate, A.R., Nicholson, A.: Annotating a corpus of clinical text records for learning to recognize symptoms automatically. In: Proceedings of the 3rd Louhi Workshop on Text and Data Mining of Health Documents, Bled, Slovenia, pp. 43–50 (2011)
11. Koeling, R., McCarthy, D., Carroll, J.: Domain-specific sense distributions and predominant sense acquisition. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada, pp. 419–426 (2005)
12. Koeling, R., Tate, A.R., Carroll, J.: Automatically estimating the incidence of symptoms recorded in GP free text notes. In: Proceedings of the First International Workshop on Managing Interoperability and Complexity in Health Systems, Glasgow, UK, pp. 43–50 (2011)
13. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the ACL, Montreal, Canada, pp. 768–774 (1998)
14. McCarthy, D., Koeling, R., Weeds, J., Carroll, J.: Unsupervised acquisition of predominant word senses. *Computational Linguistics* 33(4), 553–590 (2007)
15. NIST: Proceedings of the 2011 Text REtrieval Conference (TREC 2011). National Institute for Standards in Technology, Gaithersburg, MD (2011)
16. Pestian, J., Brew, C., Matykiewicz, P., Hovermale, D., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of BioNLP 2007: Biological, Translational, and Clinical Language Processing, Prague, Czech Republic, pp. 97–104 (2007)
17. van der Plas, L., Tiedemann, J.: Finding medical term variations using parallel corpora and distributional similarity. In: Proceedings of the 6th Workshop on Ontologies and Lexical Resources, Beijing, China, pp. 28–37 (2010)
18. Resnik, P., Niv, M., Nossal, M., Kapit, A., Toren, R.: Communication of clinically relevant information in electronic health records: a comparison between structured data and unrestricted physician language. *Perspectives in Health Information Management* (2008)
19. Roberts, A., Gaizauskas, R., Hepple, M., Guo, Y.: Mining clinical relationships from patient narratives. *BMC Bioinformatics* 9(suppl. 11), S3 (2008)

20. Tate, A.R., Martin, A., Ali, A., Cassell, J.: Using free text information to explore how and when GPs code a diagnosis of ovarian cancer: an observational study using primary care records of patients with ovarian cancer. *BMJ. Open.* (2011) doi:10.1136/bmjopen-2010-000025
21. Uzuner, Ö., Goldstein, I., Luo, Y., Kohane, I.: Identifying patient smoking status from medical discharge records. *JAMIA* 15(1), 14–24 (2008)
22. Uzuner, Ö., Solti, I., Cadag, E.: Extracting medication information from clinical text. *JAMIA* 17(5), 514–518 (2010)
23. Weeds, J., Dowdall, J., Schneider, G., Keller, B., Weir, D.: Using distributional similarity to organise biomedical terminology. *Terminology* 11(1), 107–141 (2005)
24. Weeds, J., Weir, D.: Co-occurrence Retrieval: a flexible framework for lexical distributional similarity. *Computational Linguistics* 31(4), 439–476 (2005)

# Developing an Algorithm for Mining Semantics in Texts

Minhua Huang and Robert M. Haralick

Computer Science Department  
The Graduate School and University Center  
The City University of New York  
New York, NY 10016

**Abstract.** This paper discusses an algorithm for identifying semantic arguments of a verb, word senses of a polysemous word, noun phrases in a sentence. The heart of the algorithm is a probabilistic graphical model. In contrast with other existed graphical models, such as Naive Bayes models, CRFs, HMMs, and MEMMs, this model determines a sequence of optimal class assignments among  $M$  choices for a sequence of  $N$  input symbols without using dynamic programming, running fast— $O(MN)$ , and taking less memory space— $O(M)$ . Experiments conducted on standard data sets show encourage results.

**Keywords:** semantics, algorithm, text pattern, probabilistic graphical model, semantic argument, word sense, NP chunk.

## 1 Introduction

Text patterns, such as semantic arguments of a verb, the meaning of a polysemous word, and noun phrases of a sentence, are essential patterns for capturing semantics in texts. For example, semantic arguments of a verb can be used to answer the questions of who, what, when, where, and why; the sense of a polysemous word can be used to understand the meaning of the word; noun phrases of a sentence can be used to extract the concepts of a sentence. Therefore, mining semantics patterns is useful.

In this paper, we discuss an algorithm for recognizing these text patterns. The heart of the algorithm is a probabilistic graphical model. In contrast with the existing graphical models, such as HMMs [1], MEMMs [2], or CRFs [3] that lead to an optimization for a sequence of class assignments to optimize the joint or conditional probability of the class assignments given a sequence of symbols using an implicit gain function where the gain is one if all class assignments are correct and zero if any assignment is wrong. Our model uses the gain function that gives partial credit for each correct assignment. With this criterion, the running time is reduced from  $O(M^2N)$  to  $O(MN)$  and the memory space is reduced from  $O(MN)$  to  $O(M)$ , where  $M$  is the cardinality of the set of class assignments and  $N$  is the length of an input symbol sequence. Moreover, by applying the method on standard data sets for recognizing three patterns, we find that our results exceed or approach the current state of the art.

## 2 Creating the Model

### 2.1 Economic Gain Function

Let  $s = < s_1, \dots, s_N >$  be a sequence of  $N$  symbols. Let  $C$  be a set of  $M$  classes,  $C = \{C_1, \dots, C_M\}$ . Let  $c = < c_1, \dots, c_N >$  be a sequence of assigned classes. Let  $c^T = < c_1^T, \dots, c_N^T >$  be a sequence of true classes. Let  $e$  be the economic gain function  $e : C^N \times C^N \rightarrow \mathcal{R}^+$ . For the existing probabilistic graphical models, such as HMMs [1], MEMMs [2], or CRFs [3], the economic gain function  $e(c_1^T, \dots, c_N^T, c_1, \dots, c_N) = 1$  when  $c_1^T, \dots, c_N^T = c_1, \dots, c_N$  and  $e(c_1^T, \dots, c_N^T, c_1, \dots, c_N) = 0$  otherwise. That is, the gain function specifies a gain of one if all the class assignments are correct and zero if one or more of the class assignments are wrong. No partial credit is given for some correct assignments. As a consequence, entire wrong classification subsequences can be produced around an incorrectly assigned symbol just because it has been subjected to random noise or perturbations. We use a gain function that gains some value for each correct assignment. It is defined by:

$$e(c_1^T, \dots, c_N^T, c_1, \dots, c_N) = \sum_{n=1}^N e(c_n, c_n^T) \quad (1)$$

where :  $e(c_n, c_n^T) = \begin{cases} > 0, & \text{when } c_n^T = c_n \\ 0, & \text{otherwise} \end{cases}$

Compared with the previous gain function implicitly employed by the HMMs, MEMMs, and CRFs, our gain function gives partial credits for correct assignments. In fact, this gain function is also implicitly employed by the context independent Bayes model where each symbol class pair is independent of all the other symbol class pairs. To maximize the expected gain under equation (1), we have:

$$E(e) = \sum_{n=1}^N \sum_{c_n^T} e(c_n, c_n^T) p(c_n^T, s_1, \dots, s_N)$$

When the gain matrix is diagonal and positive, then:

$$E(e) = \sum_{n=1}^N e(c_n, c_n) p(c_n, s_1, \dots, s_N)$$

When the gain matrix is the identity, assigning the value 1 for a correct assignment and the value 0 for an incorrect assignment, then:

$$E(e) = \sum_{n=1}^N p(c_n, s_1, \dots, s_N)$$

In this case, maximizing the expected gain is associated with maximizing  $p(c_n | s_1, \dots, s_N)$ , where  $n = 1, \dots, N$ .

$$\max(E(e)) = \sum_{n=1}^N \max_{c_n \in C} p(c_n, s_1, \dots, s_N)$$

## 2.2 Building the Model

To find the assigned class sequence  $\langle c_1, \dots, c_N \rangle$  for the input sequence  $\langle s_1, \dots, s_N \rangle$  that maximizes the expected gain, we only need to find an assigned class  $c_n, n = 1, \dots, N$  that maximizes the joint probability function  $p(c_n, s_1, \dots, s_N)$ . This leads to the mathematical representation for the probability in Equation (2).

$$p(c_1, \dots, c_N | s_1, \dots, s_N) = \frac{\prod_{n=1}^N p(s_{n-1} | s_n, c_n) p(s_{n+1} | s_n, c_n) p(s_n | c_n) p(c_n)}{\sum_{c_k \in C} \prod_{k=1}^N p(s_{k-1} | s_k, c_k) p(s_{k+1} | s_k, c_k) p(s_k | c_k) p(c_k)} \quad (2)$$

## 2.3 Finding $\langle c_1^*, \dots, c_N^* \rangle$

By Equation (2), to find a class sequence  $\langle c_1^*, \dots, c_N^* \rangle$  given a sequence of symbols  $\langle s_1, \dots, s_N \rangle$ , we only need to find  $c_n^*$  for  $s_n$  individually. Note, the denominator in (2) is a constant. Therefore, it does not effect a decision for assigning  $c_i$  to  $s_i$ .

$$\langle c_1^*, c_2^*, \dots, c_N^* \rangle = \prod_{n=1}^N \underset{c_n \in C}{\operatorname{argmax}} p(s_{n-1} | s_n, c_n) p(s_{n+1} | s_n, c_n) p(s_n | c_n) p(c_n) \quad (3)$$

$$\text{Letting } f(s_{n-1}, s_n, s_{n+1}, c_n) = p(s_{n-1} | s_n, c_n) p(s_{n+1} | s_n, c_n) p(s_n | c_n) p(c_n)$$

$$\langle c_1^*, c_2^*, \dots, c_N^* \rangle = \prod_{n=1}^N \underset{c_n \in C}{\operatorname{argmax}} f(s_{n-1}, s_n, s_{n+1}, c_n)$$

## 2.4 Complexities

*HMMs* or *CRFs* employ dynamic programming to obtain a sequence optimal of classes for a sequence of symbols by computing a joint probability  $p(s_1 \dots s_n | c_1 \dots c_N)$  or a conditional probability  $p(c_1 \dots c_N | s_1 \dots s_n)$ . By dynamic programming, an optimal class for the current symbol is obtained based on an optimal class of the previous symbol. Therefore, the optimal class for the last symbol is determined after the last symbol has been reached. The optimal class sequence needs to be determined by tracing back from the last optimal class to the first optimal class. For each symbol, information for  $M$  classes needs to be stored. Hence, for a sequence of  $N$  symbols, we need to have  $O(M^2 N)$  time complexity and  $O(M * N)$  memory complexity. For our model, by equation (3), for each symbol  $s_n$ , we need to assign a  $c_n$ , such that

$$f(s_{n-1}, s_n, s_{n+1}, c_n) \geq f(s_{n-1}, s_n, s_{n+1}, c'_n), \quad c'_n \in C$$

**Time Complexity.** To compute  $f(s_{n-1}, s_n, s_{n+1}, c_n)$ , we need to have four multiplications. To obtain the maximum probability value we require  $M - 1$  comparisons. In the case of a sequence of  $N$  symbols, we need

$$T_c = 4 * N * (M - 1) * (L - 1) = O(N * M * L)$$

**Memory Complexity.** Because the global maximum probability is determined by each local maximal probability, for a path of  $N$  symbols, we only need to store the information of the current node. That is, we need only store  $M$  probability values in order to find the maximal probability value. Therefore,

$$M_c = M = O(M)$$

**Comparisons.** We compute ratios of time complexity and memory complexity of our model to *HMMs* and *CRFs* to see differences. By observing these two ratios, we see that if we need to recognize a sequence of  $N$  symbols with  $M$  categories, our model only take  $\frac{1}{M}$  time and  $\frac{1}{N}$  memory space of *HMMs* or *CRFs*. For example, if the cardinality of  $C$  is ( $M = 8$ ), for a sequence of sixteen symbols ( $N = 30$ ), our method only needs to have  $\frac{1}{8}$  time and  $\frac{1}{30}$  memory space of a *HMM* or a *CRF* to recognize this sequence.

### Ratio of Time Complexity

$$\frac{NM}{M^2N} = \frac{1}{M}$$

### Ratio of Memory Complexity

$$\frac{M}{M * N} = \frac{1}{N}$$

## 3 Three Tasks

### 3.1 Identifying Semantic Arguments of a Verb

Let  $T = (V, E, r, A, L)$  be a labeled rooted tree associated with a sentence, where  $V$  is a set of vertices,  $E$  is a set of edges,  $E \subseteq V \times V$ ,  $r$  is the root,  $A$  is an alphabet defined by [4], and  $L$  is a labeling function  $L : V \rightarrow A$  that assigns labels to vertices. The parse tree of the sentence takes the form of  $T$ . Let  $\pi$  be a set of labels, s.t.  $\pi \subseteq A$ . Let  $C = \{C_1, C_2\}$  be a set of classes, where  $C_1$  represents that a path will be extended from the current node to an adjacent node;  $C_2$  represents that a path will not be extended from the current node to an adjacent node.

### The Procedure

- Form a path  $\mathcal{P}(x) = \tau_1, \rightarrow \dots, \rightarrow \tau_K$ ,  $x \in V$ ,  $L(x) \in \pi$ , and  $x$  is not a node in  $\mathcal{P}'(y)$ ,  $\mathcal{P}'(y)$  is a path that has been already formed previously.

$$\langle \tau_1, \dots, \tau_K \rangle = \underset{b_1, \dots, b_K}{\operatorname{argmax}} p(c_1, \dots, c_K, b_1, \dots, b_K)$$

Note,  $c_k \in C$ ,  $b_k \in V$ ,  $b_{k-1}b_k \in E$ .

- Form a set of roots  $R(x) = \{r_i | i = 1 \dots M\}$  from  $\mathcal{P}(x)$ , where  $r_i \leq \tau_k$ .

- For all siblings of  $\tau_k$ , find  $z$ , s.t.  $L(z) \notin \pi$  and  $z \notin \{\tau_k | k = 1, \dots, K\}$ , then  $R(x) \leftarrow R(x) \cup \{z\}$
- For all children of  $\tau_k$ , find  $y$ , s.t.  $L(y) \notin \pi$  and  $y \notin \{\tau_k | k = 1, \dots, K\}$ , then  $R(x) \leftarrow R(x) \cup \{y\}$
- Find a rooted forest  $F(x) = \{T_i | i \in \{1, \dots, I\}\}$ ,
  - Each  $T_i$  is induced from the root  $r_i$  by all its co-dependents.
  - For each  $T_i \in F(x)$ , the leaves  $\{l_i^1, \dots, l_i^K\}$  correspond to one of the semantic arguments of  $x$ .

### 3.2 Identifying the Sense of a Word

Let  $S = \langle s_1, \dots, s_t, \dots, s_N \rangle$  be a sequence of symbols associated with a sentence,  $s_t \in S$  be a given ambiguous symbol that needs to be disambiguated. Let  $C = \{C_m | m = 1, \dots, M\}$ ,  $C$  be a set of predefined senses of the ambiguous symbol  $s_t$ .

**The Contexts.** The Context of an ambiguous symbol  $s_t$  is a  $k$  – tuple, represented by  $T_t$ . Each element in  $T_t$  is a symbol in  $S$ ,  $T_t = (t_1, \dots, t_K)$ ,  $t_k \in S$ , and  $K \leq N$ .

### The Procedure

- Find the context  $T_t$  for  $s_t$ .
- Find a sequence of classes  $\langle c_1^*, \dots, c_K^* \rangle$  for  $T_t = (t_1, \dots, t_K)$ , s.t.

$$\langle c_1^*, \dots, c_K^* \rangle = \underset{c_1, \dots, c_K}{\operatorname{argmax}} p(c_1, \dots, c_K | t_1, \dots, t_K)$$

- Assign  $C_j$  to  $s_t$  if and only if

$$\#\{k | c_k = C_j\} \geq \#\{k | c_k = C_m\}, \quad m = 1, \dots, M$$

### 3.3 Identifying Noun Phrases

Let  $S$  be a sequence of symbols associated with a sentence,  $S = \langle s_1, \dots, s_i, \dots, s_N \rangle$ , where  $s_i$  is the pair (word, its speech tag). Let  $C$  be a set of classes,  $C = \{C_1, C_2, C_3\}$ , where  $C_1$  represents that a symbol is inside a noun phrase,  $C_2$  represents that a symbol is not in a noun phrase, and  $C_3$  represents that a symbol starts at a new noun phrase.

**Building Blocks.**  $\mathcal{B}$  is a block if and only if:

1. For some  $i \leq j$ ,  $\mathcal{B} = \langle (s_i, c_i), (s_{i+1}, c_{i+1}), \dots, (s_j, c_j) \rangle$
2.  $c_i \in \{C_1, C_3\}$
3.  $c_n = C_1$ ,  $n = i + 1, \dots, j$
4. For some  $\mathcal{B}'$ , if  $\mathcal{B}' \supseteq \mathcal{B}$  and  $\mathcal{B}'$  satisfies 1, 2, 3 then  $\mathcal{B}' \subseteq \mathcal{B}$

## The Procedure

- Find a sequence of classes  $\langle c_1^*, \dots, c_N^* \rangle$ , s.t.
- $$\langle c_1^*, \dots, c_N^* \rangle = \underset{c_1, \dots, c_N}{\operatorname{argmax}} p(c_1, \dots, c_N | s_1, \dots, c_N)$$
- Find  $\{B_1, \dots, B_M\}$ , where each  $B_m$  is a block satisfying the definition of  $\mathcal{B}$ .

## 4 Evaluation

In order to evaluate our method, we have conducted two types of tests. In the first type of tests, we test our method on three tasks on standard data sets and compare the results published by other researchers on the same data sets. In the second type of tests, we implement the context independent Naive Bayes method and test it on the selected tasks and compare the results with our method.

### 4.1 Experiments Set Up

**Data Sets.** Data sets that we have selected for our method are *WSJ* data from the Penn TreeBank and the PropBank [4], data developed by [5] [6], and *WSJ* data from the Penn TreeBank [7] and CoNLL-2000 Shared Task [8]. Our reasons for using these data sets were that they have been studied by numbers of other researchers and many results have been published over the years.

**Evaluation Metrics.** The evaluation metrics designed for testing the first and the third tasks were *precision* , *recall*, *f-measure* ( $F_1$ ) and for testing the second task were *accuracy*. The reason of selecting different evaluation methods was based on the design of classes described in sections 3.1, 3.2, and 3.3<sup>1</sup>. One of the classes was not needed to be evaluated in task one and three while all classes were needed to be evaluated in task two.

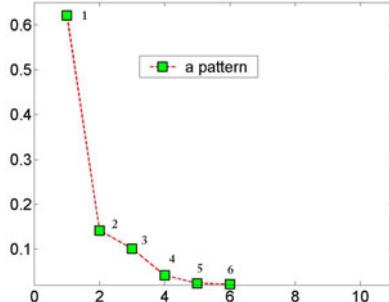
**Training Set and Testing Set Distributions.** We have used a 10-fold cross validation technique for obtaining our result for all experiments.

### 4.2 The First Type of Test

**First Task Results.** The data set, the section 00 of *WSJ* from Penn Treebank and PropBank [4], was used for testing in the first task. A total of 223 sentences is in files 20, 37, 49, and 89. Associated with each of these sentences, it was an automatically determined parse tree provided by Penn Treebank. These parse trees had an average accuracy of 95.0%. Among these sentences, there were 621 verbs. Each verb had an average of three semantic arguments. Hence about

---

<sup>1</sup> There was one class that represented none of these classes in the first task and third task while every class was a distinct sense in the second task.



**Fig. 1.** Six patterns of paths: 1.  $VBZ(VBD, VBG, VBP, VBN, VB) \rightarrow VP$ , 2.  $MD(TO) \rightarrow VP \rightarrow VP \rightarrow VB$ , 3.  $VBP(VBZ, VBD) \rightarrow VP \rightarrow VP \rightarrow VBN$ , 4.  $VBD(VBZ, VBN) \rightarrow VP \rightarrow RB \rightarrow VP \rightarrow VB$ , 5.  $TO \rightarrow VP \rightarrow VP \rightarrow VB \rightarrow VP \rightarrow VBN$ , 6.  $MD \rightarrow VP \rightarrow RB \rightarrow VP \rightarrow VBP(VB) \rightarrow VP \rightarrow VBN$

2000 semantic arguments were used. The semantic arguments were provided by PropBank. These were created manually.

Among 621 verbs, about 560 verbs were used for obtaining probability values while about 60 verbs were used to form paths based on these probability values. Some of the paths were listed on Figure 1. They were obtained based on the procedure described in Section 3.3 by applying 10-fold cross validation technique. We noticed that 86% paths fell into the first three patterns in Figure 1.

After forming a path for a verb in the test instances, a set of roots were found. From these roots, a set of labeled rooted subtrees, whose leaves were associating with semantic arguments of the verb, was formed. The test results were shown in Table 1. On the average, each time among  $\frac{1}{10}$  of the semantic arguments were classified, about 93% semantic arguments were correctly identified and 7% semantic arguments were mistakenly identified. By checking these classified instances, we found that our method was very effective in the case of a semantic argument being a sequence of consecutive words. However, if a semantic argument consisted of two or more word fragments, separated by some phrases, our algorithm was less effective. The reason was that these phrases were parts of leaves of a tree induced from a root determined by our algorithm. This suggests that in order to exclude phrases from a semantic argument, we need to develop a method so that a set of subroots needs to be found. Each of them corresponds to a fragment of a semantic argument. Moreover, other misclassified instances were generated by errors carried in original syntactic trees.

**Second Task Results.** We tested our method for identifying the sense of a word on the data sets *line*, *hard*, *serve*, and *interest*. The senses' descriptions and instances' distributions could be found in [5] and [6]. In these data sets, *line* and *interest* were polysemous nouns, *hard* was a polysemous adjective, and

**Table 1.** The First task on *WSJ* data

Files	Precision	Recall	F-Measure
20, 37, 49, 89	%	%	%
F-measure	92.335	94.1675	93.2512
Std	0.6195	0.5174	0.4605

*serve* was a polysemous verb. In our experiment, *line* had 6 senses, *serve* had 4 senses, *hard* had 3 senses, and *interest* had 3 senses (3 other senses were omitted due to insufficient number of instances). The test metric that we have used was *accuracy*.

We formed the context of each given target word by including the left four open class words and the right four open class words combining with the left word and the right word for each of these words. The test results were shown in Table 2.

**Table 2.** The second task on *line-serve-hard-interest* data

Ambiguous word	Senses	Accuracy %	Standard deviation %
<i>line</i> (noun)	6	81.16	1.92
	3	85.25	2.13
<i>serve</i> (verb)	4	79.80	1.90
<i>hard</i> (adjective)	3	82.88	3.10
<i>interest</i> (noun)	3	92.10	2.21

We found that misclassified instances were primarily generated by the ambiguity of context words. For example in Table 2, comparing with the three sense noun *interest* and the three sense noun *line* obtained by selecting three senses at each time from six senses and examining all twenty combinations, we found that the accuracy of the word *interest* was almost 9% higher than the accuracy for the word *line*. Moreover, by examining accuracies generated from each combination for the word *line*, we found that some combination had the highest average accuracy, for instance  $S_1S_2S_4$  had an average accuracy of 91.7% while some combination had the lowest average accuracy, for instance  $S_1S_3S_5$  had an average accuracy of 77.1%. The difference was almost 20%. By carefully checking these misclassified instances, We learned that if two senses were similar to each other, there were more chances that their contexts consisted of the same words. As a consequence, the misclassification rate increases.

Moreover, by observing the outputs of two polysemous nouns *line* and *interest*, we found that as the number of senses of a polysemous noun increasing, the accuracy decreased. This suggested that nouns with a larger number of senses were more difficult to recognize than nouns with small number of senses by our algorithm. Furthermore, by comparing accuracies, we noticed that nouns

were relatively easier to identify than adjectives or verbs. From comparing the standard deviations we noticed that accuracies generated by our algorithm on adjectives had a larger variance than that on nouns or verbs.

**Comparisons.** Our results are better than the results reported by other WSD researchers [9] and [5]. Our method achieves an average accuracy of 81.12% for identifying the six sense noun *line* using 2450 training context words while the method proposed by [5] achieves the average accuracy 73% using 8900 training context words. Moreover, an experiment using the Latent Semantic Analysis method conducted by [9] achieves an average accuracy of 75% for identifying only three senses of *line*. The comparisons are shown in Table 3.

**Table 3.** Comparisons on recognizing word sense on **line** data

Method	Bayesian [5]	The algorithm	LSA [9]	Context vector [5]	Neural Network [5]
Accuracy	71	81	75	72	76

**Third Task Results.** Three types of symbols were designed for identifying NP chunks on CoNLL-2000 Shared Task data set. They were the lexicon of a word, the POS tag of a word, and the lexicon and the part of speech (POS) tag of a word. The results were shown in the second row of Table 4. By comparing the results, we noticed that if the model was built only on the lexical information, it had the lowest performance 89.75%. The model's performance improved 3% if it was constructed by POS tags. The model achieved the best performance of 95.59% if both lexicon and POS tags were included.

Different from the first experiment, the second experiment on the WSJ data from Penn Treebank used only one type of symbol: the lexicon and the POS tag of a word. The main reason for using this data set was that we wanted to see whether the performance of our model could be improved when it was built on more data. In this case, the training set was seven times larger than the CoNLL-2000 shared task training data set. The test results were shown in the third row of Table 3. Note, data inside parentheses in the table represented standard deviation.

Compared with the results on these two data sets, we noticed that the average precision was improved about 2.7% from 95.15% to 97.73%. The average recall was improved about 2.8% from 96.05% to 98.65%. The average F-measure was improved about 2.7% from 95.59% to 98.2% as the training sets expanded to seven times larger. This suggested that the larger the training sets, the better the results.

**Comparisons.** Table 5 shows the best performances of the related methods [1] [10] [11] [12] [13] [14] on the CoNLL-2000 shared task data. Among of these methods, the role based learning method achieves the worst F-measure performance and our method achieves the best F-measure performance.

**Table 4.** The test results on the CoNLL-2000 and WSJ data

Data	Symbol type	Precision	Recall	F-measure
		%	%	%
CoNLL-2000	Lexicon + POS	95.15	96.05	95.59
	POS	92.27	93.76	92.76
	Lexicon	86.27	93.35	89.75
WSJ	Lexicon + POS	97.73	98.65	98.18
		(0.19)	(0.14)	(0.08)

**Table 5.** Comparisons for different methods on the CoNLL-2000 data set

Method	RBL [13]	HMM [1]	NB <sup>2</sup>	MEMM [10]	VP [11]	CRF [10]	SVM [12]	the algorithm
F-measure	91.54	93.52	93.69	93.70	93.74	94.38	94.45	95.74

### 4.3 The Second Type of Test

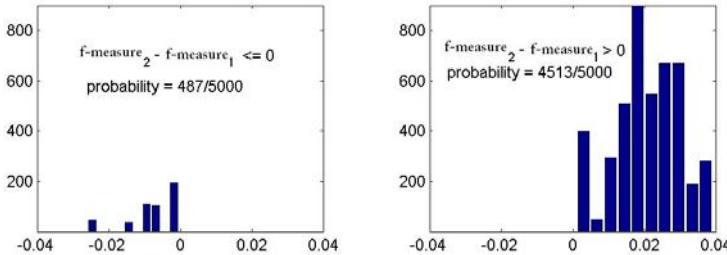
**The Context Independent Bayes Model.** We implemented the context independent Bayes model represented by  $p(c_1, \dots, c_N | s_1, \dots, s_N) = \prod_{i=1}^N p(s_i | c_i)$ . As mentioned in Section 2.1, this model also implicitly employed the economic gain function where each symbol class pair was independent of all the other symbol class pairs.

In order to compare with the context independent Bayes model with our method, we conducted experiments on two date sets for two tasks: CONLL-2000 data set for identifying noun phrases in a sentence and *interest* data for identifying the sense of the word *interest*. We still used 10 – *folder* cross validation technique to obtain an average.

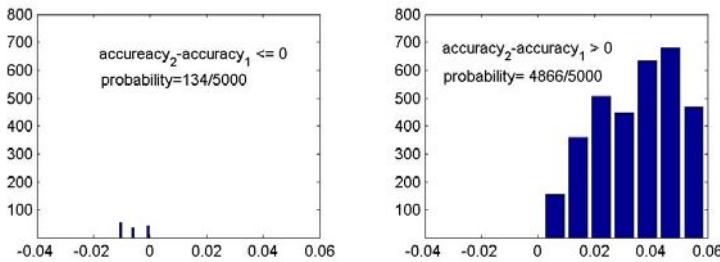
Figure 2 and Figure 3 showed the results. In these figures,  $f - measure_2 / accuracy_2$  represented an F-measure/accuracy obtained by our method while  $f - measure_1 / accuracy_1$  represented an F-measure/accuracy obtained by the context independent Bayes model. In each case, we put all results generated by these two methods into a pool, and randomly selected a pair of results (each component in the pair was generated by different methods) and computed their difference. We ran this procedure for five thousand times.

In Figure 2, the left side of the figure represented the number of occurrences that an F-measure obtained by our method was lesser than an F-measure obtained by the context independent Naive Bayes while the right side of the figure represented the number of occurrences that an F-measure obtained by our method was greater than an F-measure obtained by the context independent Naive Bayes.

By using  $\frac{mean\{f - measure_2 - f - measure_1\}}{mean\{f - measure_1\}}$ , we found that our method achieved a 2.24% better average F-measure than the context independent Bayes model on identifying NP chunks with the confidence of 90.26%. In the same way, by observing Figure 3, we found that our method achieved a 5.44% better average



**Fig. 2.** Comparisons of Context Independent Bayes with our method on the CoNLL-2000 data set



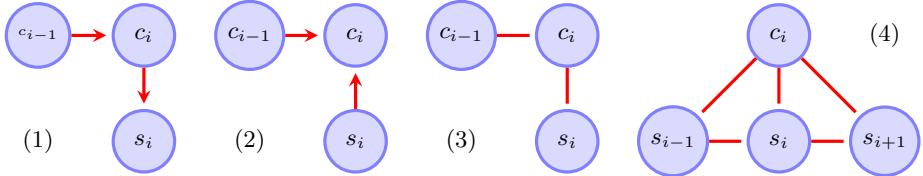
**Fig. 3.** Comparisons of Context Independent Bayes with our method on *interest* data set

accuracy than the context independent Bayes model on identifying the sense of the word *interest* with the confidence of 96.98%.

#### 4.4 Discussion

**Different Graphical Representations.** Currently existing graphical models used by most researchers are HMMs [2] [1], MEMMs[2], and CRFs[3] [10]. These models are built to obtain an optimal sequence of  $N$  classes  $c = < c_1, \dots, c_N >$  from a sequence of  $N$  symbols  $s = < s_1, \dots, s_N >$  by finding the maximum value of the joint probability  $p(c, s)$  or the conditional probability  $p(c|s)$ . These graphical models are shown in Figure 4. While HMMs and MEMMs are directed graphical models, CRFs and our model are undirected graphical models. While others have a link from  $c_{i-1}$  to  $c_i$ , our model links  $c_i$  and  $s_{i+1}$  and links  $c_i$  and  $s_{i-1}$ . We believe that  $c_i$  can be better predicated from  $s_{i-1}$  and  $s_{i+1}$  rather than  $c_{i-1}$  when symbols contain several types of information. For example, in the case of NP chunking, POS tag information carried on a symbol is much useful than the class information assigned on the previous symbol.

**Different Assumptions.** In the conditional independence graph,  $s_i$ ,  $i = 1, \dots, N$  and  $c_i$ ,  $i = 1, \dots, N$  are nodes. We have  $2N$  nodes in total. If no



**Fig. 4.** (1): a HMM model, (3): a MEMM model, (4): a CRF model, and (5): the model presented by this paper

assumption is made, there is a link between every pair of nodes. The degree of each node should be  $2N - 1$ . Some assumptions are made for the graphical models shown in Figure 4. Compared with these graphic models, we find that for each  $s_i$ , the degree of our model is three while others are two, which indicates that our model has less assumptions. The *HMM* model is built under two conditional independence assumptions. First, given its previous class, the current class is independent of other classes. Moreover, given its current class, the symbol is independent of other classes and symbols. The *MEMM* model is built under one conditional independence assumption. Given its previous class and the current symbol, the current class is independent of other classes and symbols. The *CRF* model is built under the same two conditional assumptions as the *HMM* model. The model presented in this paper makes one conditional independence assumption. Given the current, the preceding, and the succeeding symbol, the current class is independent of other classes and symbols.

**Comparisons Related to the Three Tasks.** A number of methods for NP chunking [15] [16] [1] [17] [12], word sense disambiguation [18] [19] [5] [20] [21], and semantic role labelling [22] [23] [24] [25] [26] have been developed over the years. We adopted some ideas from these methods. For instance, in NP chunking, we follow Ramshaw's idea [16] of designing three categories for a word in a sentence to determine whether the word is inside a NP chunk, outside a NP chunk, or should start a new NP chunk. However, most methods for this task use HMMs [2] [1], MEMMs [2], and CRFs [3] [10]. In contrast with these methods, we created a new algorithm for these tasks. The core technique of the method is a probabilistic graphical model. This model is fast, uses less memory, and works well for text data.

In the WSD task, in contrast with other WSD methods, the polysemous word is represented by a sequence of context symbols, each symbol is a ordered pair of the lexicon and the POS tag of a word. Each symbol is represented by it's left symbol and right symbol. Moreover, in the semantic argument identification task, most existing methods transform a syntactic tree into a sequence of constituents. Each argument of a verb is represented by a set of constituents. Each constituent is represented by a set of features. These features are extracted based on linguistic knowledge and local knowledge of the tree structure. Finally,

sophisticated classifiers such as support vector machines or maximum entropy modeling classifiers are employed to identify semantic arguments of each verb. In contrast to these methods, our method is based on the idea that if a sentence has a correspondent labeled rooted tree, a semantic argument of a verb in the sentence will be associated with a labeled rooted subtree. Hence, all semantic arguments of a verb in the sentence will be represented by a set of labeled rooted subtrees. For each verb node  $v$ , there exists a path, from which, all roots of the subtrees will be extracted. Obviously, the unique feature, which is a path, represents all semantic arguments of a verb. We find such a path for each verb in a labeled rooted tree associated with a sentence by the probabilistic graphic model.

## 5 Conclusions

We developed an algorithm for identifying three types of semantic patterns: the semantic arguments of a verb, the sense of an ambiguous word, and the noun phrases of a sentence, in texts based on a probabilistic graphical model. By this model, a sequence of optimal classes (or a path) for a sequence of symbols (or nodes) is obtained in a simple way - no need for dynamic programming, fast -  $O(NM)$ , and less memory spaces -  $O(M)$  compared with other existing models such as *HMMs*, *CRFs*, and *MEMMs*. Moreover, because the global maximum probability is achieved by finding assignments that maximize the local probabilities, where the local probabilities take into account neighboring symbol adjacency, and the method provides credit for each correct answer, our performance is comparable or better than other published results on the same data sets.

## References

1. Molina, A., Pla, F., Hammerton, J., Osborne, M., Armstrong, S., Daelemans, W.: Shallow parsing using specialized hmms. *Journal of Machine Learning Research* 2, 595–613 (2002)
2. MaCallum, A., Freitag, D., Pereira, F.: Maximum entropy markov models for information extraction and segmentation. In: *Proceedings of 17th International Conf. on Machine Learning*, pp. 591–598 (2000)
3. Lafferty, J., MaCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of 18th International Conf. on Machine Learning*, pp. 282–289 (2001)
4. Weischedel, R., Palmer, M., Marcus, M., Hovy, E.: Ontonotes release 2.0 with ontonotes db tool v. 0.92 beta and ontoviewer v.0.9 beta (2007), <http://www.bbn.com/NLP/OntoNotes>
5. Leacock, C., Towell, G., Voorhees, E.: Corpus based statistical sense resolution. In: *Proceedings of the Workshop on Human Language Technology*, pp. 260–265 (1993)
6. Bruce, R., Wiebe, J.: Word-sense disambiguation using decomposable models. In: *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 139–146 (1994)

7. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19(2), 313–330 (1994)
8. Tjong, E.F., Sang, K.: Introduction to the CoNLL-2000 Shared Task: Chunking. In: *Proceedings of CoNLL 2000*, pp. 127–132 (2000)
9. Levin, E., Sharifi, M., Ball, J.: Evaluation of utility of lsa for word sense discrimination. In: *Proceedings of HLT-NAACL*, pp. 77–80 (2006)
10. Sha, F., Fereira, F.: Shallow parsing with conditional random fields. In: *Proceedings of HLT-NAACL*, pp. 213–220 (2003)
11. Carreras, X., Márquez, L.: Phrase recognition by filtering and ranking with perceptrons. In: *The International Conference on Recent Advances on Natural Language Processing* (2003)
12. Wu, W.-C., Lee, Y.S., Yang, J.C.: Robust and efficient multiclass svm models for phrase pattern recognition. *Pattern Recognition* 41, 2874–2889 (2008)
13. Veenstra, J., den BoschJ, A.V.: Single-classifier memory-based phrase chunking. In: *Preceedings of CoNLL 2000 and LLL 2000*, pp. 157–159 (2000)
14. Huang, M., Haralick, R.M.: Recognizing Patterns in Texts. River (2010)
15. Church, K.W.: A stochastic parts program and noun phrase parser for unrestricted text. In: *Proceedings of the Second Conference on Applied Natural Language Processing*, pp. 136–143 (1988)
16. Ramshaw, L.A., Marcus, M.P.: Text Chunking Using Transformation-Based Learning. In: *Proceedings of the Third Workshop on Very Large Corpora*, pp. 82–94 (1995)
17. Abney, S., Abney, S.P.: Parsing by chunks. In: *Principle-Based Parsing*, pp. 257–278. Kluwer Academic Publishers (1991)
18. Hearst, M.A.: Noun homograph disambiguation using local context in large text corpora. In: *Proceedings of the Seventh Annual Conference of the UW centre for the New OED and Text Research*, pp. 1–22 (1991)
19. Gale, W., Church, K., Yarowsky, D.: A method for disambiguating word senses in a large corpus. In: *Computers and the Humanities*, pp. 415–439 (1992)
20. Leacock, C., Miller, G.A., Chodorow, M.: Using corpus statistics and wordnet relations for sense identification. *Computational Linguist.* 24, 147–165 (1998)
21. Yarowsky, D.: Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In: *Preceedings of the 32nd Annual Meeting* (1994)
22. Gildea, D., Jurafsky, D.: Automatic labelling of semantic roles. *Computational Linguistics*, 245–288 (2002)
23. Baldewein, U., Erk, K., Padó, S., Prescher, D.: Semantic role labeling with chunk sequences. In: *Proceedings of CoNLL-2004 Shared Task* (2004)
24. Cohn, T., Blunsom, P.: Semantic role labelling with tree conditional random fields. In: *Proceedings of CoNLL 2005 Shared Task* (2005)
25. Hacioglu, K.: A semantic chunking model based on tagging. In: *Proceedings of HLT/NACCL 2004* (2004)
26. Hacioglu, K.: Semantic role labeling using dependency trees. In: *Proceedings of Coling 2004*, Geneva, Switzerland, COLING, August 23–27, pp. 1273–1276 (2004)

# Mining Market Trend from Blog Titles Based on Lexical Semantic Similarity

Fei Wang and Yunfang Wu

Institute of Computational Linguistic, Peking University, Beijing, China  
sxjzwangfei@163.com, wuyf@pku.edu.cn

**Abstract.** Today blog has become an important medium for people to post their ideas and share new information. And the market trend of pricing Up/Down always draws people's attention. In this paper, we make a thorough study on mining market trend from blog titles in the field of housing market and stock market, based on lexical semantic similarity. We focus on the automatic extraction and construction of Chinese Up/Down verb lexicon, by using both Chinese and Chinese-English bilingual semantic similarity. The experimental results show that verb lexicon extraction based on semantic similarity is of great use in the task of mining public opinions on market trend, and that the performance of applying English similar words to Chinese verb lexicon extraction is well compared with using Chinese similar words.

**Keywords:** market trend, verb lexicon extraction, semantic similarity.

## 1 Introduction

The market trend of pricing Up/Down always draws people's attention. With the rapid development and expansion of the Internet, blog has been an important medium for people to post their ideas and share new information, and the titles of blogs make a concentrated presentation of the texts. In this paper, we try to mine market trend from blog titles, taking housing and stock market as two examples. Considering the guiding and recapitulative function of blog titles, we study the titles instead of the main body of the blogs to get the holders' opinion about market trend. The blog titles use a relatively restricted vocabulary, which makes the lexicon extraction proposed in this paper work well. But on the other hand, the expressions in titles can also be diversified and full of figurative meanings, adding the hardship of the task.

We would like to declare that the "market trend" expressed in this paper is different from traditional "opinion mining". In a traditional opinion mining system, opinions are classified as Positive/Negative/Neutral, which imply that the target invokes a positive/negative/neutral feeling. In this paper, the market trend is defined as Up and Down, and we try to mine people's idea (Up/Down) toward near-term direction of the market, instead of the emotional impact that the market trend may have on the public. For instance, the Up trend in housing market may invoke positive feeling for house

holders while negative feeling for house buyers. In sum, the opinion mining in previous work is subjective, while the market trend in our paper is objective.

The following two examples explain our work:

(1) / / / / /

Beijing/housing price/rise/without/end

Housing prices in Beijing will never stop increasing.

(2) / / 空 / /

Three/big/bad news/attack/broad market

Three negative news hit the broad market.

Example (1) tells that the house price trends Up, and (2) tells that the stock market trends Down.

A lexicon-based approach is applied to calculate the market trend of a given title, where verb lexicon plays a pivotal role (e.g., “上 /rise” in Example (1) and “/attack” in Example (2)). Realized the importance of verb lexicon, we focus on: 1) verb lexicon extraction based on distributional semantic similarity; 2) improving Chinese lexicon extraction by introducing English lexical semantic similarity. We conduct experiments using nine methods together, including two baselines, three individual semantic similarity methods and four ensemble methods. The experimental results validate our approach, and provide an easy but a promising way to use cross-lingual knowledge in distributional semantic similarity computation.

The remainder of this paper is organized as follows. In the next section, previous related work is discussed. Section 3 presents the lexicon-based approach and addresses the necessity of verb lexicon extraction. Section 4 describes our verb lexicon extraction methods in detail. Section 5 presents the experimental results and gives an analysis. Finally, Section 6 concludes this paper and proposes future work.

## 2 Related Work

### 2.1 Opinion Mining

Recently there has been extensive research in opinion mining, for which Pang and Lee (2008) give an in-depth survey of literature. Most work concerning opinion mining mainly runs on relatively complete text blocks, and a Positive/Negative tag is then attached to a given document. With the popularity of blogs, opinion mining on blog texts has attracted researchers' attention, such as the work of Chesley et al. (2006), Liu et al. (2010) and Park et al. (2010). There are only a few systems running on titles. Peramunetilleke and Raymond (2002) investigate how news headlines can be used to forecast intraday currency exchange rate movement. This paper focuses on blog titles, which is more difficult than main texts due to the diversified expressions and incomplete syntactic structures. What's more, the designed task of market trend in this paper is different from opinion mining in previous sentiment analysis work.

## 2.2 Similarity-Based Lexicon Extraction

Lexical semantic similarity has been widely applied to lexicon extraction. Previous work concentrates on entity extraction (Pennacchiotti and Pantel, 2009; Pantel et. al., 2009; Chaudhuri et. al., 2009), which is to extract instances of semantic classes (e.g., “*Ziyi Zhang*” and “*Li Gong*” are instances of the class *Actors*). Compared with noun lexicon extraction, less effort has been devoted to verb lexicon extraction. In the work of Shi et al. (2010), the performance of verb extraction is much worse than noun extraction while using semantic similarity. In this paper, we exploit semantic similarity to verb lexicon extraction in the task of mining market trend from blog titles, demonstrating the effectiveness of the application usage of verb extraction.

## 2.3 Cross-Lingual Knowledge in Semantic Similarity Computation

Most work concerning similarity-based lexicon extraction exploits only the knowledge of one language. In the field of synonym extraction, which is a small set of similar words, some studies have used cross-lingual knowledge. Wu and Zhou (2003) extract synonyms with a bilingual English-Chinese corpus, where the feature vector of a word is constructed by the translations and translation probabilities. Lin et al. (2003) identify synonyms among distributional similar words using bilingual dictionaries. Plas and Tiedemann (2006) find synonyms using automatic word alignment of parallel corpora, achieving much higher precision and recall scores than the monolingual syntax-based approach. In the field of Wikipedia-based semantic relatedness computation, which is different from distributional similarity computation in assumptions and methods, Sorg and Cimiano (2008), Potthast et al (2008) provide a new idea of using cross-lingual links/alignment of Wikipedia to map the explicit semantic analysis (ESA) vectors between different languages. In this paper, we extend the cross-lingual property from synonyms to semantic similar words, and provide a more fast way to make use of the full-fledged English resources in Chinese lexicon extraction.

## 3 Lexicon-Based Analysis

In this task, we adopt a simple but widely used lexicon-based approach. Our focus is not the algorithm itself but the impact of the lexicon, that is, we use this simple algorithm to examine the automatically extracted Up/Down verb lexicon.

### 3.1 The Lexicon-Based Algorithm

The algorithm is displayed in Table 1. Obviously, the performance of this approach relies heavily on the Up/Down lexicon, which is just the focus of this paper. The negation window is set to  $q=3$ , according to the empirical analysis.

**Table 1.** Algorithm of computing market trend

---

1. Each title $T^i$ is segmented into a word set $T^i = w_1^i, w_2^i \dots w_n^i$ ;
2. For each $w_j^i (1 \leq j \leq n)$ , look it up in Up/Down lexicon and get $Trend(w_j^i)$ ;
3. Within the window of $q$ words previous to $w_j^i$ , if there is a negation term $w'$ , $Trend(w_j^i) = -Trend(w_j^i)$ ;
4. Calculate the trend value expressed in title $T^i$ : $Trend(T^i) = \sum_{1 \leq j \leq n} Trend(w_j^i)$ ;
5. Determine the trend tag $Tag(T^i)$ : $Tag(T^i) = \begin{cases} Up & \text{if } Trend(T^i) > 0 \\ Down & \text{if } Trend(T^i) < 0 \\ unknown & \text{if } Trend(T^i) = 0 \end{cases}$

---

### 3.2 The Verb Lexicon

In order to decide what kind of word is more important in blog titles, we make a statistics analysis on POS distribution on our data (the collection of our data will be described in Section 5.1). The results show that verb is the most frequently used type of words in titles, which accounts for 26.7% in housing market and 32.8% in stock market. In addition, verbs play a pivotal role in determining the meaning of sentences in Chinese language. So we regard verb as the main information-loader in titles. Thus, in the following section, we focus on the verb lexicon extraction.

We extracted all the verbs in the titles to constitute the candidate verb set, from which we filtered out the directional verbs and dummy verbs (e.g., 以/do, / do), according to the Grammatical Knowledge-base of Contemporary Chinese (Yu et al. 2003), because these verbs cannot play as predicating words in most cases. As a result, we got 834 verbs in our collected data, which constitute the candidate verbs.

Then the task is designed to automatically determine the Up/Down properties of all the candidate verbs. Please note that the verb lexicon extraction in this paper is slightly different from previous literatures. Given two opposite sets of Up and Down seed verbs, our task is to determine the Up/Down properties of the verbs in the candidate verb set, rather than to expand the Up/Down lexicon.

## 4 Verb Lexicon Extraction

### 4.1 Using Tongyici Cilin

We extracted Up/Down seed words from Extended Tongyici Cilin (Cilin for short). Cilin is a manually built Chinese synonym thesaurus, which has been widely used in Chinese language processing. In Cilin there are five levels in the taxonomy structure, in which level 4 corresponds to synset.

In Cilin, synset of */rise* recommends Up while synset of */drop* recommends Down. So we extracted the synset of */rise* as Up words, and the synset of */drop* as Down words. As a result, we got an Up lexicon (UpSeedSet) containing 34 up verbs and a Down lexicon (DownSeedSet) containing 36 down verbs. These verbs will serve as the seed words in the following verb lexicon extraction.

Then, the Up/Down trend of a candidate verb is defined as:

$$Trend(v) = \begin{cases} 1 & \text{if } v \in \text{Up Lexicon} \\ -1 & \text{if } v \in \text{Down Lexicon} \\ 0 & \text{if } v \notin \text{Up / Down Lexicon} \end{cases} \quad (1)$$

Also, we extracted the negation word set from Cilin, containing 25 words.

## 4.2 Using Web Search Engine

In order to verify our approach of similarity-based verb lexicon extraction, we conduct a contrast experiment as one baseline, following the work of Turney and Littman (2003), which has been widely applied in determining the semantic orientation of words in sentiment analysis. Adapting to our particular task, the Up/Down property of a verb is calculated from the strength of its association with a set of words indicating Up trend, minus the strength of its association with a set of words indicating Down trend. Accordingly, the Up/Down property of a verb is computed as:

$$SO(v) = \frac{\sum_{uv \in \text{UpSeedSet}} PMI(uv, v)}{|\text{UpSeedSet}|} - \frac{\sum_{dv \in \text{DownSeedSet}} PMI(dv, v)}{|\text{DownSeedSet}|} \quad (2)$$

Where *UpSeedSet/DownSeedSet* is the Up/Down seed word set extracted from *Cilin*. In this paper, we use *Baidu* search engine<sup>1</sup>, as it is the most popular one in China.

Table 2 lists the top 5 examples in the extracted verb lexicon. That is, the top 5 verbs with the highest values of *SO(v)* in the Up lexicon, and the top 5 verbs with the smallest values of *SO(v)* in the Down lexicon.

**Table 2.** The top 5 examples in the extracted verb lexicon using web search engine

Up	/crowd, 腰/back up, /quiver, /stretch, /recall
Down	开 /levy, /liquidate, /rise slightly, 保 /preserve, /rise

The Up/Down trend of a verb is defined as:

$$Trend(v) = \begin{cases} 1 & \text{if } SO(v) > 0 \\ -1 & \text{if } SO(v) < 0 \\ 0 & \text{if } SO(v) = 0 \end{cases} \quad (3)$$

<sup>1</sup> <http://www.baidu.com/>

### 4.3 Using Chinese Semantic Similarity

Previous studies on Chinese lexical semantic similarity based on large corpora are quite limited, and this paper makes a study on this issue and proves its application usage in verb lexicon extraction.

Given a Chinese thesaurus of semantic similarity, the Up/Down trend of a candidate verb is computed using the following equation:

$$Trend(v) = \frac{\sum_{uv \in UpSeedSet} sim(uv, v)}{|UpSeedSet|} - \frac{\sum_{dv \in DownSeedSet} sim(dv, v)}{|DownSeedSet|} \quad (4)$$

Where  $sim(w_1, w_2)$  denotes the similarity score between  $w_1$  and  $w_2$ .

Lin's method (Lin, 1998) is adopted here to calculate Chinese lexical semantic similarity. The similarity  $sim(w_1, w_2)$  between two words  $w_1$  and  $w_2$  is computed as follows:

$$sim(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \cap T(w_2)} (I(w_1, r, w) + I(w_2, r, w))}{\sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w)} \quad (5)$$

Where  $(w, r, w')$  is a dependency triple consisting of two words  $w, w'$  and the grammatical relationship  $r$  between them;  $I(w, r, w')$  denotes the mutual information;  $T(w)$  is the set of pairs  $(r, w')$  such that  $I(w, r, w')$  is positive.

We use the corpus of Chinese Gigaword, provided by Linguistic Data Consortium (LDC). We take the part of Xinhua News Agency, stamped between the year of 1990 to 2004, containing 471,110K Chinese characters (1.6G) and totally 992,261 documents.

Considering the grammatical relationship  $r$  in Equation (5), two kinds of contexts are adopted: window-based and dependency-based.

**Window-Based Method.** All the texts in the corpus were automatically word-segmented and POS-tagged using the open software *ICTCLAS*<sup>2</sup>. The window context is defined as the left and right 3 words to the target word, along with their position offset to the target word.

**Dependency-Based Method.** All the texts in the corpus were automatically parsed using the Stanford Chinese dependency parser<sup>3</sup> (Chang et al., 2009). Then all the dependency triples were extracted. All of the 45 named grammatical relationships are regarded as the dependency contexts.

When calculating Chinese lexical semantic similarity, we only compute the similarity between two words sharing the same POS tag. In both window-based and dependency-based approaches, we generate up to 200 most similar words for each verb.

Table 3 and Table 4 list the top 5 examples in the extracted verb lexicon, by using window-based method and dependency-based method respectively.

<sup>2</sup> <http://ictclas.org/>

<sup>3</sup> <http://nlp.stanford.edu/software/lex-parser.shtml>

**Table 3.** The top 5 examples in the extracted verb lexicon using window-based Chinese semantic similarity

Up	提 /raise, 展/develop, /rush, /increase, 入/stride
Down	/rise, 衰 /decline, 跌 /drop, /rise, 售/undersell

**Table 4.** The top 5 examples in the extracted verb lexicon using dependency-based Chinese semantic similarity

Up	提 /raise, /lift, /enliven, /climb, /warming
Down	跌 /drop, /touch, /rebound, /frustrate, 动/fluctuate

## 4.4 Using English Semantic Similarity

### 4.4.1 Motivation

English lexical semantic similarity has been extensively studied, and has been applied to many NLP tasks. In this section, we try to apply English semantic similarity to Chinese verb lexicon Extraction.

Some studies on synonym extraction have exploited the cross-lingual knowledge, as discussed in Related Work. The assumptions of these methods as well as our methods are listed as below.

**Assumption-Wu** (Wu and Zhou, 2003): Two words are synonyms if their translations are similar.

**Assumption-Lin** (Lin et al., 2003): Translations of a word from another language are often synonyms of one another.

**Assumption-Plas** (Plas and Tiedemann, 2006): Words sharing translational contexts are semantically related.

**Our Assumption:** If two words are semantically similar in a language, their translations in another language would be also similar with the same similarity score.

We extend the cross-lingual property from synonyms to similar words, where the former is only a small set of the latter. Instead of using translational contexts (Plas and Tiedemann, 2006), we manage to directly use the translational semantic similarity. Considering that lexical semantic similarity computation is time consuming with the computation cost  $O(n^2m)$ , where  $n$  is the number of target words and  $m$  is the number of context features, our approach provides a fast way to make use of the full-fledged English resources and techniques.

### 4.4.2 Methods

Using English similar words to extract Chinese verb lexicon includes the following two phases.

- 1) To automatically translate the seed words and candidate verbs into English;
- 2) To compute the Up/Down trend of each  $tr(v) \in tr\_CandidateVerbs$  based on an English thesaurus of semantic similarity, using the following equation:

$$Trend(tr(v)) = \frac{\sum_{uv \in tr\_UpSeedSet} sim(uv, tr(v))}{|tr\_UpSeedSet|} - \frac{\sum_{dv \in tr\_DownSeedSet} sim(dv, tr(v))}{|tr\_DownSeedSet|} \quad (6)$$

Where  $sim(w_1, w_2)$  denotes the similarity score between  $w_1$  and  $w_2$ ;  $tr(v)$  is the translation of a candidate verb  $v$ ;  $tr\_CandidateVerbs$  are the translations of candidate verbs;  $tr\_UpSeedSet$  and  $tr\_DownSeedSet$  are the translations of Up seed words and Down seed words.

In the first phase, we use Google translator<sup>4</sup> to translate seed words and candidate verbs into English. In the second phase, we use Lin's proximity-based thesaurus of semantic similarity, which can be freely downloaded from DeKang Lin's homepage<sup>5</sup>. For each word, the thesaurus lists up to 200 most similar words and their similarity scores.

Then, the Up/Down trend of a candidate verb  $v$  is defined as that of the translation  $tr(v)$ :  $Trend(v) = Trend(tr(v))$ .

Table 5 lists the top 5 examples in the extracted verb lexicon using English semantic similarity, showing that the predicting results are promising.

**Table 5.** The top 5 examples in the extracted verb lexicon using English semantic similarity

Up	增/add, 增/enlarge, 增/increase, 增/increase, /jump
Down	/collapse, 倒/tumble, 溃/crash, /retreat, 跌/drop

#### 4.5 Using Ensemble Methods

In order to further investigate the complementary property of English and Chinese semantic similarity in verb lexicon extraction, we conduct experiments using ensemble methods, by combining all of the three values computed under the methods of English semantic similarity( $Td_{EN}(v)$ ), window-based Chinese semantic similarity ( $Td_{Win}(v)$ ) and dependency-based Chinese semantic similarity ( $Td_{Dep}(v)$ ).

Firstly, the Up/Down trend values in each method are normalized by dividing it with the max absolute value among all candidate verbs. For instance, the trend values with window-based Chinese semantic similarity method are normalized using the following Equation:

$$Td_{Win}(v) = \frac{Trend_{Win}(v)}{\max\{|Trend_{Win}(v^i)| \mid v^i \in CandidateVerbs\}} \quad (7)$$

We carry out the following four ensemble methods.

<sup>4</sup> <http://translate.google.cn/>

<sup>5</sup> <http://webdocs.cs.ualberta.ca/~lindek/>

**Average.** The new value is the average of all three values:

$$Trend^{Ens}(v) = \frac{Td_{EN}(v) + Td_{Win}(v) + Td_{Dep}(v)}{3} \quad (8)$$

**Max.** The new value is the one with the max absolute value among all three values:

$$Trend^{Ens}(v) = x, |x| = \max\{|Td_{EN}(v)|, |Td_{Win}(v)|, |Td_{Dep}(v)|\} \quad (9)$$

**Min.** The new value is the one with the min absolute value among all three values:

$$Trend^{Ens}(v) = x, |x| = \min\{|Td_{EN}(v)|, |Td_{Win}(v)|, |Td_{Dep}(v)|\} \quad (10)$$

**Majority Voting.** This combination result relies on the Up or Down polarity tags, rather than the absolute values. The polarity tag receiving more votes among three methods is chosen as the final tag.

## 5 Experiments

### 5.1 Data Collection

The data of blog titles was collected from *Sina* website. We manually picked out some blog titles from housing<sup>6</sup> and stock<sup>7</sup>, written from January 1<sup>st</sup> to December 31<sup>th</sup>, 2009. The two authors of this paper manually labeled Up/Down/Unknown tag to each title in a doubly blind manner. The inter-annotator agreement is in a high level with a Kappa value of 0.81. Discarding the titles with Unknown tag, finally we picked out 1,000 titles for housing and 1000 titles for stock respectively, where titles tagged with Up and Down are evenly distributed. All the data was used as test data.

### 5.2 Experimental Results

Table 6 and Table 7 list the experimental results using 9 methods on two datasets, where the two methods of *Cilin* and *Web Search* serve as two baselines.

As is expected, the method using *Cilin*, which is manually complied, gets the highest precision but rather poor recall. But to our surprise, the *Web Search* method, which has been widely used for lexicon extraction in sentiment analysis, gets the worst performance in our task, and is even worse than the seed lexicon from *Cilin* according to F score.

Both the two methods of *English semantic similarity (English-Sim)* and *Window-based Chinese semantic similarity (Win-Chinese)* outperform two baselines substantially in F score, validating the effectiveness of applying similar words to verb lexicon extraction. But the performance of *Dependency-based Chinese semantic similarity (Dep-Chinese)* is quite poor, achieving an even lower F score than *Cilin* baseline in the housing market.

<sup>6</sup> <http://bj.house.sina.com.cn/HouseBlog/>

<sup>7</sup> <http://blog.sina.com.cn/lm/114/111/day.html>

By applying English similar words to Chinese verb lexicon extraction, we harvest competitive results in overall F score comparing with window-based Chinese similar words in both two datasets, validating our assumption of cross-lingual property of lexical semantic similarity.

Among ensemble methods, the *Average* and *Max* methods rival three individual approaches in both two datasets, proving the complementary property of three different semantic similarity scores. *Max* gets the best performance among 9 methods, which outperforms *Cilin* baseline by 13.51% in F score on housing data and 14.24% on stock data.

### 5.3 Discussion

Verb lexicon extraction using lexical semantic similarity improves the performance of market trend mining, especially in recall score. By applying verb extraction, many candidate verbs that do not appear in the seed words in *Cilin* are given an appropriate Up/Down property. There are 70 seed words in *Cilin*, and the number of indicator words is expanded to 309 using *English semantic similarity* method and 457 using *Window-based Chinese semantic similarity*. We guess that the poor performance of *Dependency-based Chinese semantic similarity* lies in the unsatisfactory performance of Chinese dependency parser.

**Table 6.** Experimental results on housing data<sup>8</sup>

Methods	UP(%)			DOWN(%)			OVERALL(%)		
	P	R	F	P	R	F	P	R	F
Cilin	<b>87.39</b>	38.80	53.74	<b>84.80</b>	34.60	49.15	<b>86.15</b>	36.70	51.47
Web Search	55.53	42.20	47.95	56.81	39.20	46.39	56.14	40.70	47.19
English-Sim	74.72	53.20	62.15	69.63	48.60	57.24	72.20	50.90	59.71
Dep-Chinese	60.89	27.40	37.79	51.54	63.40	56.86	54.05	45.40	49.35
Win-Chinese	72.06	45.40	55.71	61.29	<b>68.40</b>	64.65	65.18	56.90	60.76
Average	72.34	54.40	62.10	64.99	67.20	66.08	68.09	60.80	64.24
Max	72.85	<b>55.80</b>	<b>63.19</b>	65.88	67.20	<b>66.53</b>	68.87	<b>61.50</b>	<b>64.98</b>
Min	57.91	32.20	41.39	52.52	64.60	57.94	54.20	48.40	51.14
Voting	72.63	41.40	52.74	64.37	54.20	58.85	67.71	47.80	56.04

<sup>8</sup> P,R,F stand for Precision, Recall and F score.

**Table 7.** Experimental results on stock data

Methods	UP(%)			DOWN(%)			OVERALL(%)		
	P	R	F	P	R	F	P	R	F
Cilin	<b>88.81</b>	23.80	37.54	<b>85.71</b>	32.40	47.02	<b>87.00</b>	28.10	42.48
Web Search	50.28	35.80	41.82	48.19	37.20	41.99	49.19	36.50	41.91
English-Sim	71.28	<b>42.20</b>	<b>53.02</b>	64.80	48.60	55.54	67.66	45.40	54.34
Dep-Chinese	51.49	20.80	29.63	50.56	63.00	56.10	50.79	41.90	45.92
Win-Chinese	72.00	28.80	41.14	55.80	<b>70.20</b>	62.18	59.71	49.50	54.13
Average	66.08	37.80	48.09	57.12	67.40	61.83	60.05	52.60	56.08
Max	67.13	38.40	48.85	57.63	68.00	<b>62.39</b>	60.74	<b>53.20</b>	<b>56.72</b>
Min	63.83	36.00	46.04	55.89	66.40	60.69	58.45	51.20	54.58
Voting	64.61	23.00	33.92	56.40	58.20	57.28	58.50	40.60	47.93

The distributional hypothesis states that words sharing similar meanings tend to appear in similar contexts (Harris, 1968). In a strict view, distributional hypothesis generates words sharing semantic relatedness rather than semantic similarity. Some researchers believe that distributional related words minimize their usage in many applications, and only synonym words are useful and should be identified (e.g., Lin, 2003; Plas and Tiedemann, 2006). However, our experimental results show that semantically related words are of great use in verb lexicon extraction in mining market trend. Words like " /jump" " /bloom" are assigned as Up trend while words like " /gliding" " /collapse" are assigned as Down trend. To our delight, some figurative meanings of words are also given correct Up/Down properties. For instance, " /the beginning of winter", " /cooling down", " /shrink" and " /cutting sb. in two at the waist" are given Down trend orientations. These words obviously are not the synonyms of /drop, but are semantically related with /drop. So, it is the more-general idea of semantic relatedness, rather than semantic similarity, that we need for some NLP applications.

To our delight, applying English similar words into Chinese verb lexicon extraction brings an obvious improvement in performance, which is well compared with *Window-based Chinese semantic similarity* method and far better than *Dependency-based Chinese semantic similarity* method. Considering the noises introduced by Google translator, applying English similar words would get better results. The experimental results confirm our assumption of cross-lingual property of semantic similarity. Considering the cost of lexical semantic similarity computation, our approach provides a fast way to make full use of English knowledge and resources in Chinese semantic similarity computation.

The poor performance using web search engine lies in two reasons. 1) The returned results of the Chinese search engine are not as good as English. 2) The noises

presented in the Chinese web are huger than English, mostly because that Chinese words are not naturally segmented. The experimental results tell us that some well-developed techniques using web search engine in English perhaps would not work well in Chinese.

## 6 Conclusion

In this paper, we make a thorough study on applying semantic similarity to verb lexicon extraction, aiming at the task of mine Up/Down market trend from blog titles. We harvest a great increase of F score by integrating English and Chinese semantic similarity using the *Max ensemble* method. Both the two methods of *English semantic similarity* and *Window-based Chinese semantic similarity* achieve promising results. Introducing English similar words to Chinese verb lexicon extraction brings an obvious improvement, which is well compared with the *Window-based Chinese semantic similarity* method but is far batter in computation cost.

Our experimental results show that verb lexicon extraction based on semantic similarity is of great use for some NLP applications. In future work, we will apply semantic similarity to other application usages. Also, our experimental results confirm the assumption of cross-lingual property of semantic similarity. In future work, we will make full use of English resources in Chinese semantic similarity computation.

We also realize that, the lexicon-based approach is far from enough to recognize the Up/Down trend expressed in blog titles, and there are still a lot of challenges in this task.

**Acknowledgments.** This work was supported by 2009 Chiang Ching-kuo Foundation for International Scholarly Exchange (under Grant No. RG013-D-09).

## References

1. Chang, P., Tseng, H., Jurafsky, D., Christopher, D.: Discriminative reordering with Chinese grammatical relations features. In: Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation, pp. 51–59 (2009)
2. Chaudhuri, S., Ganti, V., Xin, D.: Exploiting web search to generate synonyms for entities. In: Proceedings of WWW 2009, pp. 166–172 (2009)
3. Chesley, P., Vincent, B., Xu, L., Srihari, R.: Using verbs and adjectives to automatically classify blog sentiment. In: Proceedings of AAAI 2006, pp. 27–29 (2006)
4. Peramunetilleke, D., Wong, R.: Currency exchange rate forecasting from news headlines. In: Proceedings of the 13th Australian Database Conference, pp. 129–137 (2002)
5. Harris, Z.: Mathematical Structures of Language. Wiley, New York (1968)
6. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of Coling/ACL 1998, pp. 768–774 (1998)
7. Lin, D., Zhao, S., Qin, L., And Zhou, M.: Identifying synonyms among distributional similar words. In: Proceedings of IJCAI 2003, pp. 1492–1493 (2003)
8. Liu, F., Wang, D., Li, B., Liu, Y.: Improving blog polarity classification via topic analysis and adaptive methods. In: Proceedings of NAACL 2010 (2003)

9. Potthast, M., Stein, B., Anderka, M.: A Wikipedia-based multilingual retrieval model. In: Proceedings of ECIR 2008 (2008)
10. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval (2008)
11. Pantel, P., Crestan, E., Borkovsky, A., Popescu, M., Vyas, V.: Web-scale distributional similarity and entity set expansion. In: Proceedings of EMNLP 2009 (2009)
12. Park, K., Jeong, Y., Myaeng, S.: Detecting experiences from weblogs. In: Proceedings of ACL 2010 (2010)
13. Pennacchiotti, M., Pantel, P.: Entity extension via ensemble semantics. In: Proceedings of EMNLP 2009 (2009)
14. Sorg, P., Cimiano, P.: Cross-lingual information retrieval with explicit semantic analysis. In: Proceedings of CLEF 2008 (2008)
15. Plas, L., Tiedemann, J.: Finding synonyms using automatic word alignment and measures of distributional similarity. In: Proceedings of COLING/ACL 2006, pp. 866–873 (2006)
16. Shi, S., Zhang, H., Yuan, X., Wen, J.: Corpus-based semantic class mining: distributional vs. pattern-based approaches. In: Proceedings of COLING 2010, pp. 993–1001 (2010)
17. Turney, P., Littman, M.: Measuring praise and criticism: inference of semantic orientation from association. ACM Transactions on Information Systems, 315–346 (2003)
18. Wu, H., Zhou, M.: Optimizing synonym extraction using monolingual and bilingual resources. In: Proceedings of the Second International Workshop on Paraphrasing: Paraphrase Acquisition and Application (2003)
19. Yu, S., Zhu, X., Wang, H., Zhang, H., Zhang, Y., Zhu, D., Lu, J., Guo, R.: The Grammatical Knowledge-base of Contemporary Chinese. Tsinghua University Press (2003)

# Ensemble Approach for Cross Language Information Retrieval

Dinesh Mavaluru<sup>\*</sup>, R. Shriram<sup>\*\*</sup>, and W. Aisha Banu<sup>\*\*\*</sup>

School of Computer and Information Sciences,  
B.S. Abdur Rahman University, Chennai, India  
dineshavaluru@gmail.com, {Shriram,aisha}@bsauniv.ac.in

**Abstract.** Cross language information retrieval (CLIR) is a sub field of information retrieval (IR) which deals with retrieval of content from one language (source language) for a search query expressed in another language (target language) in the Web. Cross Language Information Retrieval evolved as a field due to the fact that majority of the content in the web is in English. Hence there is a need for dynamic translation of web content for a query expressed in the native language. The biggest problem is that of ambiguity of the query expressed in the native language. The ambiguity of languages is typically not a problem for human beings who can infer the appropriate word sense or meaning based on context, but search engines cannot usually overcome these limitations. Hence, methods and mechanisms to provide native languages access to information from the web are needed. There is a need, to not only retrieve the relevant results but also, present the content behind the results in a user understandable manner. The research in the domain has so far focused in terms of techniques that make use support vector machines, suffix tree approach, Boolean models, and iterative results clustering. This research work focuses on a methodology of personalized context based cross language information retrieval using ensemble-learning approach. The source language for this research is taken, as English and the target language is Telugu. The methodology has tested for various queries and the results are shown in this work.

**Keywords:** Information Retrieval, Cross Language Information Retrieval, Ontology, and Summarization.

## 1 Introduction and Literature Review

The rapid development of communication technologies has enabled large volume of information to be available on the World Wide Web. However, majority of the content is expressed in a few languages alone. Cross Language Information Retrieval [1] aims to alleviate this. In Cross Language Information Retrieval, different

---

<sup>\*</sup> Junior Research Fellow.

<sup>\*\*</sup> Professor.

<sup>\*\*\*</sup> Associate Professor.

approaches are used to translate the user query and to retrieve the information. One of the common methodologies in CLIR is the use of machine translation (MT), in which query translated automatically into the target language. This is one of the simplest and useful because the query can be translated from the language of the source to another language for search and then the results can be translated back. However, there are some issues with MT system as there is a large possibility of translation errors because of missing information in the term index or ambiguous descriptions [2]. Machine Translation can give quality translations for specific domains only such as those containing specific technical terminology. This is possibly because semantic accuracy suffers when insufficient domain knowledge is incorporated into a translation system. For example, in German-Spanish CLIR, [3] the user was not able to find direct German/Spanish MT. So the user had to use German/English MT, then English/Spanish MT.

The controlled vocabulary approach has been most forceful and effective in the extended run. It is a way to insert an informative layer of semantics between the term entered by the user and the underlying database to better representation of the original intention of the terms of the user query [4]. The descriptors can be added to the thesaurus manually or automatically if the system can learn from previous indexing which terms are likely to be important [5]. The drawback in this approach is that a query must be generated using only words from the thesaurus in which case it may be difficult to search for specific terms that are not included in the thesaurus. This work proposes a new method to search for words that are not in the thesaurus.

In [6] multilingual dictionary based CLIR system each term in the user query is looked up in to the machine-readable bilingual dictionary. Some form of ambiguity resolution or equivalent selection is applied to pick the best translation of that term from the list given by the dictionary. This translation is added to the language semantic mapping of the words in user query.

Nowadays, the semantic web [7] can play an important role in CLIR. Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary. For example, CLIR projects (MuchMore and LIQUID) [8] uses a domain specific ontology that contains the information of the domain and serves as an inter-lingual support for a bilingual thesaurus. Related terms contained in a user query translated into several languages using the term-to-concept links established in the multilingual thesaurus.

In [9], the translation ambiguity is solved by using the name transliteration, single words translation, and collocation translation. The user query is translated fully by automated query translation or by the user-assisted query translation [10]. A study in [11] showed that a new source to obtain query translations is Wikipedia. With Wikipedia, the cross-lingual links of articles can be used to obtain translations. Because of the continuous contribution of users, the wiki is up to date. Wikipedia covers a large number of domain specific terms and named entities also.

In [12], a hybrid approach for query processing Chinese-English is described. The work uses a graph based model for handling ambiguity and resolving out of vocabulary terms. This work uses a graph based model for handling the ambiguity and builds on the approach given in [13] for handling out of vocabulary terms.

For Indian languages, the corpus based and dictionary based query translations have been used for effective translation of user query [14]. In [15], a Telugu Cross

Language Information Retrieval method using a focused crawler and the transcoding processes is described. [16] Uses a cognate identification system for CLIR.

The proposed ensemble approach combines the best characteristics of the existing literature. The query is disambiguated using a lookup in the ontology. If the query is not available in the ontology, the web is searched for matching terms and words relevant to the query term are shown to the users as query suggestions. Thus, the scalability issues are alleviated and the need to constrain the vocabulary for a domain is not needed. The query suggestions overcome the problem of ambiguity. The translation to the source language is started once the query is finalized. The translation is user assisted and named entity issues are overcome. In the search stage, there are two distinct processes. The query and related terms are sent to the search engine via the application-programming interface in both the source language and the target language. This widens the scope of the search and generates a larger subset of results for re-ranking. The role of the re-ranking algorithm is crucial. The re-ranking algorithm uses the query suggestion process, the ontology contents and the contents from the web for generating a clustered result set. The summarization/smoothening process translates the contents back into the target language for search results in the source language. Thus, the entire set of results is shown only in the source language of the users. The work has been specifically developed for Telugu using the grammar rules for the language. The overall approach is called the ensemble approach as different methods are combined in a single process for generating appropriate results.

The specific objectives of this work are to demonstrate a methodology for Cross lingual information retrieval which addresses the issues of ambiguity in query processing, improves the relevancy of the retrieved content and presents the final outcome in a user friendly manner.

This paper is organized into five sections. In the next section, we sketch the overall methodology and the components of the ensemble approach. The operation of the system is described in Section 3. Section 4 describes the implementation and the results, followed by a conclusion and future work in Section 5.

## 2 Ensemble Approach

The ensemble approach consists of four interlinked components for Cross Language Information Retrieval. The approach consists of modules for: a) query preprocessing b) information retrieval from the web c) Re-ranking and d) content presentation. The overall methodology of the ensemble approach is shown in Fig 1.

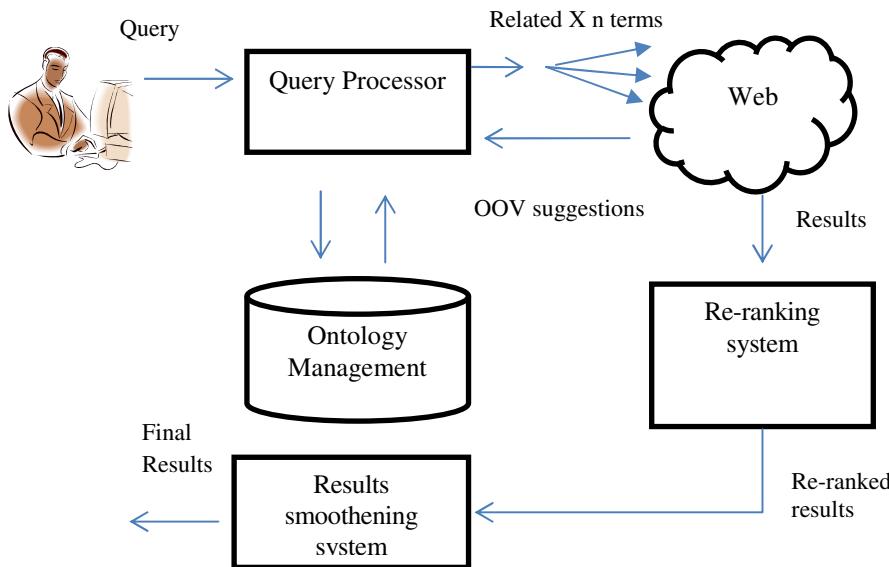
In the preprocessing stage, the query expanded using the ontology and relevant suggestions generated for the user. The suggestion-query interaction is used in the



**Fig. 1.** Overall Process

results retrieval stage, where related- X-n approach is used. The related-X-n approach is a method by which the query and its n related terms are sent to the search engine. The results of search are re-ranked using a query dependency tree based approach. Finally, the content selected by the user from the results is transformed by using a summarization – smoothening approach and the results shown to the users in the target language itself the overall interaction is used in the interaction stage for implicit rule formation that helps refine the order of query suggestion generation and result generation. The various components that implement the Ensemble approach are shown in Fig 2. There are three major components:

- Query Processor
- Results Re-Ranking System
- Results Smoothening System



**Fig. 2.** Ensemble Approach Components

## 2.1 Query Processor

The Query processing system receives the query in the target language (Telugu). The objective of the system is to generate suggestions to resolve ambiguity, handle out of vocabulary (OOV) words that are not present in the ontology and proper nouns. The outcome of the query processing system is to generate a set of related-X- n terms that are closely related to the query in both the source and target language. For this, the ontology design is crucial. For any term, the meanings, related terms, and relationships alone are stored. The structure ‘meaning’ stores the synonyms of the words. The ‘related’ terms of the term are terms that are not exact meanings but

related to the query in some way. The use of related terms is to identify the terms that can be used in the query processing in the related -X- n approach and the post processing of the queries. The relationship term can denote explicit relationships in the content. For example we can indicate that laptop is a mobile device where 'is a' is the relationship term. For book (పుస్తకం), the Telugu meanings are pusthakam (పుస్తకం), grandam (పుస్తకం). But the English word stored in the Ontology is monograph. A word can have multiple English meanings too.

The query entered by the users is now checked for proper noun (named entity). If the query is not a named entity, the query is checked for presence in the ontology. The meanings, related terms, and class relationships are generated as suggestions by the system. The user now can decide to expand the suggestions further by choosing one option from the suggestion in which case the suggested term is further expanded by search in the Ontology. The order of displaying the suggestions is shown below in Fig 3. The meaning, relationship terms, and related terms are expanded in the order and shown to the users.

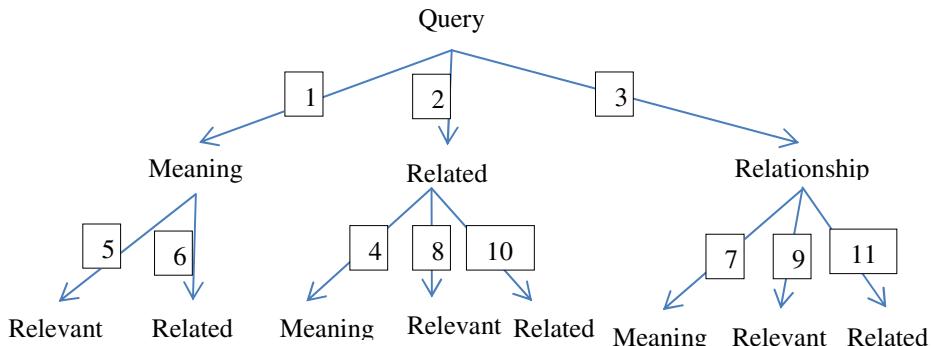


Fig. 3. Query Suggestion Order

In case the query term is not found in the system, the out of vocabulary processing system is called. The out of vocabulary processing system sends the target language query to the web. The snippets are pre-processed and stop words removed. The top 10 most frequently occurring terms are shown to the user as suggestions. As of now, only frequency is taken as the metrics for generating the suggestions for OOV cases. In future, we will refine the system further. There are two stages in the out of vocabulary processing system: a) Preprocessing and b) Dependency tree modeling. In the preprocessing stage a tokenizer and stop word remover are used. The stop words list is a customized set of words, which are commonly present in snippets. Preprocessing reduces the size and number of the input documents considerably, which is essential in any information retrieval system. Preprocessing removes all types of stop words, special characters, extensions, etc., to reduce the processing overhead created by including the stop words into the system's preprocessing framework. The lexical analysis is used to divide a stream of characters into a stream of words. In the dependency tree modeling stage, the contents are initially modeled in the form of  $S_{ij}$ ,

where a single  $S_{11}$  represents the first word of the snippet  $S_i$ . Thus,  $S_{11}$  represents the first term of the first snippet. Now, the query represents the root.

The first term  $S_{11}$  is aligned with reference to the query. If there are any relationship exists (meaning, related, and relationship), they are modeled first (for example term t and the connector relation). If no relationships exist, then a new connector is created (new 1, 2, 3...) and the term is placed in order. Now, for every subsequent term the alignment is done accordingly, namely, if there is an existing relationship in the ontology, then the terms are aligned in line with these relationships. The outcome, thus, is a set of clustered terms. The cluster head is chosen as the term which is most strongly connected. The operation is shown below. The snippets are shown in Fig 4.

<ul style="list-style-type: none"> <li><b><u>మొబైల్ కంప్యూటింగ్ - వికీపీడియా</u></b>  <a href="http://te.wikipedia.org/wiki/మొబైల్_కంప్యూటింగ్">te.wikipedia.org/wiki/మొబైల్_కంప్యూటింగ్</a>  <b>మొబైల్ కంప్యూటింగ్</b> (Mobile computing) అనేది చలనంలో ఉన్నప్పుడు సాంకేతిక పసుపులను వాడటానికి ఒక ప్రక్తికున్న సామర్థ్యాన్ని పర్షించడానికి వాడే సాధారణ పదం, స్థిరముగా ఒక చేటి అమరిక చేసి మాత్రమే వాడునానికి వీలైన సులభంగా ... </li> </ul>
<ul style="list-style-type: none"> <li><b><u>మొబైల్ టిపి - వికీపీడియా</u></b>  <a href="http://te.wikipedia.org/wiki/మొబైల్_టిపి">te.wikipedia.org/wiki/మొబైల్_టిపి</a>  <b>మొబైల్ టిపి</b> లిపిజన అంట చేతిలో ఇమిడ్ పరికరముతో లిపిజన చూడడం. ఆ ... </li> </ul>
<ul style="list-style-type: none"> <li><b><u>మొబైల్ సంబర్ పోర్టబిలిటీ - వికీపీడియా</u></b>  <a href="http://te.wikipedia.org/wiki/మొబైల్_సంబర్_పోర్టబిలిటీ">te.wikipedia.org/wiki/మొబైల్_సంబర్_పోర్టబిలిటీ</a>  <b>మొబైల్ సంబర్ పోర్టబిలిటీ</b> (Mobile Number Portability or MNP) మొబైల్ పోను వాడకందర్కు, ఒక మొబైల్ నెట్వర్కు ఆపరేటర్ నుండి మరొక ఆపరేటర్కు మార్పినపుడు తమ మొబైల్ లిఫోన్ సంబర్ను ఉంచుకోగలిగే సాలబ్యం కల్పిస్తుంది. ... </li> </ul>

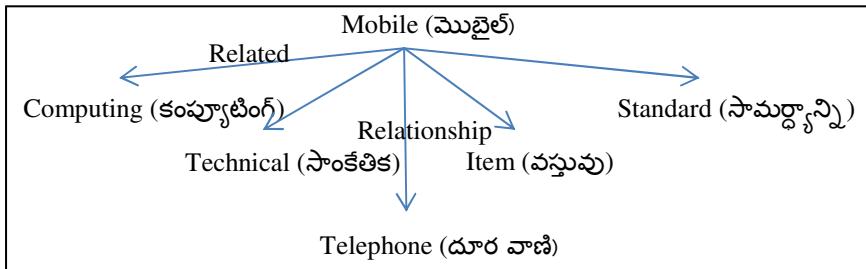
**Fig. 4.** Snippets for the User Query

After stop word removal the snippets are shown in Fig 5 and tree expansion is shown in Fig 6.

<b>R1: మొబైల్ కంప్యూటింగ్</b> (Mobile computing) సాంకేతిక పసుపులను ప్రక్తికున్న సామర్థ్యాన్ని పర్షించడానికి సాధారణ పదం, స్థిరముగా అమరిక చేసి వాడటానికి వీలైన సులభంగా...
<b>R2: మొబైల్ టిపి</b> లిపిజన అంట చేతిలో పరికరముతో లిపిజన చూడడం. ఆ ...
<b>R3: మొబైల్ సంబర్ పోర్టబిలిటీ</b> (Mobile Number Portability or MNP) మొబైల్ పోను, మొబైల్ నెట్వర్కు ఆపరేటర్ మార్పినపుడు మొబైల్ లిఫోన్ సంబర్ను...

**Fig. 5.** The Snippet Contents, After Stop Word Removal

The ontology used for this research is in Telugu language and related to the terms that are available in the domain specified. The size of the ontology is related with 4000 terms. The method is not limited to Telugu domain only it can use for different domains also.



**Fig. 6.** Dependency Tree Expansion for snippet R1

Now, all the snippets are processed in the same way and the suggestions are generated as

- మొబైల్ కంప్యూటింగ్ (Mobile Computing)
- మొబైల్ టెలివిజన్ (Mobile Television)....

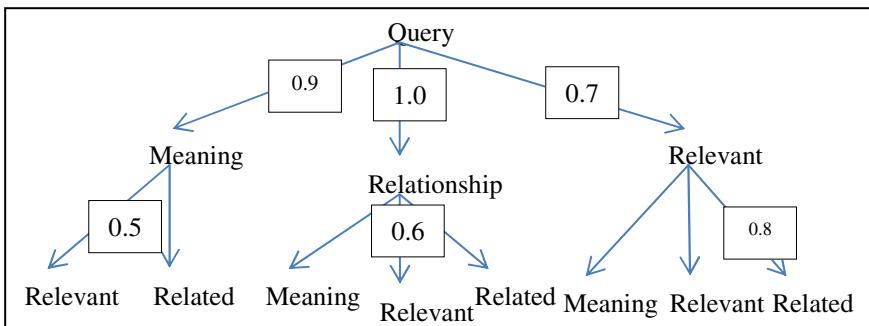
In the order of the Snippet generation. Thus, the first preference is for a connected structure. The next preference is for the order of terms in the order of frequency. The last preference is for the order of the snippets. Once the user stops the query processing and indicates the search for the web, the query and 10 terms in Telugu and English are sent to the search engine. The value of  $n$  is computed in the following way. The meanings in Telugu and English are assembled first. All the meanings in Telugu, their corresponding meanings in English are assembled. If a term has more than 5 meanings, the first 5 meaning terms in English and 5 meaning terms in Telugu are taken. If less than 5 meanings are available in English and more meanings are available in Telugu, additional meanings are taken in Telugu language. For the rest of English terms, the meanings of related terms are taken. If less than 5 meanings are available in Telugu, then the first 5-m related terms are taken and so on. Thus a minimum of 5 Telugu terms and 5 English terms are taken. The 5 is a number we arrived at with experimentation. In future a strategy for arriving at the optimal number based on the term type will be framed.

## 2.2 Re-ranking System

The ontological method models the set of keywords retrieved by the search process as a unified whole, from where the re-ranking of the content can be done using the fuzzy relations between the query term and the ontology. In the Web search, it has observed that the majority of the snippet contents contained the search query. Hence, methods to manipulate the results based on the snippets, must also take into account the linkages of the search term in the context of the snippet, thus needing the ontology. The snippets are assigned a rank based on the inter-term relationships in an organized set of steps. Each term processing is considered as a step in the computation of the information gain, and the consolidated information gain  $tf_{ij}$  is calculated for the entire snippet contents. Here, the notation  $tf_{ij}$  represents a term 't' in the snippet 'f'. The term

‘i’ stands for the snippet value and the term ‘j’ stands for the term in the snippet. Each snippet is randomly chosen from among the search results. The terms visited in each snippet can be written as  $tf_{i1}, tf_{i2}, tf_{i3}, \dots$

For each term in the snippet, the distance vector measure is calculated in terms of the term-relationship frequency where the term relationship frequency is calculated as the measure of the term-relationship value level. Now the term relationship is calculated for the snippet as to how each term is related to the contents of the ontology in the dependency tree order in Fig 7. The position in the parse tree is found. For relationships, the value is 0.9. For meanings it is 1.0. The third position (related) is 0.75. The next positions are each assigned a value of 0.60, 0.55, 0.50, 0.45...etc. till the 10 terms are reached. For all other terms 0.05 is assigned. Anything beyond is not assigned any value, and left off. The sample term frequency calculated for the ontology given in Fig 7.



**Fig. 7.** Term Frequency for the query terms

The similarity of the query results are found next. This is done by comparing the non-stop terms of the snippets. Two snippets are considered to be similar if more than 60% of the terms in the terms match. The value 60% has been arrived after experimentation and in future theoretical basis for the same will be derived. Similar snippets are clustered in the order (meaning, related, relationship; snippet number). The results contain mix of English and Telugu content. For the English results the result smoothening approach is used.

### 2.3 Smoothening Approach

The resultant snippets in English are taken one at a time. The basic unit of the process is to identify the root words of each term in the snippet. First the snippets are delineated in terms of sentences. Sentences are classified into simple and complex based on the structure. A simple sentence is one which follows the subject verb object form. All other sentences are complex sentences. For each sentence the terms are identified into – clauses and stop words. A clause is a verb/adverb/adjective. The stop

words are identified from the sentences. The terms are converted into the root word using porter's stemming algorithm. Now language specific rules are applied to identify the translation heuristics. A single term may exist in different tense and word forms. Hence the query specific information tree sequence is used to disambiguate the sense of the term. Now, morphological rules are applied to get the translation for known grammar forms and terms. Out of Vocabulary terms are treated in the same manner as Proper nouns. Such terms are transliterated automatically. The resultant effect is of an imperfect translation as of now. In future the approach will be improved by the use of concept maps and automated translation systems.

### 3 Operation of Ensemble Approach

A query is a search term given in the local language (target language). The user query (1) is given to the query processor. The search term can be a proper noun or any sentential form. The query processor searches the query in the ontology for the meaning and related terms (2). If the related terms for the query are found in the Ontology, the terms are shown to the user (3). Else, the web is searched for the contents and results get back to the user (4). In case the query entered is a proper noun, the user can qualify the noun with further categories. Now, the outcome of this stage is a set of related terms for the query (5). The user can refine the query further or stop with the query related terms but use a set of related terms to process the query over web (6). In (7, 8) the results are re-ranked using the re-ranking algorithm. The re-ranked results are smoothening using the smoothening system (10). This system suggests that suggestions must focus on categories. Hence, the query expansion is done using the related terms is used. In (11, 12), the query is translated into the target language (English) using the ontology mapping and the language interface. The n-terms for the meaning are identified and corresponding representations in English are framed. For this, (13) the bi-lingual ontology is used to convert the Telugu word to the English word. Thus, all the previous stages mentioned are repeated again until the user is satisfied with the target language's representation. In case, the user wants, they can skip this stage and see the results directly. In case of proper nouns, the query term transliterated directly (15). The outcome of this stage is representation of source query in the target language. The expanded mode relies on the web for generating the suggestions. The operation of this context mode is depicted in Fig 8. The idea here is that, a process by which the content terms are extracted from the web search results to form categories is far preferred by the users than a complex meaning extraction stage. Once the user finalizes the query, the search results are retrieved and shown (4, 16).

For example, if the query is “లక్ష్మన్ (Laxman)” the user can qualify it as “బాట్స్‌మాన్ (batsman) or బంతి కోట్టేవాడు (ball hitter)”. If the query is “దాచ సారధి (captain)”, the user can choose a set of related term as “భారతదేశ దాచ సారధి (India captain)”.

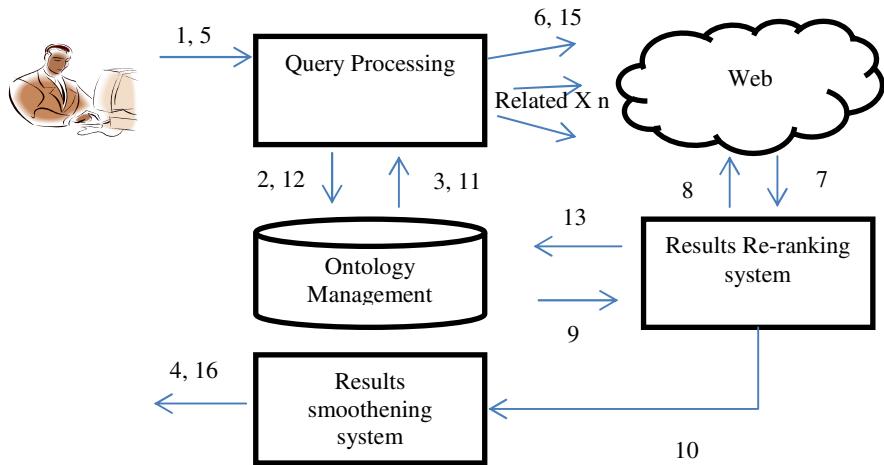


Fig. 8. Framework Operation

## 4 Experiment and Results

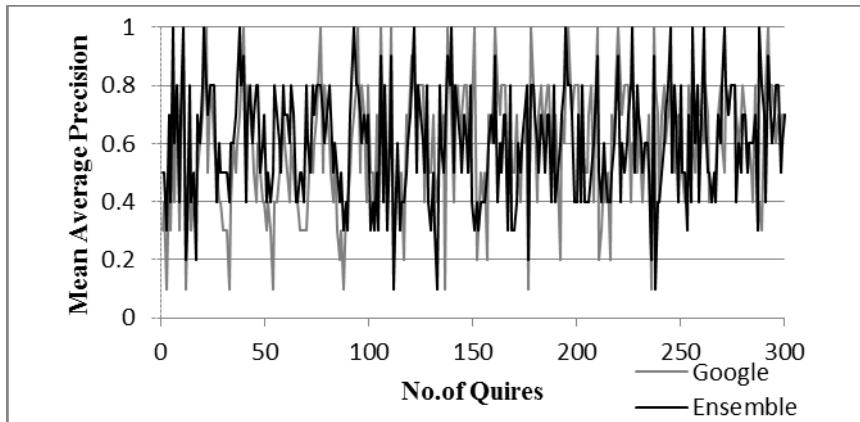
Table 1 shows, the analysis of different query results for the Indian language Telugu because the availability of resources and the query terms are given in English. The overall system has implemented in Java and the Ontology has built for Telugu language. The system has tested for accuracy. The Google search interface was used. The comparison has done between Google results and that of our system. Around 300 queries corresponding to various scenarios were tested.

Table 1. Analysis of Retrieved Results

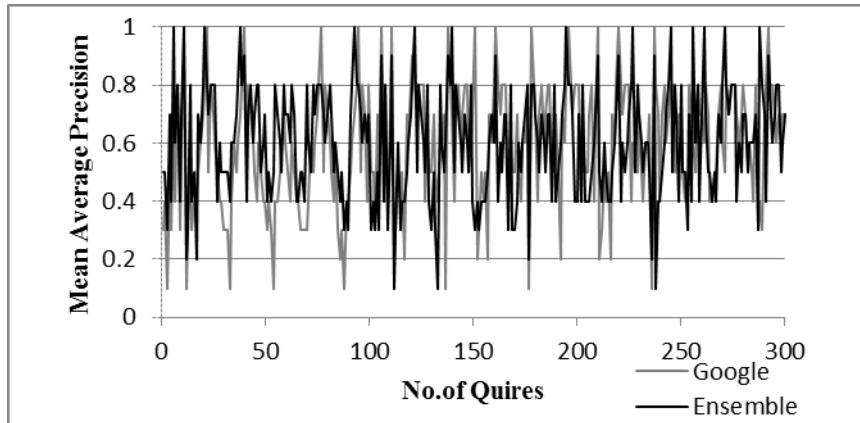
Category	Google	Ensemble Approach
Average Precision	0.56	0.76
Average Recall	0.86	0.84

The key metric used was the Positional accuracy and the mean average precision and mean average recall [17] is used for both systems and shown in the Fig 10 and Fig 11. Here, Y-axis is the precision in fig 9 and recall in fig 10 and in the X-axis, the query terms are shown. The Positional accuracy is computed by manually validating all the positions of each query and plotting the results.

The queries that are tested is available in the ontology and some query terms are not available in ontology. The search engine had an accuracy of 77 % whereas the proposed model had an average range of 80%. This shows that the proposed model is effective.



**Fig. 9.** Precision for Retrieved Results



**Fig. 10.** Recall for Retrieved Results

## 5 Conclusion and Future Work

In this system, Cross Lingual Information retrieval using the ensemble approach. The key aspect of the system are methods for handling query suggestions for out of vocabulary queries, use of a related-X-n approach for query processing and a method for re-ranking the results. The results are encouraging. In future, the experimentation will focus on the performance of the system as per the different query combinations like simple, complex, compound words and proper nouns. In addition, the overhead of the system will be tested. A mechanism for mapping the contents back into the local language is also proposed in this work. This will be expanded further. The work is general and can be applied for other language pairs too. Hence, the emphasis can be on validating this aspect too.

## References

1. Makin., R., Pandey., N., Pingali, P., Varma, V.: Experiments in Cross lingual IR among Indian Languages. In: International Workshop on Cross Language Information Processing (CLIP 2007), Genoa, July 9-10 (2007)
2. Saraswathi, S., Siddhiqaa, M., Kalaimagal, K.: Bilingual Information Retrieval System for English and Tamil. *Journal of Computing* 2(4) (April 2010)
3. Lazarinis, F., Jesus, S., John, V.: Current research issues and trends in non-English Web searching. Springer Science (2009)
4. Vijayanand, K., Seenivasan, R.P.: Named Entity Recognition and Transliteration for Telugu Language. *Language in India, Special Volume: Problems of Parsing in Indian Languages* (May 2011), <http://www.1languageinindia.com>
5. Sieg, A., Mobasher, B., Burke, R.: Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *IEEE Intelligent Informatics Bulletin* 8(1) (November 2007)
6. Carpineto, C., Romano, G., Snidero, M.: Mobile information Retrieval with Search Results Clustering: Prototypes and Evaluation. *Journal of the American Society for Information Science and Technology* 60(5), 877–895 (2009)
7. Huo, Z., Zhao, J., Hu, X.: Web Data Management for Mobile Users, Network and Parallel Computing Workshops. In: IFIP International Conference on NPC Workshops, September 18-21 (2007)
8. Banu, W.A., Kader, P.S.A.: A Hybrid Context Based Approach for Web Information Retrieval. *International Journal of Computer Applications*, article 5 (2010)
9. Nasharuddin, N.A., Abdullah, M.: Cross-lingual Information Retrieval: State-of-the-Art. *Electronic Journal of Computer Science and Information Technology* 2 (2010)
10. Petrelli, D., Levin, S., Beaulieu, M., Sanderson, M.: Which user interaction for cross-language information retrieval? Design issues and reflections. *Journal of the American Society for Information Science and Technology* 57(5), 709–722
11. Damjanovic, V., Gasevic, D., Devedzic, V.: Semiotics for Ontologies and Knowledge Representation. In: Proc. of Wissens Management, pp. 571–574 (2005)
12. Zhou, D., Truran, M., Brailsford, T., Ashman, H.: A Hybrid Technique for English-Chinese Cross Language Information Retrieval. *ACM Transactions on Asian Language Information Processing* (2008)
13. Wang, X., Broder, A., Gabrilovich, E., Josifovski, V., Pang, B.: Cross-language query classification using web search for exogenous knowledge. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining (February 2009)
14. Maeda, A., Kimura, F.: An Approach to Cross-Age and Cross-Cultural Information Access for Digital Humanities. In: Digital Resources for the Humanities and Arts 2008 Conference (DRHA 2008), Cambridge, U.K (September 2008)
15. Khan, A., Naveed, A.M.: Corpus Based Mapping of Urdu Characters for Cell Phones. In: Proceedings of the Conference on Language & Technology (2009)
16. Prasad, P., Varma, V.: Hindi and Telugu to English Cross Language Information Retrieval at CLEF 2006. In: Working Notes for the CLEF 2006 Workshop (Cross Language Adhoc Task), Alicante, Spain, September 20-22 (2006)
17. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (2001)

# Web Image Annotation Using an Effective Term Weighting

Vundavalli Srinivasarao and Vasudeva Varma

Search and Information Extraction Lab  
International Institute of Information Technology  
Gachibowli, Hyderabad-32, India  
[srinivasarao@research.iiit.ac.in](mailto:srinivasarao@research.iiit.ac.in), [vv@iiit.ac.in](mailto:vv@iiit.ac.in)

**Abstract.** The number of images on the World Wide Web has been increasing tremendously. Providing search services for images on the web has been an active research area. Web images are often surrounded by different associated texts like ALT text, surrounding text, image filename, html page title etc. Many popular internet search engines make use of these associated texts while indexing images and give higher importance to the terms present in ALT text. But, a recent study has shown that around half of the images on the web have no ALT text. So, predicting the ALT text of an image in a web page would be of great use in web image retrieval. We propose an approach on top of term co-occurrence approach proposed in the literature to ALT text prediction. Our results show that our approach and the simple term co-occurrence approach produce almost the same results. We analyze both the methods and describe the usage of the methods in different situations. We build an image annotation system on top of our proposed approach and compare the results with the image annotation system built on top of the term co-occurrence approach. Preliminary experiments on a set of 1000 images show that our proposed approach performs well over the simple term co-occurrence approach for web image annotation.

## 1 Introduction

With the advent of digital devices like digital cameras, camera-enabled mobile phones, the number of images on the World Wide Web is growing rapidly. Providing search services for the web images has been difficult. Traditional image retrieval systems assign annotations to each image manually. Although it is a good methodology to retrieve images through text retrieval technologies, it is gradually becoming impossible to annotate images manually one by one due to the huge and rapid growing number of web images.

Automatic Image Annotation has become more and more important and witnessed rapid development in recent years.

Even the search giant, Google, has attempted to recruit its users to tag random images from its index (see figure 1), by re-framing the process as a collaboration between users with those tags matching between users selected as the labels

for the images to improve the quality of Google's image search results<sup>1</sup>. Given that the search giant is using this manual means of image tagging demonstrates the difficulty inherent in the automated image tagging process particularly with regard to scaling those models suggested in the literature to multi-million scale web images and other image libraries.

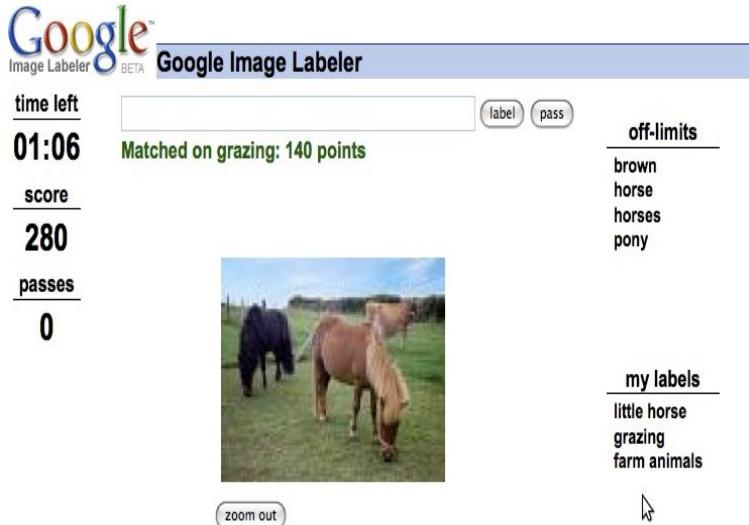


Fig. 1. Google Image Labeler

A common view is that semantics of web images are well correlated with their associated texts. Because of this, many popular search engines offer web image search based only on the associated texts. ALT text is considered the most important of all associated texts. ALT attribute is used to describe the contents of an image file. It's important for several reasons: ALT attribute is designed to be an alternative text description for images. It represents the semantics of an image as it provides useful information to anyone using the browsers that cannot display images or image display disabled.

Many popular internet search engines like Google Image Search<sup>2</sup> make use of these associated texts while indexing the images and give higher importance to the terms present in ALT text. Google states the importance of ALT text in their official blog<sup>3</sup>: *"As the Googlebot does not see the images directly, we generally concentrate on the information provided in the "ALT" attribute."*

<sup>1</sup> <http://images.google.com/imagelabeler> – Google shut it down in September 2011.

<sup>2</sup> <http://images.google.com/>

<sup>3</sup> <http://googlewebmastercentral.blogspot.com/2007/12/using-alt-attributes-smartly.html>

The ALT attribute has been used in numerous research studies of Web image retrieval. It is given the highest weight in [1]. In [2], the authors consider query terms that occur in ALT text and image names to be '*very relevant*' in ranking images in retrieval. Providing '*a text equivalent for every non-text element*' (for example, by means of the ALT attribute) is a checkpoint in the W3C's Web Content Accessibility Guidelines<sup>4</sup>. The authors of [3] also state the importance of using ALT text. However, a recent study[4] has shown that around half of the images on the web have no ALT text at all. The author collected 1579 images from Yahoo!'s random page service and 3888 images from the Google directory. 47.7% and 49.4% of images respectively had ALT text, of which 26.3% and 27.5% were null. It is clear from this study that most of the web images don't contain ALT text.

[5] proposed a term weighting model based on term co-occurrences to predict the terms in ALT text. One advantage of the approach is that it can be extended to any image dataset with associated texts. In this paper, we combine the above term weighting model with the natural language processing applications. We build an image annotation system on top of the above term weighting model and our proposed model.

The reminder of the paper is organized as follows. Section 2 gives an overview of related work. In Section 3, we describe our proposed approach to term weighting using term co-occurrences and natural language processing applications. We describe the dataset, evaluate our system and prove the usefulness of it in Section 4. We summarize the paper and give an account of our future directions in Section 5.

## 2 Related Work

There has been plenty of work done in Automatic Image Annotation. Some of the early approaches[6,7] to image annotation are closely related to image classification. Images are assigned a set of sample descriptions(predefined categories) such as people, landscape, indoor, outdoor, animal. Instead of focusing on the annotation task, they focus more on image processing and feature selection.

Co-occurrence Model[8], Translation Model[9], Latent Dirichlet Allocation Model[10], Cross-Media Relevance Model[11], etc infer the correlations or joint probabilities between images and annotation keywords. Other works like linguistic indexing[12], and Multi-instanced learning[13] try to associate keywords (concepts) with images by learning classifiers. To develop accurate image annotation models, some manually labeled data is required. Most of the approaches mentioned above have been developed and tested almost exclusively on the Corel<sup>5</sup> database. The latter contains 600 CDROMs, each containing about 100 images representing the same topic or concept, e.g., people, landscape, male. Each topic is associated with keywords and these are assumed to also describe the images under this topic.

<sup>4</sup> Web content accessibility guidelines 1.0. Retrieved 26 August, 2005 from <http://www.w3.org/TR/WAI-WEBCONTENT/>

<sup>5</sup> <http://www.corel.com/>

[14] demonstrates some of the disadvantages of data-sets like Corel set for effective automatic image annotation. It is unlikely that models trained on Corel database will perform well on other image collections. Web images differ from general images in that they are associated by plentiful of texts. Various approaches[15,16] have been employed to improve the performance of web image annotation based on associated texts. Our work differs from previous work in that our approach is evaluated on a large number of images and works well for any image dataset with associated texts.

### 3 Term Weighting Model

In this section, we propose a term<sup>6</sup> weighting model which is a combination of the model proposed in [5] and natural language processing applications. We will give a brief overview of the model proposed in [5]. The model computes the term weights based on the term co-occurrences in image associated texts to predict the terms in ALT text. A term is said to be important if it occurs in many associated texts and co-occurs with many other terms present in different associated texts.

For each image, we calculate term weights using the following equation.

$$w(t) = \left( \frac{\sum_i s(t, t_i)}{N} \right) (Imp(t)) \quad (1)$$

$s(t, t_i)$ , the similarity between two terms  $t$  and  $t_i$ , is calculated using Jaccard Similarity as follows:

$$s(t, t_i) = \frac{|S_t \cap S_{t_i}|}{|S_t \cup S_{t_i}|} \quad (2)$$

$S_t \cap S_{t_i}$  is the set of associated texts in which both  $t$  and  $t_i$  occur,  $|S_t \cup S_{t_i}|$  can be calculated as  $|S_t| + |S_{t_i}| - |S_t \cap S_{t_i}|$ , and  $S_t$  is the set of associated texts which contain the term  $t$ .  $N$  is the total number of unique terms in all associated texts.  $Imp(t)$  is the importance of a term which is calculated as follows:

$$Imp(t) = \frac{\sum_i boost(a_i)}{|A|} \quad (3)$$

$boost(a_i)$  is the boost of the associated text  $a_i$  which contains the term  $t$  and  $A$  is the set of associated texts which contain the term  $t$ . The extracted associated texts are assigned a boost based on the heuristic of importance(image caption, HTML title, image filename, anchor text, source page url, surrounding text in that order). Value of boost for each associated text is given based on the importance of the associated text as stated above. Once the weight for each term has been computed, the terms are ranked in descending order based on term weights and top  $k$  terms are selected as terms in ALT text.

---

<sup>6</sup> We use *term* and *word* interchangeably in this paper.

Based on the assumption that noun phrases(NP) are the best lexical category to describe the images[17], we extract noun phrases from the associated texts and consider the terms present in noun phrases as candidate terms for the ALT text. We then considered the terms present in both noun phrases and verb phrases as candidate terms for ALT text. Next, we combined the above approaches with the term co-occurrence approach where the terms in the extracted noun phrases and verb phrases are given more importance. We use OpenNLP tool<sup>7</sup> to extract noun phrases and verb phrases.

We describe the evaluation procedure and present the results of the approaches in the following section.

## 4 Evaluation

In this section, we present the evaluation procedure of our approach. We briefly describe the data collection and preprocessing steps, present the evaluation procedure and finally results are discussed.

### 4.1 Data Collection and Preprocessing

A crawler is used to collect images from many websites. Images like banners, icons, navigation buttons etc, which are not so useful are not considered. The web documents are preprocessed to extract associated texts so that the images can be indexed and retrieved with these text features. The associated texts we considered are extracted from ALT attribute, HTML page title, image filename, source page url, anchor text, image caption and surrounding text.

We used Guardian<sup>8</sup>, Telegraph<sup>9</sup> and Reuters<sup>10</sup> as the source urls and collected a total of 200000 images which have ALT text. We selected these news websites because the ALT text provided in them is accurate and is very useful for evaluation. The pages in which the images are present, cover a wide range of topics including technology, sports, national and international news, etc. Stopwords are removed and stemming is used to filter the duplicate words from the extracted textual information.

We compare our results with the term co-occurrence approach proposed in [5].

### 4.2 Evaluation Procedure

In order to evaluate the effectiveness of our method, we compare the predicted terms produced by our approach against the terms extracted from ALT attribute of an image in the corresponding web page.

---

<sup>7</sup> <http://incubator.apache.org/opennlp/>

<sup>8</sup> <http://www.guardian.co.uk/>

<sup>9</sup> <http://www.telegraph.co.uk/>

<sup>10</sup> <http://www.reuters.com/>

We present results using the top 5, 10, and 15 words. We adopt the recall, precision and F-measures to evaluate the performance in our experiments. If  $P_t$  is the set of terms predicted by our approach and  $A_t$  is the set of terms in ALT text, then in our task, we calculate precision, recall and F-measure as follows:

$$precision = \frac{\text{Number of common terms between } P_t \text{ and } A_t}{\text{Total number of terms in } P_t} \quad (4)$$

$$recall = \frac{\text{Number of common terms between } P_t \text{ and } A_t}{\text{Total number of terms in } A_t} \quad (5)$$

$$F - Measure = \frac{2 * precision * recall}{precision + recall} \quad (6)$$

### 4.3 Analysis

The approach, NP, considers the terms in noun phrases as candidate terms for ALT text. Similarly, NP + VP, considers the terms in both noun phrases and verb phrases as candidate terms for ALT text.

NP + Term co-occurrence is our proposed term co-occurrence approach in which we give more boost to the terms in noun phrases. Similarly, NP+VP+Term co-occurrence is our proposed term co-occurrence approach in which we give more weights to the terms present in noun phrases and verb phrases.

As we can observe from the results in tables 1 to 3, both term co-occurrence approach and term co-occurrence approach combined with NP+VP give better results compared to other approaches and they give almost the same results.

**Table 1.** Comparison of approaches for top 5 predicted terms for 200000 images

Approach	Precision@5	Recall@5	F-Measure@5
Term co-occurrence	53.19	41.95	<b>46.91</b>
NP	40.78	33.59	36.83
NP + VP	41.67	34.78	37.91
NP + Term co-occurrence	48.49	39.24	43.38
NP + VP + Term co-occurrence	49.79	41.46	45.24

For ex: Consider the figures 2 to 5.

Figure 2 is the image of a doll bearing the faces of Russian leader Vladimir Putin and Dmitry Medvedev. The ALT text of the image is “*Dmitry Medvedev: Vladimir Putin is more popular than me*”. The term ‘popular’ is predicted by the term co-occurrence approach where as it is not predicted by any of other approaches.

Figure 3 is the image of Jose Mourinho. The ALT text of the image is “*Jose Mourinho handed two-match ban for Super Cup eye poke*”. The term ‘poke’ is predicted by the term co-occurrence approach where as it is not predicted by any of other approaches.

**Table 2.** Comparison of approaches for top 10 predicted terms for 200000 images

Approach	Precision@10	Recall@10	F-Measure@10
Term co-occurrence	44.11	60.87	51.15
NP	36.87	53.84	43.76
NP + VP	38.13	55.81	45.30
NP + Term co-occurrence	39.07	56.14	46.07
NP + VP + Term co-occurrence	43.60	61.86	51.15

**Table 3.** Comparison of approaches for top 15 predicted terms for 200000 images

Approach	Precision@15	Recall@15	F-Measure@15
Term co-occurrence	37.96	72.98	<b>49.94</b>
NP	32.14	61.42	42.19
NP + VP	32.73	65.87	43.73
NP + Term co-occurrence	31.79	62.41	42.12
NP + VP + Term co-occurrence	37.35	69.25	48.52

Figure 4 is the image of Nicolas Sarkozy and Angela Merkel in a meeting. The terms Angela and Merkel are very relevant to the image. But they are not predicted by term co-occurrence approach where as they are predicted by term co-occurrence approach combined with NP+VP.

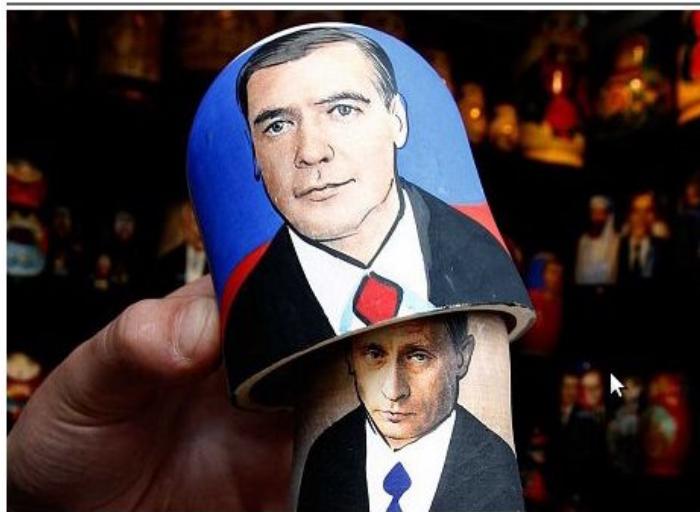
Figure 5 is the image of Casey Stoner at Italian MotoGP. The term MotoGP is very relevant to the image. But it is not predicted by the term co-occurrence approach, where as it is predicted by term co-occurrence approach combined with NP+VP.

There are a few such cases where we missed such important entities from the predicted terms with the term co-occurrence approach. And also some of the terms in the alternate text are not predicted by approaches other than the term co-occurrence approach.

We build an image annotation system on top of the term co-occurrence approach and compared it with the image annotation system built on top of the term co-occurrence approach combined with NP+VP. For the evaluation of image annotation system, we select a subset of 1000 images from the original dataset. Human annotators are chosen to manually assign tags to the images.

Given an image, the following points are taken into account while annotating the image.

- The subject who annotates the image, assigns the keywords which he/she thinks are relevant to the image, without taking a look at the web page which contains the image.
- Once he/she assigns the keywords to the image, he/she visits the web page which contains the image and refines the annotations using the terms extracted from the associated text.



**Fig. 2.** A doll bearing the faces of Russian leader Vladimir Putin and Dmitry Medvedev



**Fig. 3.** Image of Jose Mourinho

As we can see from the results, tables 4 to 6, of the image annotation systems, term co-occurrence+NP+VP gives better results over simple term co-occurrence. For the task of predicting terms in ALT text, both simple term co-occurrence and term co-occurrence + NP + VP work well. However, we prefer simple term co-



**Fig. 4.** Image of Nicolas Sarkozy and Angela Merkel



**Fig. 5.** Image of Casey Stoner at Italian MotoGP

occurrence approach over term co-occurrence+NP+VP as the former is language independent.

**Table 4.** Comparison of annotation systems for top 5 annotations

Approach	Precision@5	Recall@5	F-Measure@5
Term co-occurrence	53.97	50.26	52.04
NP + VP + Term co-occurrence	55.83	52.92	54.34

**Table 5.** Comparison of annotation systems for top 10 annotations

Approach	Precision@10	Recall@10	F-Measure@10
Term co-occurrence	33.95	64.26	44.43
NP + VP + Term co-occurrence	36.04	67.76	47.05

**Table 6.** Comparison of annotation systems for top 15 annotations

Approach	Precision@15	Recall@15	F-Measure@15
Term co-occurrence	25.27	70.44	37.20
NP + VP + Term co-occurrence	26.80	75.14	39.51

## 5 Conclusions and Future Work

In this chapter, we presented a term weighting approach that makes use of term co-occurrences in associated texts and predicts terms occurring in ALT text of an image. We compared the performance of our approach against a few baseline approaches which use term frequency, document frequency, terms in noun phrases and terms in verb phrases respectively. Experiments on a large number of images showed that our model is able to achieve a good performance for the prediction task. We built image annotation systems on top of the above approaches and found out that the term co-occurrence approach in combination with noun phrases and verb phrases performs better than the term co-occurrence approach. The simple term co-occurrence approach for the prediction of terms in alt text is language independent and is preferable over term co-occurrence combined with noun phrases and verb phrases even though both of them produce almost same results. However, the later performs well for the task of web image annotation.

For web image annotation task, we would like to experiment on more number of images and come to a conclusion that the term co-occurrence approach combined with NP+VP works better than the simple term co-occurrence approach for the annotation task.

## References

1. Cascia, M.L., Sethi, S., Sclaroff, S.: Combining textual and visual cues for content-based image retrieval on the world wide web. In: IEEE Workshop on Content-based Access of Image and Video Libraries, pp. 24–28 (1998)
2. Mukherjea, S., Hirata, K., Hara, Y.: Amore: A world wide web image retrieval engine. *World Wide Web* 2, 115–132 (1999)
3. Petrie, H., Harrison, C., Dev, S.: Describing images on the web: a survey of current practice and prospects for the future. In: Proceedings of Human Computer Interaction International, HCII 2005 (2005)
4. Craven, T.C.: Some features of alt texts associated with images in web pages. *Information Research* 11 (2006)
5. Srinivasarao, V., Pingali, P., Varma, V.: Effective Term Weighting in Alt Text Prediction for Web Image Retrieval. In: Du, X., Fan, W., Wang, J., Peng, Z., Sharaf, M.A. (eds.) APWeb 2011. LNCS, vol. 6612, pp. 237–244. Springer, Heidelberg (2011)
6. Vailaya, A., Figueiredo, M.A.T., Jain, A.K., Zhang, H.J.: Image classification for content-based indexing. *IEEE Transactions on Image Processing* 10, 117–130 (2001)
7. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1349–1380 (2000)
8. Hironobu, Y.M., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. Boltzmann machines, *Neural Networks* 4 (1999)
9. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
10. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 127–134 (2003)
11. Jeon, J., Lavrenko, V., Manmatha, R., Callan, J., Cormack, G., Clarke, C., Hawking, D., Smeaton, A.: Automatic image annotation and retrieval using cross-media relevance models. *SIGIR Forum*, 119–126 (2003)
12. Jia, L., Wang, Z.J.: Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1075–1088 (2003)
13. Yang, C., Dong, M.: Region-based image annotation using asymmetrical support vector machine-based multi-instance learning. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2006)
14. Tang, J., Lewis, P.: A study of quality issues for image auto-annotation with the corel data-set. *IEEE Transactions on Circuits and Systems for Video Technology* 1, 384–389 (2007)
15. Rui, X., Li, M., Li, Z., Ma, W.Y., Yu, N.: Bipartite graph reinforcement model for web image annotation. *ACM Multimedia*, 585–594 (2007)
16. Shen, H.T., Ooi, B.C., Tan, K.L.: Giving meaning to www images. *ACM Multimedia*, 39–47 (2000)
17. Kuo, C.H., Chou, T.C., Tsao, N.L., Lan, Y.H.: Canfind: A semantic image indexing and retrieval system. In: ISCAS (3), pp. 644–647 (2003)

# Metaphone-pt\_BR: The Phonetic Importance on Search and Correction of Textual Information

Carlos C. Jordão<sup>1</sup> and João Luís G. Rosa<sup>2</sup>

<sup>1</sup> São Carlos City Hall

Department of Information Technology

São Carlos, SP, Brazil

[carlosjordao@gmail.com](mailto:carlosjordao@gmail.com)

<sup>2</sup> University of São Paulo

Computer Science Department

São Carlos, SP, Brazil

[joaoluis@icmc.usp.br](mailto:joaoluis@icmc.usp.br)

**Abstract.** The increasing automation in the communication among systems produces a volume of information beyond human administrative capacity to deal with on time. Mechanisms to find out the inconsistent information and facilitate the decision-making are required. The use of a phonetic algorithm (Metaphone) adapted to Brazilian Portuguese proved to be a valuable tool in searching for name and address fields for automatic decisions, increasing substantially the performance regular database queries could obtain in information retrieval.

## 1 Introduction

Today, each sector of society is constantly searching for improvements to its registration forms, in order to make them increasingly accurate. One of the tools used to minimize problems of inconsistency is the use of closed form questions, which allows the user to choose only one of a predefined set of options, such as “select box” and “radio buttons” used in HTML forms. This mechanism makes indexing and information exchange among systems easier.

However, there are fields and systems that need to work with textual data, such as names and addresses, regardless of the reason. This makes the cross-checking of information among different systems difficult, because the most common solutions to determine matching between two different registers are not suitable for these cases. The command SQL *like*, for example, cannot guarantee a good match of words, except for very simple variations. Even so, names and addresses have complex variations of spelling without being semantically different. So, manual programming becomes unfeasible, because of the phonetic variation that may occur, which needs to be taken into account when comparing two words.

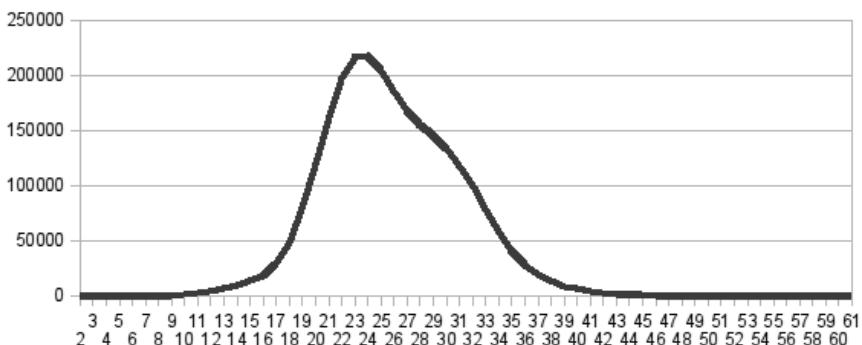
## 2 Objectives

Since simple methods of textual comparison are not efficient, it is essential to study other mechanisms that could deal with the challenge of the phonetic

variation. Algorithms of this category do at least one of the two options: create an index of similarity between two words or supply a simplified representation of the word.

Those algorithms that compare words, such as Levenshtein [3], demand a lot of processing because it is necessary to completely scan the database [1], comparing a word with every other word stored, for example, to find which words are closer to the one given for comparison. Thus, it is preferable to use simplified representation of words once the scan can be done at a low cost, for example, by comparing the simplified phonetic representation of the words. The individual phonetic conversion is normally a very simple process, so that algorithms such as Metaphone [5] and Soundex [4] do not need more than one *loop* to scan the word in order to create its representation.

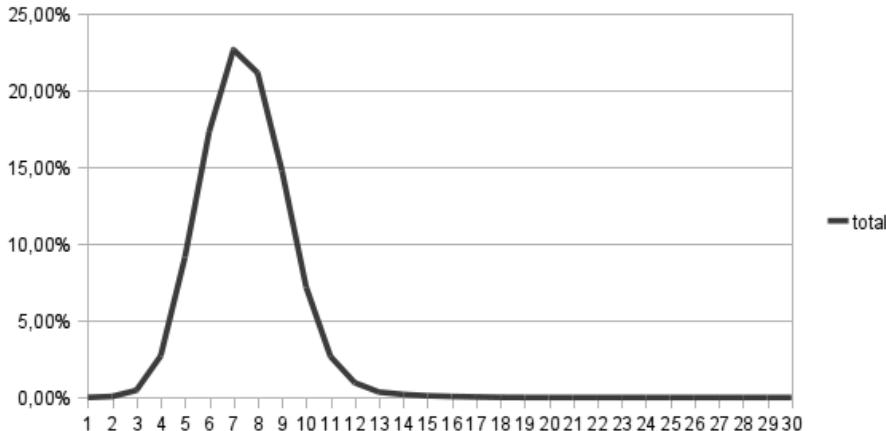
A sample of 2,591,562 proper names from the database of beneficiaries of Brazilian government's social programs, available in its website, resulted in 8,799,513 words and 226,686 exclusive words, which represents an average of 3.39 words per name. Those names were taken in September 2008 with the purpose of building a real base to evaluate the impact of phonetic algorithms for searching names. Figures 1 and 2 show the name length distribution and the word length distribution, respectively, in the sample.



**Fig. 1.** Distribution of amount of names (y-axis) by word length (number of characters) (x-axis)

Considering that the complexity of the algorithm Levenshtein is  $\mathcal{O}(n * m)$ , where  $n$  and  $m$  are the lengths of the words to be compared, it is possible to calculate the necessary effort to scan through this mechanism by the average length of the stored words. In this case, other types of phonetic algorithms are very helpful.

Phonetic algorithms seek to build simplified representation of words, which can be seen as indexes for a database. Their objective is to find words that two people consider as equal or equivalent even if they were spelled differently due to the fact of phonetic context, by allowing these words to be clustered in several



**Fig. 2.** Distribution of word percentage found (y-axis) by word size (number of characters) (x-axis)

clusters, according to their representation. Each algorithm, such as Soundex, Metaphone and Double Metaphone [6], produces different clustering results, in order to find a solution to several situations proposed.

In any case, each algorithm processes the word individually, so that its complexity for the worst case is of order  $\mathcal{O}(n)$ . This allows the representation to be stored in the database in advance, resulting in an index, which improves substantially the search time, minimizing the total computational effort necessary to come up with a smaller set of word records close to the one searched. Thus, the application of a second algorithm of comparison, even Levenshtein, becomes much more feasible. Sanae [8] shows that the best results come from hybrid methods, normally by using a phonetic algorithm with any other type.

Soundex, created by Rober C. Russell and Margaert K. Odell, patented in 1918, is the most famous phonetic algorithm, which inspired many other variations, with adaptation to foreign languages (it was created based on English phonetic rules). It was used for a retrospective analysis of the US censuses from 1880 to 1920 by the United States Census Bureau. Metaphone was created in 1990 by Lawrence Philips [5] as an alternative to resolve the deficiencies of Soundex. Later, the author released another version, called Double Metaphone, which returns two sequences of representations. The first one is called primary, the second, secondary. The primary is closer to the first Metaphone and to the English phonetic rules. The secondary works with a larger set of phonetic rules, such as Chinese and German.

But this alternative is not able to cluster efficiently words from the same language. This paper shows that the Metaphone version for Brazilian Portuguese produces better clustering results when used with the words of the main language than in the multilingual variant. The difference is the search for a method to compare the qualities of two algorithms, so that it is possible to measure the impact of the subsequent changes of the algorithm or variations.

### 3 A Brazilian Metaphone

The understanding of phonetic algorithms as searching tools and the intention of finding a way of searching names and addresses emphasized the need for implementing an algorithm for Portuguese phonetic rules during the project RE-DECA<sup>1</sup>, which is a software aimed to help entities that take care of children and adolescents, exchanging information among themselves. Most of these entities keep a very poor register, consisting mainly of textual information with no possibility of associating one with each other automatically. Therefore, in order to help them to work as a whole, it was important to regulate the mechanism of registration and guarantee the non-existence of duplicated names in the database. This task soon proved to be a challenge. Since there are several different types of documents, each entity uses the one that best suits it. Also, not all of them have all kinds of documents, for many reasons, which makes the indexing of a name to a particular document not reliable. So that, it is necessary to check the entries by the name, in order to avoid duplicity in the database.

So, the Brazilian Metaphone was created to fill this need, since the existing solutions were not sufficient. Other authors have also chosen a similar approach [10] to use in their native languages, once they have found the same difficulties when applying algorithms to their daily activities. Since such algorithms behave as clustering algorithms, a poor classification, i.e. similar words in different clusters, would yield a bad search result [7,9]. Then, since this approach produced positive results, it was expected that it would be equally useful when applied to Portuguese language.

Metaphone has to be understood as an algorithm that excerpts the phonetic information from the consonants of the words. That is, when the vowels are taken out from the word, it is still possible for a person to recognize the essence of the original word in most of western languages. Consequently, the simplification method reduces to the maximum the number of characters that the final representation may have.

It is possible to correlate most phonemes used in the original Metaphone by following the Portuguese spelling rules with the addition of three new symbols for sounds not present in English pronunciation. For the representation of the rules on table 1, a mnemonic notation was used since many rules depend on the analyses of up to three prior or subsequent rules for a final decision of which phoneme will be used.

It is important to analyze how the choice of rules affects the word clustering, when converting phonetic rules. In Portuguese, for example, the consonant “L” sounds as the vowel “U” when preceded by any vowel. So, words like MAL (bad = not well) and MAU (bad = not good) have the same sound. Of 226,686 words, there are 33,682 that fit in this rule, which affect 1,364,712 (52,66%) of the analyzed names. In this case, it is better to consider the vowel as a consonant too, or unnecessary distinctions between words could be created.

---

<sup>1</sup> [http://www.softwarepublico.gov.br/ver-comunidade?community\\_id=18016032](http://www.softwarepublico.gov.br/ver-comunidade?community_id=18016032)

**Table 1.** Symbols used on rule phonetic mapping table (Table 2) of Brazilian Metaphone algorithm

<i>symbol</i>	<i>meaning</i>
^	word beginning.
\$	word end.
[]	any characters inside the brackets.
v	any vowel (lower case letter 'v').
c	any consonant (lower case letter 'c').
.	any letter.
0	empty. It means that the character found was ignored (not mapped). capital letters specific vowel or consonant found.

The sounds not present in English were mapped to the symbols “1”, “2”, “3”, to represent the sounds of the (combinations of) letters “LH”, “R” (voiced uvular fricative) and “NH”, respectively. As a result of this work, tables 3 and 4 illustrate two word clustering that share the same phonetic representations, showing the similarity between them.

Finally, the main challenge in a language-dependent phonetic algorithm is to work with foreign names and its adaptation to the language usage, by adjusting the spelling of a foreign name to a local spelling. Therefore, being limited to lexical rules is also a problem because the algorithm would be helpful only for the dictionary words, but not for names and addresses, which suffer variations due to surnames and foreignness, which have consonant clusters unusual for the local language. This reinforces how far this complexity is from being solved by simple approach of comparison between name and database.

Thus, it is crucial to be able to exchange information among several systems through registration cross-checking of names and addresses in environments where the accurate correlation through numerical keys is not possible, as well as the correlation of information through address, for georeferencing.

The implementation of the algorithm can be found at SourceForge<sup>2</sup> under the GPL license.

### 3.1 Comparison between the Brazilian Metaphone and the Double Metaphone

The first experimental results with the algorithm were quite satisfactory, but it is necessary to measure how accurate it could be, having the secondary string of Double Metaphone as reference.

Since each phonetic algorithm has its own rules to create a phonetic representation of a word, it is not possible to directly compare the rules or the output of each word individually.

<sup>2</sup> <http://sourceforge.net/projects/metaphoneptbr/>

**Table 2.** Rules for the Brazilian metaphone algorithm

<i>Letters</i>	<i>Phonetic representation (comments)</i>
$\hat{v}$	v (repeats the vowel.)
B	B
$C[AOU]$	K
Cc	K
C\$	K
$C[EI]$	S
CHR	K
CH	X (this rule applies if the most specific does not match first.)
C	S
D	D
F	F
$G[AOU]$	G
$G[EI]$	J
$GH[EI]$	J
GHc	G
$\hat{H}v$	v
H	0
J	J
K	K
LH	1 (new sound)
$\hat{L}$	L
Lv	L
vLc	c (keep last consonant)
M	M
N\$	M
NH	3
P	P
PH	F
Q	Kv
$\hat{R}$	2
R\$	2
RR	2
vRv	R
.Rc	R
cRv	R
SS	S
SH	X
$SC[EI]$	S
$SC[AOU]$	SK
SCH	X
Sc	S
S	S (again, specific rules come first.)
T	T
TH	T
V	V
Wv	V
Wc	0
$\hat{E}Xv$	Z
$[MV]EX$	X
.EX[EI]	X
.EX[AOU]	KS
EX[PTC]	S
EX.	KS
$[vCKGLRX][AIOU]X$	X
$[DFMNPQSTVZ][AIOU]X$	KS
X	X
Y	I
Z\$	S
Z	Z

**Table 3.** Words sharing the phonetic key 2BK

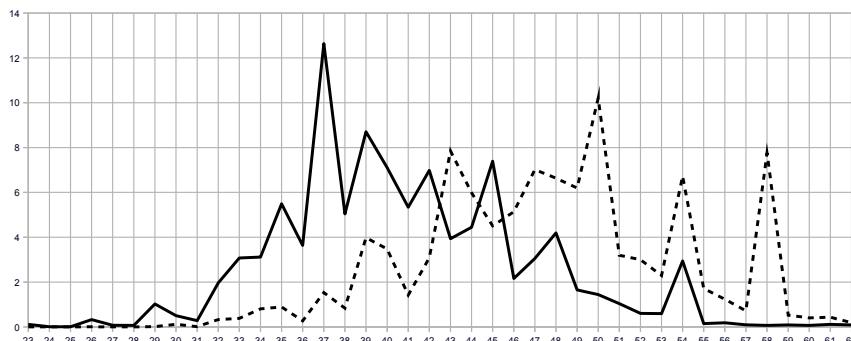
ROBECA	REBCA	REBELC	REBOUCO	REBOLCA
RHEBECA	REBBEKA	RBECA	RABACO	REBOUCA
REBHECA	RABEKHA	REBEKAH	RABECO	REBEQUE
REBEKKA	REBBECA	REBELK	RHEBEKA	REBEKA
REBEECA	RUBICA	REBEK	REBEKHA	REBECA
REBECAH	REBOLCO	RABECA	REBEQUI	

**Table 4.** Words sharing the phonetic key JLRD

GELIARD	GILIARDO	GILYARD	GELIARDE	GILIARDY	JILIARD
GILARD	GILLARDE	JILLIARD	GILEARDE	GILLIARD	JOLYARDE
GILEARDY	GILLIARDE	JULIARD	GILIARD	GILLIARDI	JULIARDE
GILIARDE	GILLIARDY	JULIARDI	GILIARDI	GILLYARD	JULIARDO

On the other hand, the words grouped by each Metaphone word are expected to be as homogeneous as possible, even if the quantity and the length of each one vary according to the algorithm used.

Therefore, to measure this uniformity, an algorithm of similarity was applied between the words of each cluster, calculating homogeneity by the average of the figures within that cluster.



**Fig. 3.** Comparison between Double Metaphone (filled line) and Brazilian Metaphone (dotted line). The x-axis is the homogeneity percentage and the y-axis is the cluster counting percentage.

This process was made for the following algorithms of similarity: Levenshtein, Q-gram [12], Jaro [2], Jaro-Winkler [11], Similarity (implementation of trigraphs for PostgreSQL). Each of these algorithms has its own particularities, so a comparison using several of them would minimize those differences, as each of them returns a number between 0 and 1 representing the percentage of similarity. After being applied to all, the average for the cluster was calculated from the averages obtained with each algorithm.

The final result of the amount of clusters per similarity index is shown in Figure 3. The weighted average of each distribution is 40.8% and 47.9% for Double Metaphone and Brazilian Metaphone, respectively. That demonstrates that the expectation of the Brazilian algorithm resulting in a more homogeneous cluster is around 17% better than the Double Metaphone. Naturally, this figure is just a reference since it varies very much according to the similarity algorithm used. In the present experiment, the largest variation obtained was with Jaro-Winkler.

## 4 Conclusion

Considering that there is no previous formal specification for Metaphone to Brazilian Portuguese language, this work not only provides a new one, but also shows how this specification is better than the main Metaphone algorithm, through the analyses of more than 2 million names.

The qualitative comparison is important to lead the experiments with rule variations in the algorithm itself, as well as in other similar ones, to verify how it affects the way the words are clustered, impacting in the textual searches, specially for its more immediate application, that is, to find similar names and addresses, albeit spelled differently.

**Acknowledgments.** João Luís G. Rosa thanks Fapesp - Fundação de Amparo à Pesquisa do Estado de São Paulo, Brazil, for the research support under project number 2008/08245-4, with which this paper is associated. Also, the authors would like to thank the anonymous reviewers for their constructive criticism and useful suggestions, and their families for the unconditional support.

## References

1. Freeman, A.T., Condon, S.L., Ackerman, C.M.: Cross linguistic name matching in english and arabic: a “one to many mapping” extension of the levenshtein edit distance algorithm. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, NAACL 2006, pp. 471–478. Association for Computational Linguistics, Stroudsburg (2006), <http://dx.doi.org/10.3115/1220835.1220895>
2. Jaro, M.A.: Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Association 84(406), 414–420 (1989), <http://dx.doi.org/10.2307/2289924>

3. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10, 707 (1966)
4. Odell, M.K., Russell, R.C.: U.S. Patents 1261167 (1918), 1435663 (1922)† (1918/1922), cited in Knuth (1973)
5. Philips, L.: Hanging on the metaphone. *Computer Language* 7(12) (1990)
6. Philips, L.: The double metaphone search algorithm. *C/C++ Users Journal* 18(5) (June 2000)
7. Piltcher, G.: Correção de palavras em chats: Avaliação de bases para dicionários de referência. In: *Anais do XXV Congresso da Sociedade Brasileira de Computação*, pp. 2228–2237 (2005)
8. Sanae, C.: A comparison and analysis of name matching algorithms. *Proceedings of World Academy of Science, Engineering and Technology* 21, 252–257 (2007)
9. Snae, C.: A comparison and analysis of name matching algorithms. *International Journal of Applied Science. Engineering and Technology* 21, 252–257 (2007)
10. UzZaman, N., Khan, M.: A double metaphone encoding for approximate name searching and matching in bangla. *Computational Intelligence*, 108–113 (2005)
11. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: *Proceedings of the Section on Survey Research*, pp. 354–359 (1990)
12. Zobel, J., Dart, P.W.: Phonetic String Matching: Lessons from Information Retrieval. In: Frei, H.P., Harman, D., Schäble, P., Wilkinson, R. (eds.) *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pp. 166–172. ACM Press, Zurich (1996), <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2138>

# Robust and Fast Two-Pass Search Method for Lyric Search Covering Erroneous Queries Due to Mishearing

Xin Xu and Tsuneo Kato

KDDI R&D Laboratories, Inc.  
2-1-15 Ohara Fujimino-shi, Saitama, 356-8502, Japan  
{sh-jo,tkato}@kddilabs.jp

**Abstract.** This paper proposes a robust and fast lyric search method for music information retrieval (MIR). The effectiveness of lyric search systems based on full-text retrieval engines or web search engines is highly compromised when the queries of lyric phrases contain incorrect parts due to mishearing. Though several previous studies proposed phonetic pattern matching techniques to identify the songs that the misheard lyric phrases refer to, a real-time search algorithm has yet to be realized. This paper proposes a fast phonetic string matching method using a two-pass search algorithm. It consists of pre-selecting the probable candidates by a rapid index-based search in the first pass and executing a dynamic-programming-based search process with an adaptive termination strategy in the second pass. Experimental results show that the proposed search method reduces processing time by more than 87% compared with the conventional methods, without loss of search accuracy.

## 1 Introduction

Current commercial music information retrieval systems accept queries in a range of forms by text, humming, singing, and acoustic music signals. Among these, text queries of lyric phrases are commonly used [1]. As many MIR systems apply full text search engines to lyric search, it has been widely regarded that the issue of lyric search has been solved by state-of-art text retrieval techniques. However, some investigations on the real world queries suggested that users are likely to input incorrect lyric phrases into MIR systems resulting in a failure. The incorrect lyric phrases are due to mishearing or the unreliability of human memory, as users only memorize the lyric phrases when they are impressed by hearing a part of a song without the aid of a lyric sheet. An analysis of Japanese question and answer website found that 19% of queries replace a word with another word having a similar pronunciation [4]. Xu's research verified that major commercial web search engines implemented with fuzzy algorithms, were not helpful for misheard queries [2].

Several related studies attempted to use phonetic string matching methods to solve this problem, which were verified to be more robust. Ring and Uitenbogerd [3] tried to find the correct lyric by minimizing the edit distances between

phoneme strings of queries and the lyrics. To model the similarity of misheard lyrics to their correct versions statistically, Xu et al. proposed an "acoustic distance" derived from a phoneme confusion matrix based on an automatic speech recognition experiment [4]. It was obtained by training a phonetic acoustic model for a speech recognition engine using Japanese phonetically balanced telephone speech and counting the number of correct and incorrect phonemes on the results of the speech recognizer. In a similar study, Hussein [5] introduced a probabilistic model of mishearing trained using examples of actual misheard lyrics, and developed a phoneme similarity scoring matrix based on the model.

To realize satisfactory music distribution services, the lyric search must be at once robust and fast. Most of the implementations of the search algorithm are based on exhaustive dynamic programming (DP) matching over the entire search space of lyrics. The computational complexity results are in the order of  $m * n * I_t$  per query, where  $m$  is the length of the query,  $n$  is the average length of a lyric and  $I_t$  is the number of lyrics to search. Since commercial MIR systems usually provide hundreds of thousands of lyrics, the computational complexity is too high to realize a real-time search.

Conventional high-speed DP matching processors use index or tree-structured data to pre-select the hypothetical candidates [7]. As an example, [7] used a suffix array as the data structure and applied phoneme-based DP matching for detecting keywords quickly from a very large speech database without using a large memory space. Moreover, in order to avoid an exponential increase in the processing time caused by increasing keyword length, a long keyword is divided into short sub-keywords. As well as other high-speed DP methods, it used a predetermined threshold that is dependent on the length of the queries to prune out the impossible paths during the DP process.

However, lyric search has a distinctive characteristic that it is too difficult to determine an absolute threshold to decide whether a lyric is the exact correct one for the incorrect query or not, since it is related to the individual differences of mishearing. Therefore, the previous studies on lyric search used a common criterion such that by looking up the entire lyric search space, a lyric of the minimum distance from the query is estimated as the user's target. Based on the investigation of real world queries, the DP distances between the queries and the correct lyrics have no statistical relationship with the lengths of the queries. The conventional high-speed DP processors are not able to keep high search accuracy for the lyric search case.

The authors proposed a fast and robust two-pass search method that uses an inverted-index in the first pass and DP-based search with an adaptive termination strategy in the second pass. In the first pass, the proposed method pre-selects the probable lyric candidates by means of a rapid approximate search based on the accumulation of pre-computed and indexed partial acoustic distances. Then, in the second pass, the candidate lyrics are sorted by the approximate acoustic distances and evenly divided into groups. The exhaustive DP matching between the query and the lyrics is carried out group by group. During the DP matching, a cut-off function is calculated by the DP distances. Once the function value

exceeds a predetermined threshold at some group, which means the correct lyrics have been found, the search is terminated. The experimental results show that the processing time is greatly reduced by using the proposed two-pass search strategy, without loss of search accuracy.

The remainder of this paper is organized as follows: the analysis of mistaken queries is described in Section 2. In Section 3, a fast and robust search method is introduced. In Section 4 the experiments are carried out to evaluate the proposed method in terms of search accuracy and processing time. The paper is summarized in Section 5.

## 2 Analysis of Mistaken Queries

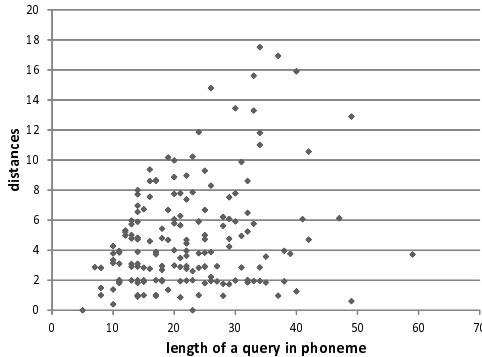
To analyze real word queries of lyric phrases for MIR, the authors investigated some question and answer community web sites, where many questions to request the names of songs and singers by lyric phrases were found. A total of 220 misheard or mismemorized queries were collected. This type of mistaken query can be categorized into a class as substitution of a word with another word having a similar pronunciation; or substitution of words where the spelling is unknown with syllable strings having a similar pronunciation. Two examples of queries are given in Fig. 1. “/kotoganai/” and “/kotobawanani/” have similar pronunciations while the text strings have no common parts. In the second example, Japanese syllable string is used as a query that has a similar pronunciation to English phrase, “You’ve been out riding fences for so long now” in the target lyric. This was used when users were not able to spell the foreign words that they heard in a song. Therefore, textual or semantic retrieval algorithm does not work well. A search test was carried out to evaluate how robust web search engines are against collected incorrect queries due to mishearing and memorizing. The results are shown in Table 1. Correct queries mean the correct versions of the incorrect queries. Comparing the number of hits with the correct queries, the performance of both web search engines are severely degraded by incorrect queries.

		Examples	
		Correct lyric	Mistaken queries
Ex. 1	Text (Japanese):	好きな事がない	好きな言葉は何
	Pronunciation:	/sukina{kotoganai/	/sukina{kotobawanani /
	Meaning:	There is <b>nothing</b> I like.	What are you favorite <b>words</b> ?
Ex. 2	Text:	You've been out riding fences for so long now	ユーベーナウプラウドゥンシーンクスゾーセングナウ
	Pronunciation:	/yuubiibiiNNautoraidiNNgufeNNshizufooso oroNNgunau /	/yuubeenaupurauduNNshiiNNkususooseNNg unau/
	Meaning:	You've been out riding fences for so long now	* no actual meaning

Fig. 1. Examples of misheard or mismemorized queries

**Table 1.** Number of hits by two popular web search engines

web search engines	Web Search Engine 1	Web Search Engine 2
220 correct queries	175	157
220 incorrect queries	27	16

**Fig. 2.** The distribution of the length of a query in phoneme and the DP matching distances from the correct lyric for the real world incorrect queries

As introduced in Section 1, the conventional high-speed DP-based search method can adjust the pruning threshold depending on the length of queries. To find out whether it is a practicable approach to the lyric search problem, the DP matching distance between the queries and the correct lyric are also analyzed. In Fig. 2, the horizontal axis is the phoneme number of each query, representing the length of the queries. The vertical axis is the acoustic distance between the queries and the correct lyrics that is defined in [4]. It shows that phoneme number of queries is spread in a broad range from 5 to 57. In addition, the distance values between the queries and the correct lyrics show no statistical relationship with the length of queries. Therefore, it is practically difficult for the conventional method, such as the one in [7], to find the appropriate threshold based solely on the length of queries.

### 3 Fast Two-Pass Search Algorithm in Consideration of Acoustic Similarity

A method with two-pass search strategy is proposed and realized with following steps: preparing work including acoustic distance definition and index construction, a rapid index-based search in the first pass and a DP-based search process with an adaptive termination strategy in the second pass.

### 3.1 Acoustic Distance Derived from a Phoneme Confusion Matrix

In order to take the degree of acoustic confusability between phonemes into account for string matching, this paper introduces acoustic distance proposed in [4], which has been proved to be robust against incorrect queries due to mishearing. Acoustic distance between two strings is calculated by DP matching with cost values derived from phonetic confusion probabilities instead of the constant cost values used for edit distance. First, a phonetic confusion matrix is obtained by running a phoneme speech recognizer over a set of speech and by aligning the phoneme strings of recognition results with reference phoneme strings, which uses the same speech recognition experiment as in [8].

For the elements of the confusion matrix,  $n(p, q)$  means the number of phoneme  $q$  obtained as recognition results by the actual utterances of phoneme  $p$ . As “ $\phi$ ” represents a null,  $n(\phi, p)$  means the number of the wrongly inserted phoneme  $p$  (insertion) and  $n(p, \phi)$  means the number of the deleted phoneme  $p$  (deletion).  $M$  represents the set of phonemes including null.

For each phoneme  $p$ , the phonetic confusion probabilities of an insertion  $P_{ins}(p)$ , deletion  $P_{del}(p)$  and substitution for phoneme  $q$   $P_{sub}(p, q)$  are calculated on the basis of the confusion matrix elements, by Eq.1~3.

$$P_{ins}(p) = \frac{n(\phi, p)}{\sum_{k \in M} n(k, p)} \quad (1)$$

$$P_{del}(p) = \frac{n(p, \phi)}{\sum_{k \in M} n(p, k)} \quad (2)$$

$$P_{sub}(p, q) = \frac{n(p, q)}{\sum_{k \in M} n(p, k)} \quad (3)$$

As a large value of  $P_{ins}(p)$  represents a high confusability for an insertion of  $p$ , it corresponds to a low cost of an insertion operation for  $p$  in string matching based on DP. Therefore the value of insertion cost  $C_{ins}(p)$  is calculated by Eq.4. In the same way, the value of deletion cost  $C_{del}(p)$  and substitution cost  $C_{sub}(p, q)$  are calculated from the corresponding phonetic confusion probabilities by Eq.5 and Eq.6.

$$C_{ins}(p) = 1 - P_{ins}(p) \quad (4)$$

$$C_{del}(p) = 1 - P_{del}(p) \quad (5)$$

$$C_{sub}(p, q) = 1 - P_{sub}(p, q) \quad (6)$$

Second, with the calculated cost values, edge-free DP matching between the phoneme strings  $S_1, S_2$  is carried out by Eq.7~9. Here,  $S[x]$  is  $x$ th phoneme of phoneme string  $S$  and  $len(S)$  means the length of  $S$  ( $S_1, S_2 \in S$ ).  $D(i, j)$  designates the minimum distance from the starting point to the lattice point  $(i, j)$ .  $D_{S_1, S_2}$  is the accumulated cost of DP matching between  $S_1$  and  $S_2$ , which

is defined as the acoustic distance. It reflects acoustic confusion probability for each phoneme.

1. Initialization:

$$D(0, j) = 0 (0 \leq j \leq \text{len}(S_2)); \quad (7)$$

2. Transition:

$$D(i, j) = \min \begin{cases} D(i, j - 1) + C_{ins}(S_2[j]) \\ D(i - 1, j - 1) + C_{sub}(S_1[i], S_2[j]) \\ D(i - 1, j - 1), (\text{if } S_1[i] = S_2[j]) \\ D(i - 1, j) + C_{del}(S_1[i]) \end{cases} \quad (8)$$

3. Determination

$$D_{S_1, S_2} = \min\{D(\text{len}(S_1), j)\} (0 < j \leq \text{len}(S_2)); \quad (9)$$

### 3.2 Preliminary Indexing

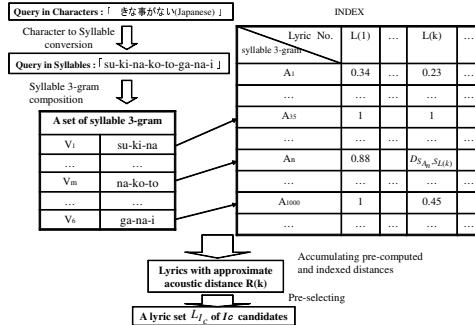
Theoretically, DP matching computation for the acoustic confusion distance between queries and lyric text should be done beforehand. However, this is impossible in reality because the number of query patterns is too numerous to be predicted. An inverted index construction is preliminarily incorporated for the first pass search. The lyrics  $L_{I_t}$  are converted into syllable strings using a morphological analysis tool such as Mecab [6]. The syllable strings are converted into phoneme strings by referring to a syllable-to-phoneme translation table. Consequently, a phoneme string  $S_{L(k)}$  represents a lyric  $L(k)$  ( $L(k) \in L_{I_t}$ ). On the other hand, a list of linguistically existing units of  $N$  successive syllables (syllable  $N$ -gram)  $A_1 \dots A_n$  are collected from the lyric corpus. The units are organized as index units for fast access, as shown in Fig. 3. The acoustic distance  $D_{S_{A_n}, S_{L(k)}}$  between the phoneme strings of  $A_n$  and  $L(k)$  are pre-computed by Eq.7~9 and stored in the index matrix. It can be regarded as an index of acoustic confusion.

### 3.3 Index Search in the First Pass

By accessing the index described above, a fast search is realized using the following steps. The flowchart is shown in Fig. 3:

1. The input query  $Q$  is converted into a syllable string  $v$  by Macab.
2. By Eq.10 the syllable string is converted into syllable  $N$ -gram sets,  $V_1, \dots, V_m, \dots, V_M$ . Here,  $v[m]$  is the  $m$ th syllable of  $v$ .

$$V_m = \{v[m], v[m + 1], \dots, v[m + N - 1]\}; \quad (10)$$



**Fig. 3.** Flowchart of the first pass search

3.  $V_1, \dots, V_m, \dots, V_M$  are matched with the index units  $A_1, \dots, A_n, \dots$ . By accumulating the pre-computed and indexed distance values  $D_{S_{A_n}, S_{L(k)}}$ , the approximate acoustic distance  $R(k)$  is calculated by Eq.11.

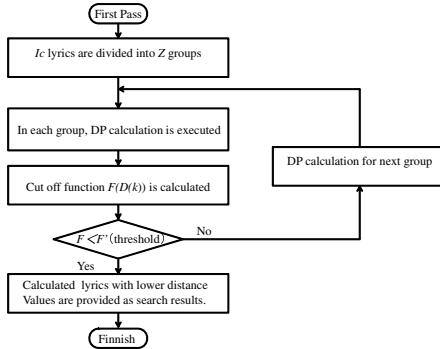
$$R(k) = \sum_{m=1, \dots, M} D_{S_{A_n}, S_{L(k)}}, (V_m = A_n) \quad (11)$$

4. To narrow the search space of lyrics,  $L(k)$  with higher  $R(k)$  is pruned off, and a lyric set  $L_{I_c}$  containing  $I_c$  ( $I_c < I_t$ ) best lyric candidates is preserved for the second pass.

As seen in the four steps, the order of the syllable  $N$ -grams is not considered in the first pass.

### 3.4 DP-Based Search Process with an Adaptive Termination Strategy in the Second Pass

By means of the pre-selection in the first pass, the range of target lyrics is narrowed down to  $L_{I_c}$ . DP matching with the lyrics in  $L_{I_c}$  is then carried out to calculate the precise distance. The candidates with the minimum acoustic distance  $D(k)$  are indicated as the search results. Since the  $R(k)$  is calculated as an approximate value of DP matching distance  $D(k)$ , after  $L_{I_c}$  is sorted by  $R(k)$ , the correct lyric with the minimum  $D(k)$  rises into the forward ranks in most cases. Thus, instead of the exhaustive DP matching over the entire set of pre-selected lyrics  $L_{I_c}$ , a DP-based search with an adaptive termination criterion is proposed. The termination is adaptive to a cut off function  $F(D(k))$ . The second pass search is designed as shown in the flowchart in Fig. 4. Lyrics  $L_{I_c}$  are first sorted by  $R(k)$  and then divided into  $Z$  groups, thus each group has  $I_c/Z$  lyrics. A DP matching calculation is executed in one group after another, while the cut-off function  $F(D(k))$  does not fulfill a terminating condition. Once the value of  $F(D(k))$  reaches a threshold  $F'$ , the DP matching process is aborted at that



**Fig. 4.** Flowchart of the second pass search

group. Within the lyrics of the calculated groups, the lyrics are ranked in the order of the  $D(k)$ , and then the lyrics with lower distance values are provided as search results.

## 4 Experiments

In order to decide the parameters of the proposed method, preliminary experiments are carried out. Then, the proposed method is compared with three other conventional methods to evaluate its performance on search accuracy and processing time.

### 4.1 Preliminary Experiments to Determine Parameters for the First and the Second Passes

A database of 50000 lyric texts was collected. It contains both Japanese and English lyrics. To find corpus-independent parameters, a test set of queries that are different from the one described in Section 2 were used. From a user-submitted misheard lyrics website [9], 842 misheard lyric queries in English were collected. The lyrics corresponding to the queries are all included in the database. The following parameters in the proposed method need to be decided:

- $I_c$ : the number of candidates in the first pass
- $F$ : the cut off function in the second pass

First, an experiment has been carried out to decide  $I_c$ . The first pass search using the index described in Section 3.3 is executed to investigate the relationship between search accuracy and  $I_c$  to choose the best value for  $I_c$ . The results are shown in Fig. 5. The horizontal axis shows the values of each tested  $I_c$  from 100 to 2,000, and the vertical axis is the hit rate, which is defined as the rate of the total number of hits within the candidates to the total number of search

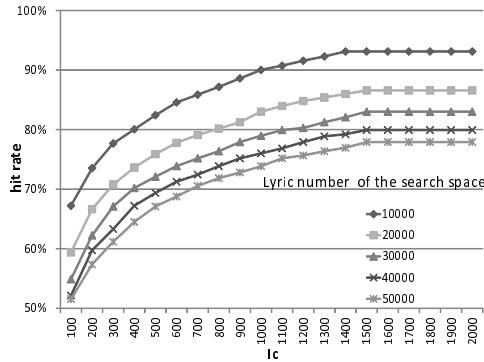


Fig. 5. Rlationship between hit rates and  $I_c$  for various sizes of lyric database

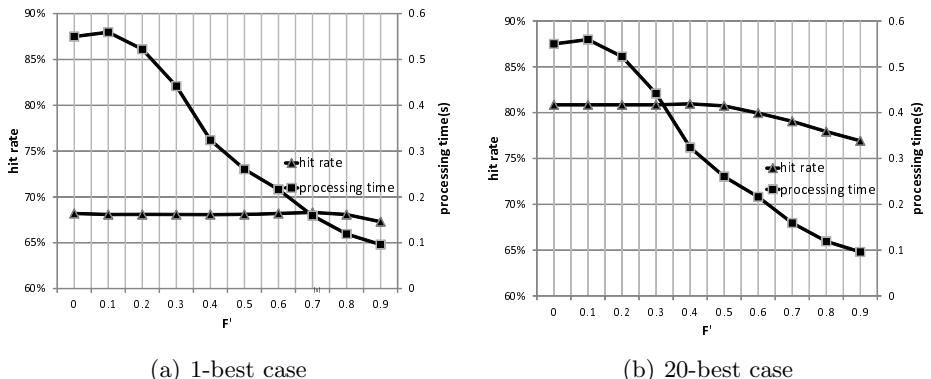


Fig. 6. Search accuracy and processing time with respect to  $F'$

accesses. Each line represents a different number of lyrics in the search space. The hit rates are almost saturated when  $I_c$  is larger than 1500, in spite of the change of search space. Therefore,  $I_c$  is set to 1500 in this paper.

Second, an investigation was undertaken to decide  $F$ . In most of the 842 queries, it was found that, by sorting the lyrics according to the approximate distance  $R(k)$  and dividing them into groups, the target lyric has a significantly lower DP distance  $D(k)$  than other lyrics in the same group. Based on the investigation above,  $F$  is defined by Eq.12, where  $D_{min}$  is the minimum value and  $D_{mean}$  is the mean value of the group. The experimental results that reveal the relationship between the processing time and search accuracy with respect to  $F'$  are shown in Fig. 6, where the horizontal axis represents  $F'$ , the right vertical axis represents processing time, and the left vertical axis represents the hit rate. Panel(a) shows the results in the case of 1-best, while panel(b) shows the results in the case of 20-best.  $T$ -best means that the top  $T$  candidates of the

ranked lyrics. Both panels show that the value of  $F'$  between  $0.4 \sim 0.6$  is the optimal threshold to reduce processing time without deteriorating search accuracy.

$$F = \frac{D_{min}}{D_{mean}} \quad (12)$$

## 4.2 Evaluation of the Overall Performance

To evaluate the overall performance of the proposed method, its hit rate and processing time are compared with those of conventional methods.

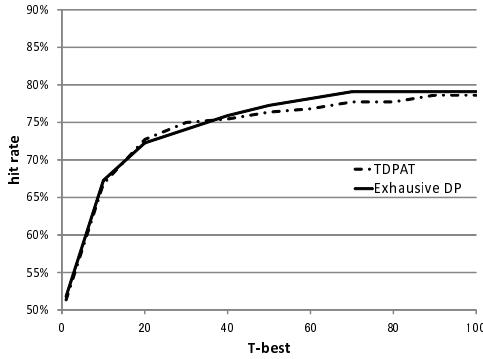
Four methods as described below are compared.

- "Exhaustive DP" is an exhaustive DP-based search over the entire search space of lyrics.
- "Two-pass DP search with Adaptive Termination (TDPAT)" is the proposed method described in Section 3.  $F'$  is tuned from 0 to 1. Considering the balance of index size and search accuracy, here  $N$  of syllable  $N$ -gram index is set to 3. A total of 50,000 entries of syllable 3-grams, which cover 92% of all syllable 3-grams in the collected lyric corpus, are prepared in the index. As all the syllable 3-grams which exist in the queries are prepared, no search errors come from out-of vocabulary syllable 3-grams in the experiments. The acoustic distance is normalized by the length of the corresponding DP path.
- "Two-pass DP search with Distance-based Termination (TDPDT)" is a method that has almost the same processes as the proposed method, with the exception that the DP is terminated when the acoustic distance  $D(k)$  exceeds a predetermined threshold value, that is tuned from 0 to 1.
- "High-speed DP search with Suffix Array (HDPSA)" is based on the method in [7]. In the experiment, since the input query and the database are both text, the texts are converted into syllable strings instead of the phonemes originally used, and divided into syllable  $N$ -gram. Also, a suffix array records the boundary information of lyrics in order to avoid matching queries across two lyrics. Here  $N$  is set to 3 because it was shown in a preliminary experiment that this value results in better performance than when  $N = 2$  or  $N = 4$ . The total threshold is tuned from 0 to 1.3 to find the optimal value balancing search accuracy and processing time.

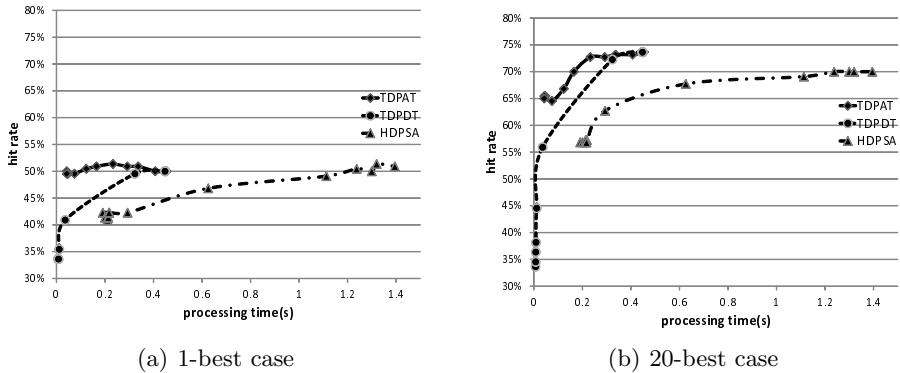
The experiments were executed on a personal computer (Intel Core 2 Duo E8400 3.00GHz CPU, 4G RAM). The lyric database contains 10,000 lyrics. The test set consisted of the 220 incorrect queries mentioned in Section 2.

First, to evaluate the robustness of "TDPAT", a comparison with "Exhaustive DP" and "TDPAT" (where  $F' = 0.4$ ) is represented in Fig. 7. "TDPAT" keeps almost the same hit rate as the  $T$  of  $T$ -best is ranged from 1 to 100. It is due to the well-designed two-pass search algorithm that avoids the loss happening in the pre-selection and the adaptive termination processes.

Then, the search accuracy and time complexity of three high-speed DP methods "TDPAT", "TDPDT" and "HDPSA" are shown in two panels of Fig. 8,



**Fig. 7.** Search accuracy of "TDPAT" and "Exhaustive DP"



**Fig. 8.** Average processing times and search accuracy of three search methods

where the horizontal axis represents processing time and the vertical axis represents hit rate. Each point in the figures indicates the processing time cost and the hit rate achieved when a particular threshold is set. Panel(a) and (b) show the results in the cases of 1-best and 20-best, respectively. "TDPAT" shows the similar relationship between search accuracy and processing time with respect to  $F'$ , comparing with the results of preliminary experiment. When  $F'$  is set up to 0.4, the average processing time for each query only costs 0.23 seconds, which reduces processing time by 51% without any loss of search accuracy compared with the time when  $F'$  is 1, that is exhaustive DP matching which deals with all of  $L_{I_c}$ . As shown in Fig. 8, "TDPAT" obtains higher search accuracy than "TDPDT" at the same processing time, especially in short processing time. It proves that the hypothesis of the definition for  $F(D(k))$  is correct. Also, the performance of "TDPAT" is superior to that of "HDPSA" in terms of both processing time and search accuracy. In the case of 1-best, to achieve the same hit rate of 50%, "TDPAT" reduces processing time by 96% compared with "HDPSA". In the case of 20-best, to achieve the same hit rate of 70%, "TDPAT" reduces processing

time by 87%. These results indicate that the proposed search algorithm is more efficient than the conventional algorithms that determine the pruning threshold according to the length of queries. And finally, an analysis of the queries that failed to identify the target lyric text using the proposed method reveals that most of them are smaller than 6 syllables, indicating that the distance between the query and lyric texts was too close to make the target lyric distinguishable.

## 5 Conclusions

This paper proposed a robust and fast lyric search method with a two-pass search algorithm using an index-based approximate pre-selection for the first pass and DP-based search process with an adaptive termination strategy in the second pass. For the incorrect queries that are misheard or mismemorized, the experimental results show that the proposed method keeps almost the same search accuracy with an exhaustive DP-based search over the entire search space of lyrics. It also makes a real-time search an actuality, and significantly reduces processing time by more than 87% compared with the conventional high-speed DP search method. It was shown to be the most practical solution for misheard queries and strikes the optimal balance between high search accuracy and fast processing time.

## References

1. Downie, Cunningham: Toward a theory of music information retrieval queries: System design implications. In: Proceedings of ISMIR 2002, pp. 299–300 (2002)
2. Xu, X., Naito, M., Kato, T., Kawai, H.: An Introduction of a Fuzzy Text Retrieval System For Music Information Retrieval. Information Processing Society of Japan SIG. Notes 127, 41–46 (2008)
3. Ring, N., Uitenbogerd, A.: Finding ‘Lucy in Disguise’: The Misheard Lyric Matching Problem. In: Proceedings of AIRS 2009, pp. 157–167 (2009)
4. Xu, X., Naito, M., Kato, T., Kawai, H.: Robust and Fast Lyric Search Based on Phonetic Confusion Matrix. In: Proceedings of ISMIR 2009, pp. 417–422 (2009)
5. Hirjee, H., Brown, D.G.: Solving Misheard Lyric Search Queries using a Probabilistic Model of Speech Sounds. In: Proceedings of ISMIR 2010, pp. 137–148 (2010)
6. <http://mecab.sourceforge.net>
7. Katsurada, K., Teshima, S., Nitta, T.: Fast Keyword Detection Using Suffix Array. In: Proceedings of INTERSPEECH, pp. 2147–2150 (2009)
8. Yamada, M., Kato, T., Naito, M., Kawai, H.: Improvement of Rejection Performance of Keyword Spotting Using Anti-Keywords Derived from Large Vocabulary Considering Acoustical Similarity to Keywords. In: Proceedings of INTERSPEECH, pp. 1445–1448 (2005)
9. <http://www.kissthisguy.com/>

# Bootstrap-Based Equivalent Pattern Learning for Collaborative Question Answering

Tianyong Hao<sup>1</sup> and Eugene Agichtein<sup>2</sup>

<sup>1</sup>Department of Chinese, Translation and Linguistics

City University of Hong Kong, Hong Kong

haotianyong@gmail.com

<sup>2</sup>Mathematics & Computer Science Department,

Emory University, USA

eugene@mathcs.emory.edu

**Abstract.** Semantically similar questions are submitted to collaborative question answering systems repeatedly even though these questions already contain best answers before. To solve the problem, we propose a precise approach of automatically finding an answer to such questions by identifying “equivalent” questions submitted and answered. Our method is based on a new pattern generation method *T-IPG* to automatically extract equivalent question patterns. Taking these patterns from training data as seed patterns, we further propose a bootstrap-based pattern learning method to extend more equivalent patterns on these seed patterns. The resulting patterns can be applied to match a new question to an equivalent one that has already been answered, and thus suggest potential answers automatically. We experimented with this approach over a large collection of more than 200,000 real questions drawn from Yahoo! Answers archive, automatically acquiring over 16,991 equivalent question patterns. These patterns allow our method to obtain over 57% recall and over 54% precision on suggesting an answer automatically to new questions, significantly improving over baseline methods.

**Keywords:** Collaborative question answering, Equivalent pattern, Bootstrap, Pattern extension.

## 1 Introduction

Collaborative Question Answering (CQA) systems, such as Yahoo! Answers, Baidu Knows, and Naver, are becoming popular online information services. One of the useful by-products of this popularity are the resulting large archives of questions, answers and ratings – which in turn could be good sources of information for automatic question answering. For example, Yahoo! Answers [1] alone has acquired an archive of more than 40 Million questions and 500 Million answers, according to 2008 estimates.

Many questions, that are syntactically different while semantically similar, contain best answers posted before in these CQA archives. While identifying all such groups of questions and making usage of them for new question answering are the goal of

this work, we propose exploiting the existing archives to first identify a small group of clearly equivalent questions, and then use these groups to learn and extend equivalent patterns to match more questions.

Our approach is based on the following observation: in CQA systems, an asker often chooses one posted answer as “best” if it fulfills the information need expressed by the question. Therefore, in the cases when the best answers chosen for *different* questions are exactly the same, these questions express the same information need, and thus semantically similar.

Based on this observation, we propose an automatic question answering method over CQA archives by generating equivalent question patterns. First, we retrieve “equivalent” question groups from a CQA archive by grouping the questions by the text of “best” answers. To avoid generating spurious equivalent question groups, such as the case of certain different questions share a same answer by chance, we propose a topical diversity (TD) filter, based on the estimation of the topic similarity between the question groups. To extract equivalent patterns, we explore a new syntactic tree-based pattern generation method named *Tree-based Incremental Pattern Generation (T-IPG)*. On the same question, *T-IPG* extracts a group of patterns incrementally and these patterns are then automatically evaluated, by matching against the whole question set, to select the best pattern specificity for a given equivalent group. The resulting equivalent pattern groups are then used to match all questions in the archive. By comparing question similarity and answer similarity, extended equivalent patterns can be extracted round by round using a bootstrap-based pattern extension method. When a new question is submitted, it is compared to the set of available equivalent patterns. In case of a match, the best answer from previously submitted questions in the matched group could be returned.

Experiments over a dataset of more than 200,000 questions retrieved from Yahoo! Answers are preformed to test the effectiveness of the propose method. We initially detect 1,349 equivalent patterns in 452 groups, which are then used to learn more equivalent patterns. The final extended 16,991 patterns are applied to automatically seek a best answer to new (hold-out) set of questions. Our method correctly suggests an answer to a new question, 54.5% of the time – outperforming previously reported state-of-the-art translation-based method for similar question finding.

The rest of this paper is organized as follows: Section 2 introduces related work. In Section 3, the syntactic tree-based pattern generation method is presented. Section 4 describes the bootstrap-based pattern extension method in detail. The experimental results with evaluation are shown in Section 5 and Section 6 concludes the paper.

## 2 Related Work

Our work builds on the long tradition of research in automatic Question Answering (QA). Automatic QA systems attempt to find the most relevant parts (usually in short paragraph or just one or two sentences) in long documents with respect to user queries. Auto-FAQ, Whitehead [2], relied on a shallow, surface-level analysis for similar question retrieval. FAQ-Finder, Hammond [3], adopted two major aspects, i.e., concept expansion using the hypernyms defined in WordNet and the TF-IDF

weighted score in the retrieval process. In the FAQ-Finder, certain question types may not be detected correctly, for examples, in the cases when interrogative words like "what" and "how" are the substrings of interrogative phrases "for what" and "how large", respectively. To eliminate the above problem in FAQ-Finder, Tomuro [4] combined lexicon and semantic features to automatically extract the interrogative words from question corpus. Besides WordNet, Lenz [5] retrieved FAQs via case-based reasoning (CBR). Sneiders [6] used question templates with entity slots that are replaced by data instances from an underlying database to interpret the structure of queries or questions. Berger et al. [7] proposed a statistical lexicon correlation method for FAQ retrieval.

With respect to pattern usage, some QA systems attempted to learn patterns to help identify potential answers. For example, Ion gave three different linguistic patterns to extract relevant information [18]. There have also been much prior efforts on automatic pattern extraction and most of them focused on extracting patterns from human-labeled training corpus. Ravichandran and Hovy [14] proposed a surface text pattern generation algorithm to find answers to new questions. Zhang and Lee [22] introduced a pattern learning algorithm to extract answer patterns for a given question. The essential idea was to find one answer instance and generalize the question target. However, the defined answer targets were too general to differentiate between the answer types thus the generated patterns are usually too domain-specific to be efficiently applied to a new domain. Mark and Horacio [12] extended Zhang and Lee's patterns by using four answer instances instead of one to overcome the over-generalization problem. Hu et al. [20] utilized a kind of semantic pattern for question answering, in which two granularity evaluation algorithms SIIPU and DEXT were used to control the granularity of the patterns in order to increase their flexibility. A more recent work was focusing on learning semantic pattern by Hao et al. [19]. However, the computational time required to directly process semantic patterns was high, with the result that the pattern does not appear to be feasible in processing of huge amount of data archives.

The idea of finding similar questions in CQA is related to passage retrieval in traditional QA, with the exception that question-to-question matching is much stricter than question-to-passage matching. There have been significant new efforts focusing on CQA retrieval (e.g., Wu et al. [21], Bian et al. [17] and Wang et al. [24]). Bernhard [15] consulted 6 different types of question similarity methods on WiKianswers, which is a CQA site. The comparison shown that Lucene's Extended Boolean Model get best performance but only overcome a little than term vector similarity. Jijkoun and Rijke [23] proposed to retrieve answers from frequently asked question pages on the Web and return a ranked list of QA pairs in response to a user's question. They used the implementation of the vector space model in Lucene as the core of retrieval system and exploited the performance of different models. However, the vector space similarity, as the core of all the baselines, processed same words between the user's questions and Q/A pairs while the similar syntax structures of questions were not concerned. Jeon [9, 10] used word translation probabilities to find similar questions and it was proved to exceed Cosine similarity method much. Kosseim [11] tried to improve question answering by retrieving equivalent answer patterns. However, all the manual and automatically generated patterns were based on TREC 8 & 9 data, which have quite unified formats. Thus the patterns cannot process common questions

in current CQA systems even the questions started with *why* and *which*. Jeon et al. [9, 10] extended this method by introducing word translation probability to find similar questions in CQA archives, and have shown significant improvements over previous methods. We will compare the method with our approach in this paper.

## 3 Equivalent Pattern Learning

### 3.1 Topical Diversity Filtering

While most questions that share exactly same “best” answer are indeed semantically equivalent, some may share the same answer by chance. To filter out such cases, we propose an estimate of Topical Diversity (TD), calculated based on the shared topics for all pairs of questions in one group. If the diversity value is larger than a threshold, the questions in this group are considered *not* equivalent, and no patterns are generated. To calculate the topical diversity on a question, we define the topics as “Notional Words” (NW), which contains head nouns, the heads of verb phrases identified by the OpenNLP parser [13]. Using these words as “topics”, we can obtain the topical diversity by calculating the not shared topics in the whole topics for a group of similar questions. To calculate these topics, we firstly compare the notational words of each two questions in the group and calculate the probability of the topics without sharing. Since a group may contain more than two questions, average probability is then calculated on all pairs in the same group. Therefore, topical diversity  $TD$  for a question group  $G$  is represented as equation (1).

$$TD(G) = \frac{1}{n(n-1)} \times \sum_{i=1}^n \sum_{j=1}^n (1 - \frac{|\mathcal{Q}_i \cap \mathcal{Q}_j|}{|\mathcal{Q}_i \cup \mathcal{Q}_j|}) \quad (i > j) \quad (1)$$

$\mathcal{Q}_i$  and  $\mathcal{Q}_j$  represent the notional word subsets of any two different questions in the same group  $G$ , which contains total  $n$  questions. From this equation, we can see that the diversity is higher when there are less shared topics in this group. After topical diversity filtering, only the question groups with diversity value lower than a threshold, which is further described in experiment section, are kept as equivalent question groups. These equivalent groups can be further used to generate equivalent question patterns.

### 3.2 Pattern Generation

Based on these filtered question groups, we can generate equivalent question patterns, which are the patterns generated in a same equivalent question group. The resulting generated patterns, regarded as seed patterns, are then used to extend and extract more equivalent patterns. In the pattern generation process, traditional chunk-based pattern generation methods usually consider the structure of original questions. However, some questions may be very long or they may contain subordinate clauses, which in turn could affect the performance of pattern matching. The syntactic tree-based method is to find and use different levels of syntactic tree to extract the “core”

structure of a question thus is more preferable. Based on this, we propose a new syntactic tree-based pattern generation method named *Tree-based Incremental Pattern Generation (T-IPG)*.

On a syntactic tree, the *T-IPG* method tries to extract all potential “valuable” patterns from root node to all leaf nodes incrementally. In the case of a question for generation is very long, many incremental generation steps may take and generation efficiency may be affected. To improve it by reducing computational volume, the *T-IPG* method firstly preprocesses the tree to merge nodes in a *single chain*, which is defined as follows:

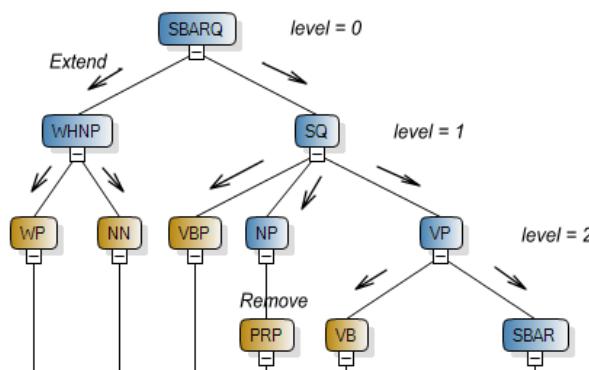
**Definition 1:** Given a node  $n_{x+1}$  and its parent node  $n_x$ ,  $(n_x \text{ to } n_{x+1})$  is a *single chain* if and only if  $n_{x+1}$  has only one child and is the only child of  $n_x$ .

From the definition, each node in a same *single chain* has only one child thus a *single chain* can be extended to contain more than two nodes. To merge them, all nodes in a *single chain* are compared with their priorities. The node with highest priority is selected to represent the other nodes. The priority of POS tag for a notional word (described in previous section) is predefined as larger than that of an interrogative word such as “WDT, WP, WP\$”. The priority of the latter is further larger than that of other types of POS tags.

This method starts to construct an initial pattern from the root node of a syntactic tree. The initial pattern is then extended to the leaf nodes of the tree level by level. With considering the parent-child relation in a level, the child nodes are added in each level from left to right in the tree. Each extension action forms a new extended sub-tree based on a sub-tree. The difference between the two sub-trees is defined as *incremental part*. A sub-tree is then judged to decide if it is “valuable” to be a pattern by the following constraints.

**Constraint 1:** the *incremental part* contains either a tag of notional word or a tag of interrogative word.

**Constraint 2:** the total number of tags in the sub-tree is larger than 1.



**Fig. 1.** Pattern generation on a question example using the *T-IPG*

**Table 1.** Incrementally generated patterns on the example

Patterns	Levels
WP NN VBP NP	2
WP NN VBP NP VB	3
WP NN VBP NP VB NN	4
WP NN VBP NP VB NN MD VB	5
WP NN VBP NP VB NN MD VB IN NN	7

The *T-IPG* generates a group of validated patterns for a query and the pattern quantity increases when the query is longer. To save computational cost brought by matching on large number of patterns for question answering, we further propose a *maximum pattern matching* method to acquire a most appropriate pattern in each generation procedure. The main idea is to find a generated pattern which can match original question (generation source) “better” than any other question. We define this “better” as matching gap  $\sigma$  and define matching score as  $MS$ , in which  $MS$  is calculated by the possibility of matched tags in the pattern with sequence. The matching gap to a certain pattern  $p$  thus can be represented as follows:

$$\sigma(p) = MS(p, q_p) - \text{Maximum}(MS(p, q_i)), q_p \neq q_i \quad (2)$$

Once we get  $\sigma$ , the best pattern can be selected by its sub-tree level. The reason to consider its level is that we suppose the higher level (root level is highest) is, the more questions this pattern can match. However, from equation (2), the matching gap is larger when the sub-tree level is lower, which is opposite to matching coverage. To balance the two factors, we define a matching threshold as  $\lambda$  and find the patterns with matching gap larger than the threshold. After that, only one pattern with matching gap is most “close” to  $\lambda$  but larger than  $\lambda$  is selected as the best pattern. The equation to find best pattern  $p_{best}$  is defined as follows:

$$p_{best} = \{p_i \mid \sigma(p_i) \geq \lambda; \sigma(p_i) \rightarrow \lambda\} \quad (3)$$

## 4 Bootstrap-Based Pattern Extension

Though the CQA dataset is large, the question groups that share the exact same answers are not too much. In our investigation of 215,974 questions and 2,044,296 answers crawled from the Yahoo! Answers, such kind of questions with exact same answers are only 2,166 before topical diversity filtering. Thus, the generated patterns directly on the limited training questions are far not enough for answering newly posted questions even with the incremental generation ability of the *T-IPG* method. Therefore, we propose *bootstrapping* the learning process by automatically acquiring additional training questions and name the method as “bootstrap-based pattern extension”.

This algorithm firstly generates equivalent patterns, as seed patterns, from initial training equivalent question groups. Such pattern groups are then matched with a large scale of QA archive to extract more equivalent question candidate groups.

Each question pair in the groups is evaluated by calculating both question similarity and answering similarity. The similarity calculation uses the normal *Cosine* similarity method to extend the similar question cases thus to extract more equivalent patterns. If the similarity is larger than a threshold, the question group is regarded as equivalent and is added into equivalent question group candidates for next round pattern generation. In each round, the generated patterns are evaluated with the metrics of *F1* measure, which is further described in evaluation section. In the case that the average *F1* score begins to drop, the extension loop stops and the generated pattern group candidates from all previous rounds are the final extended equivalent patterns, which are then added into our pattern database for next usage.

---

**Algorithm.** Bootstrap-based Equivalent Pattern Extension

---

1. **Input** a question set  $G_j < Q_j, BA_j >$  in the whole dataset  $\{QA\}$ ;
  2. Set final generated pattern list as  $EPL$ ; question set for next round generation as  $NG$
  3. **For** ( $j=1$ ;  $G_j$  is not empty;  $j++$ )
    4. **Foreach** (question group  $g$  **In**  $G_j$ )
      5.  $ep \leftarrow$  Generate pattern:  $T\text{-IPG}(g)$ ;
      6.  $EP_j = EP_j + ep$
    7. **End For**
    8. Match  $EPL$  and  $(EPL + EP_j)$  with  $\{QA\}$  to calculate *F1* score
    9. **If**  $F1(EPL + EP_j) < F1(EPL)$ 
      10. **Stop** iteration;
      11. **Else**
      12. Set  $G_{j+1} = NG \leftarrow$  **Pattern\_Extension**( $EP_j$ )
      13.  $EPL = EPL + EP_j$ ;
    14. **End If**
    15. **End For**
    16. **Output**  $EPL$ ;
- 

**Algorithm 1.** Bootstrap-based equivalent pattern extension

The detailed algorithm of the bootstrap-based pattern extension method is shown as Algorithm 1. Line 1 and 2 are the initial definition of parameters. Lines 3-15 are the main function of the bootstrap-based pattern generation. The pattern generation using the *T-IPG* on equivalent question groups is shown as lines 4-7. Lines 8-14 present the matching of equivalent patterns with the whole QA archive to calculate average *F1* score. If the *F1* score begins to drop, the extension iteration stops. Otherwise, the equivalent patterns generated in current step, as seed pattern, are sent to pattern extension function to acquire new question set for next round processing.

To be understood easily, the function of the pattern extension (line 12) is shown separately as Algorithm 2. This algorithm first matches each group of equivalent patterns on the whole QA archive as line 3. The similarity of matched questions and their answers are then calculated using Cosine as line 6 and line 7, respectively. The groups with the higher similarity than thresholds are kept as the equivalent question candidate groups. These groups, as shown in line 14, are returned to next round for further processing.

**Algorithm. Pattern\_Extension Function**

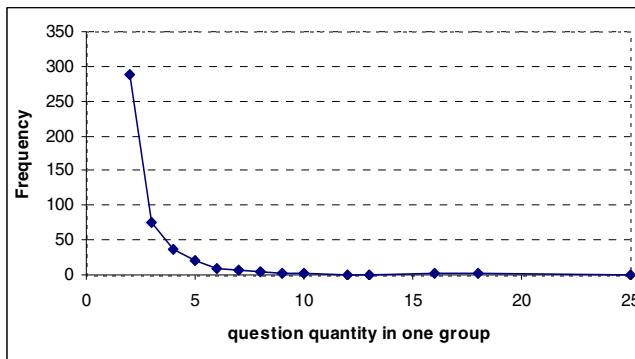
1. **Input** an equivalent pattern set  $EP$
2. **For**each (pattern group  $ep_i$  In  $EP$ )
3.      $G_i < Q_i, BA_i > \leftarrow \text{Match}(ep_i, \{QA\})$  with threshold  $\delta$
4.     **For**each (question  $q_m$  In  $G_i$ )
5.         **For**each (question  $q_n$  In  $G_i$  And  $q_n \neq q_m$ )
6.             Calcuulate similarity:  $\text{Cosine}(q_n, q_m) \rightarrow simq$ ;
7.             Calcuulate similarity:  $\text{Cosine}(\text{answer of } q_n, \text{answer of } q_m) \rightarrow sima$ ;
8.             **If** ( $simq > \tau_1$ ) **Or** ( $sima > \tau_2$ ) **Then**
9.                 Add  $q_m, q_n$  into  $NG$
10.             **End If**
11.         **End For**
12.         **End For**
13.     **End For**
14. **Output**  $NG$

**Algorithm 2.** Pattern\_Extension function

## 5 Experiment and Evaluation

We adapt standard evaluation metrics from information retrieval, namely, *Precision*, *Recall*, and *F1*-measure. The task is, for a given question, to retrieve its set of semantically equivalent questions (that is, questions from the same equivalent group but not itself). Therefore, *Precision* for this question is defined as, the number of correctly matched questions, divided by the number of the questions retrieved. Similarly, *Recall* is defined the correctly matched questions divided by the number of questions in the original group. Finally, the *F1* measure is computed in the standard way as  $2 * Precision * Recall / (Precision + Recall)$ .

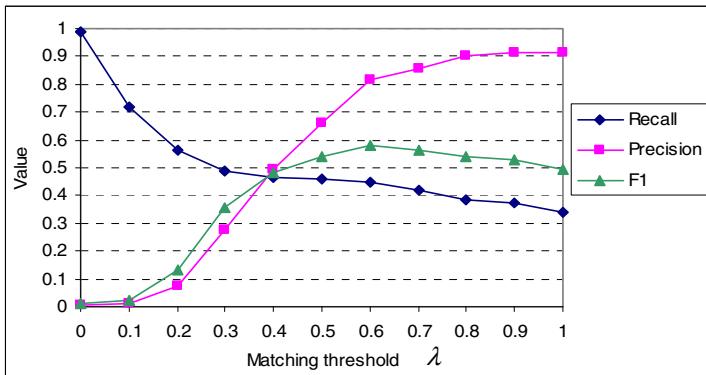
Our dataset consists of 215,974 questions and 2,044,296 answers crawled from Yahoo! Answers in 2008 [1]. From these questions, we acquired 833 groups of

**Fig. 2.** Question quantity distribution of whole dataset

similar questions distributed in 65 Yahoo! Answer categories. After automatic filtering by topical diversity calculation, 452 groups remain for parameter tuning and as seed data, with human verification, for equivalent pattern generation. These groups contain 1,349 questions, with, 2.98 questions per group on average. Fig.2 reports the distribution of group sizes, in which groups containing fewer than 5 questions account for almost 90% of all the questions.

In the experiment, the seed data is split into two categories: 603 questions for training (200 groups) and 746 questions for testing (the remainder). To make the experiments sound, we add a large set of additional questions on the testing data category to form two testing datasets. Dataset-1 contains the 746 testing questions with additional 10,000 questions and Dataset-2 is the whole archive of 215,974 questions. Since a full experiment on the large dataset is very time-consuming, we use a smaller one (Dataset-1) for evaluation of pattern generation methods and the other one (Dataset-2) for comparison with baseline methods.

The weight of using equivalent pattern (EP) for question matching is set as  $W_{EP}$  and that of notional word (NW) is  $1 - W_{EP}$  accordingly. Parameter  $\theta$  is a matching threshold used for pattern matching to find similar patterns. Defined in equation 2 and 3, parameter  $\lambda$  is a threshold for matching gap  $\sigma$  comparison.



**Fig. 3.** Performance with different values of threshold  $\lambda$  using the *T-IPG* method

To train all weights and parameters, with the 603 training questions, we firstly set the value of  $W_{EP}$  as 1, which means that the matching procedure only works on equivalent patterns other than notional words at this stage. Precision and recall are then calculated on the training dataset by using different values (0-1) of  $\theta$ . After that, the best value of  $\theta$  can be obtained by comparing best  $F1$  score on the precision and recall. The best value of  $\theta$  is further used to train  $W_{EP}$  by using equivalent patterns and notional words at the same time. To get more questions matched, the matching gap  $\sigma$  is set to a very small value 0.001 from experimental experience. With the trained parameters  $W_{EP}$ ,  $\theta$  and  $\sigma$ , the performance is calculated on the training dataset again to find best value of  $\lambda$  considering highest  $F1$  score. Fig.3 shows the question matching performance with different values of  $\lambda$  using the *T-IPG* method.

The final trained parameter values are reported in Table 2, and are further used for the subsequent experiment.

**Table 2.** Final trained parameter values

Method	$W_{pattern}$	$\theta$	$\lambda$
<i>T-IPG</i>	0.2	0.7	0.6

We now evaluate the bootstrap-based pattern extension algorithm, which is used to extract more equivalent question groups on each round of extension. The whole archive is used for this extension procedure. From experiment result (we will not show it due to space limitation), in first two rounds, the number of generated equivalent patterns change a little as well as the performance (*F1* score). In round 3, 4 and 5, the number of generated patterns increases dramatically and the corresponding performance values continuously increase until to the round 5. Therefore, the system regards the round 4 as the best round. As the result of this experiment, 16,991 equivalent patterns are finally extracted in the pattern extension.

After parameter training and pattern extension, based on the Dataset-1, three variants of the *T-IPG* are further compared to find best one for question answering. *T-IPG*(EP) uses equivalent pattern only for question answering while *T-IPG*(NW) uses notional word only. *T-IPG*(EP+NW) is a combination method on both equivalent pattern and notational word with the trained parameter  $W_{EP}$ . Using the same matching method, the performances of the three variants are reported in Table 3. From the result, *T-IPG*(EP+NW) achieves the highest precision and *F1* score over the other methods while *T-IPG*(NW) has best recall thus *T-IPG*(EP+NW) is regarded as the best one considering *F1* score.

**Table 3.** Performance comparison of the three variants of the *T-IPG*

Method	Variants	Recall	Precision	<i>F1</i> score
<i>T-IPG</i>	<i>T-IPG</i> (EP)	0.472	0.431	0.451
	<i>T-IPG</i> (NW)	<b>0.496</b>	0.637	0.558
	<i>T-IPG</i> (EP+NW)	0.478	<b>0.763</b>	<b>0.588</b>

Baseline methods are implemented to compare with the *T-IPG*(EP+NW). A traditional Cosine model from Code Project [16] is firstly selected as it is a classical similarity calculation method. A vector space model - TFIDF(NW), which keeps the notional words filtered by phrase chunking, is also implemented as the improvement of the traditional TF-IDF method. To further compare with solid baselines, the Translation Model proposed by Jeon [9, 10] is also implemented. This method uses IBM statistical machine translation model to estimate word translation probabilities. Previous experiment results show that it overcomes LM and Okapi method specifically with significant improvements [9, 10]. To implement this method, we use GIZA++ toolkit [8] to learn the model with smoothing parameter setting to 0.01.

*T-IPG*(EP+NW), as the best from the previous experiment, and all the baselines are implemented and evaluated on the Dataset-2. The performances, as shown in Fig.4, show that the recall and precision of *T-IPG*(EP+NW) reach 57.1% and 54.5%, respectively. Considering the highest *F1* score 55.8%, the performance of our method *T-IPG*(EP+NW) outperform Cosine, TFIDF(NW), and the Translation Model significantly.

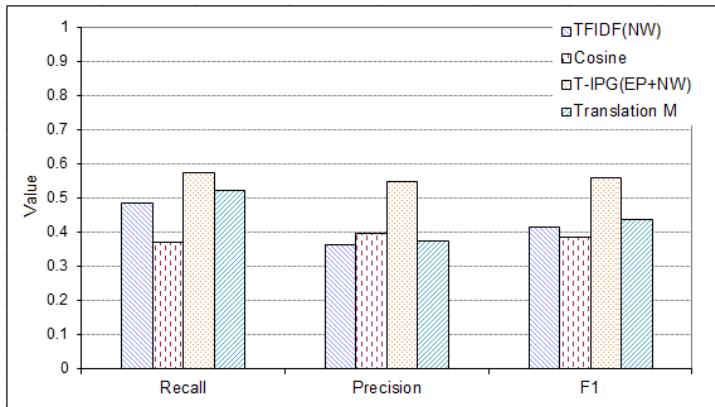


Fig. 4. Performance comparison with the baselines on the Dataset-2

## 6 Conclusions

This paper present a new syntactic tree-based pattern generation method *T-IPG*, and its variants: *T-IPG*(EP), *T-IPG*(NW) and *T-IPG*(EP+NW). The patterns generated automatically from initial equivalent question groups are regarded as seed patterns after topical diversity filtering. These seed patterns are further extended by a bootstrap-based pattern extension algorithm. The resulting patterns, combining syntactic patterns and notional words in the questions, can be used to answer new questions with existing answers. The experiment is conducted on a large collection of more than 200,000 real questions drawn from Yahoo! Answers archive. From the result, our method *T-IPG*(EP+NW) can achieve over 57% recall and over 54% precision on finding similar questions to new questions, significantly outperforming the baseline models for this task. Future improvements would focus on incorporating additional semantic information into the matching process.

## References

1. Yahoo! Answers (2011), <http://answers.yahoo.com/>
2. Whitehead, S.D.: Auto-FAQ: An Experiment In Cyberspace Leveraging. *Journal of Computer Networks and ISDN Systems* 28, 137–146 (1995)
3. Hammond, K., Bruke, R., Martin, C., Lytinen, S.: FAQ-Finder: A Case Based Approach to Knowledge Navigation. In: *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*, AAAI, pp. 80–86 (1995)

4. Tomuro, N.: Question Terminology and Representation for Question Type Classification. *Terminology* 10(1), 153–168 (2004)
5. Lenz, M., Hbner, A., Kunze, M.: Question Answering With Textual CBR. In: Proceedings of the International Conference on FQAS, Denmark, pp. 236–247 (1998)
6. Sneiders, E.: Automated Question Answering Using Question Templates That Cover the Conceptual Model of the Database, Natural Language Processing and Information Systems. In: Proceedings of the NLDB Conference, Sweden, pp. 235–239 (2002)
7. Berger, A., Caruana, R., Cohn, D., Freitag, D., Mittal, V.: Bridging the Lexical Chasm: Statistical Approaches To Answer-finding. In: Proceedings of ACM SIGIR Conference, New York, pp. 192–199 (2000)
8. GIZA++: Training of statistical translation models (2010),  
<http://fjoch.com/GIZA++.html>
9. Jeon, J., Croft, W.B., Lee, J.H.: Finding Semantically Similar Questions Based on Their Answers. In: Proceedings of the 28th ACM SIGIR Conference, Salvador, Brazil (2005)
10. Jeon, J., Croft, W.B., Lee, J.H.: Finding Similar Questions in Large Question and Answer Archives. In: Proceedings of the 14th CIKM, pp. 84–90 (2005)
11. Kosseim, L., Yousefi, J.: Improving the Performance of Question Answering With Semantically Equivalent Answer Patterns. *Journal of Data & Knowledge Engineering* 66, 57–67 (2008)
12. Mark, A.G., Horacio, S.: A Pattern Based Approach to Answering Factoid, List and Definition Questions. In: Proceedings of the 7th RIAO Conference, Avignon, France (2004)
13. OpenNLP (2010), <http://opennlp.sourceforge.net/>
14. Ravichandran, D., Hovy, E.: Learning Surface Text Patterns for a Question Answering System. In: Proceedings of the 40th ACL Conference, Philadelphia (2002)
15. Bernhard, D., Gurevych, I.: Answering Learners' Questions by Retrieving Question Paraphrases from Social Q&A Sites. In: Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications, pp. 44–52 (2008)
16. Term frequency/Inverse document frequency implementation in C# (2011),  
<http://www.codeproject.com/KB/cs/tfidf.aspx>
17. Bian, J., Liu, Y., Agichtein, E., Zha, H.: Finding the Right Facts in the Crowd: Factoid Question Answering Over Social Media. In: Proceedings of WWW Conference (2008)
18. Ion, M.: Extraction Patterns for Information Extraction Tasks: a Survey. In: Workshop on Machine Learning for Information Extraction, Orlando (1999)
19. Hao, T.Y., Hu, D.W., Liu, W.Y., Zeng, Q.T.: Semantic Patterns for User-interactive Question Answering. *Journal of Concurrency and Computation-practice & Experience* 20(7), 783–799 (2008)
20. Hu, D.W., Liu, W.Y.: SIIPU\*S: A Semantic Pattern Learning Algorithm. In: Proceedings of the SKG Conference, Guilin, China (2006)
21. Wu, C.H., Yeh, J.F., Chen, M.J.: Domain-specific FAQ Retrieval Using Independent Aspects. *Journal of ACM Transactions on Asian Language Information Processing* 4(1), 1–17 (2005)
22. Zhang, D., Lee, W.S.: Web based Pattern Mining and Matching Approach to Question Answering. In: Proceedings of TREC-10 (2001)
23. Jijkoun, V., Rijke, M.D.: Retrieving Answers From Frequently Asked Questions Pages on the Web. In: Proceedings of the 14th CIKM Conference, Bremen, Germany (2005)
24. Wang, K., Ming, Z., Chua, T.S.: A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based QA Services. In: Proceedings of SIGIR Conference, pp. 187–194 (2009)

# How to Answer Yes/No Spatial Questions Using Qualitative Reasoning?

Marcin Walas

Faculty of Mathematics and Computer Science,  
Adam Mickiewicz University, Poznań, Poland  
[mwalas@amu.edu.pl](mailto:mwalas@amu.edu.pl)

**Abstract.** We present a method of answering yes/no spatial questions for the purpose of the open-domain Polish question answering system based on news texts. We focus on questions which refer to certain qualitative spatial relation (e.g. Was Baruch Lumet born in the United States?). In order to answer such questions we apply qualitative spatial reasoning to our state-of-art question processing mechanisms. We use Region Connection Calculus (namely RCC-5) in the process of reasoning. In this paper we describe our algorithm that finds the answer to yes/no spatial questions. We propose a method for the evaluation of the algorithm and report results we obtained for a self-made questions set. Finally, we give some suggestions for possible extensions of our methods.

## 1 Question Answering Task

The goal of Questions Answering (QA) is to provide the answer to a question posed in the natural language. QA can obtain answers from the variety of data sources. Open Domain QA is aimed at processing a wide domain of questions (as opposed to Closed Domain QA which focuses on a specific topic).

We aim at developing an open domain QA system for the Polish language, which retrieves answers from the collection of news texts. We mainly focus on shallow methods, which do not require deep language analysis (e.g. semantic analysis). However, in order to process spatial questions we use some more sophisticated techniques, namely automatic reasoning.

In this paper we describe a method for finding the answer to a yes/no spatial question. Yes/No questions (or polar questions) are questions with two answers possible: yes or no (e.g. *Was Baruch Lumet born in Poland?*). We define a yes/no spatial question as a yes/no question which refers to some spatial relation (e.g. is something located in a particular place). The method has been implemented in our QA system prototype Hipisek.pl ([www.hipisek.pl](http://www.hipisek.pl)).

### 1.1 Previous Work

There are several approaches to the problem of answering yes/no questions. The Webclopedia project is an example of the open domain QA system for

the English language (see: [1]). In Webclopedia questions are represented in structures composed from **QTargets** and **QArgs**. **QTargets** can be described as an expected information type which should be provided in the answer. **QArgs** are the arguments of the **QTarget** (e.g. important parts of the question such as named entities). The system extracts answers by matching potential answer-bearing text fragments to instantiated **QTargets**.

Another approach consists in incorporating a reasoning system into the QA system. Such approach is taken in the SHAKEN system described in [2]. SHAKEN is a system for knowledge entry through the graphical assembly of concepts. One of its functionalities is to answer questions posed by experts. The mechanism for answer extraction includes application of the RCC-8 calculus and the Cardinal Calculus (which both are constraint calculi). The answer is found using path-consistency algorithm. The system belongs to the closed-domain QA category. SHAKEN was developed for the English language.

There exist a few QA systems for the Polish language. [3] describes the POLINT-112-SMS system. One of its components is a QA module. In the process of answering spatial questions (including yes/no questions) the system uses constraint reasoning (namely Cardinal Calculus). The system belongs to the closed-domain category, since it focuses on a narrow domain of topics, concerning public safety. For the authors knowledge there exists no open-domain QA system for the Polish language with similar capabilities for answering yes/no questions, as those implemented in our project.

The problem of answering yes/no questions can be reformulated as follows. Assuming that we are able to identify documents from which the answer should be extracted we can treat this problem as a sort of Natural Language Inference (NLI). NLI consists in determining whether a natural language hypothesis  $h$  can be reasonably inferred from a natural language premise  $p$  [4]. There are several approaches to NLI including shallow approaches (e.g. by using pattern based extraction or lexical overlap) and deep approaches (e.g. by using full semantic analysis and performing automated reasoning). In [5] it was showed that QA systems can benefit from the NLI incorporation.

## 1.2 Hipisek.pl Project Overview

In our project we incorporated NLI techniques into the QA system by treating the potential answer-bearing documents set as the set of premises and the question under consideration as the hypothesis.

We define three classes of the potential answer: YES, NO and UNKNOWN. We additionally require that the system extracts an explanation for the given answer (e.g. by showing a part of text from which the answer was extracted). In order to extract the answer the question is transformed into **QQuery** entity (which is similar to the notion of **QTarget** in the Webclopedia project).

In the **baseline version** of our system a method of answering a yes/no question was based on shallow NLI algorithms described in [4]. It includes lexical overlapping with weighting (e.g. named entities have higher weight in the process of weighting).

We process the question in the following steps:

1. Transformation into QQuery representation.
2. Retrieval of the documents and paragraphs (short extracts), which are likely to contain an answer for the question.
3. Answer extraction from the retrieved documents/paragraphs set.

One of the QQuery's elements is a question topic. **Question topic** is the main subject of the question (we assume that the question is a single non-contextual question, hence it contains only one topic). The question is transformed into QQuery representation using self-made rules set and a set of heuristics. The full description of the QQuery representation is given in [6].

In order to retrieve documents and paragraphs we use information retrieval techniques such as transformation of the question to a search engine query or lexical overlapping (similar to those described in [1]).

QQuery representation with a set of paragraphs form the input for the yes/no answering module.

### 1.3 The Notion of Space in Yes/No Questions

We observed that naive methods are insufficient for handling spatial questions. This is due to the fact that the process of answering a question related to some spatial entity depends on information which is not present directly in the source text. For example, consider the following part of the news article:

Tulips named after Maria Kaczyńska, [...] will be placed in the tomb of the Presidential Couple in the Wawel castle.

One can ask the following question: *Is the tomb located in Kraków?* (for which the answer is “Yes”). The system has to determinate if the Wawel castle is located in the city of Kraków. Certainly this information is not present in the article extract, although it is obvious for the Polish citizen.

As a solution we propose the methodology of spatial reasoning. We use Region Connection Calculus in the process of answer extraction. For our purposes we have chosen the RCC-5 calculus which allows for five relations between regions. The idea is as follows: we extract a spatial relation from a question and check whether it is consistent with those extracted from the article's text. The answer is “Yes” if there are no identified inconsistencies or “No” otherwise. We use qualitative spatial knowledge database to maintain *naive* knowledge.

The paper is organized as follows: first we describe our approach to spatial knowledge representation. Next we give a short overview of our QA system and the input data for the answering algorithms. We describe algorithms and the application of the reasoning module. Next we propose a method for evaluation of algorithms and present results. Finally we formulate conclusions and discuss further work.

## 2 Maintaining Spatial Knowledge

In the section we briefly describe knowledge database and its representation. We present RCC-5 as a method for spatial reasoning in our system. We discuss the consistency issue in RCC-5. Finally we present our reasoning module.

### 2.1 RCC-5 Calculus

RCC is a topological approach to qualitative spatial representation based on a simple primitive relation of connection between regions [7]. Relationships are defined by means of the  $C(a, b)$  relation (*connection relation*) which holds iff regions  $a$  and  $b$  share the common point. On the basis of relation  $C$  five base relations for RCC-5 are defined, namely:  $DR$  (*discrete*),  $EQ$  (*equal*),  $PP$  (*proper part*),  $PPI$  (*proper part inverse*) and  $PO$  (*partial overlap*) [8].

Any subset of a base relation set forms a relation in RCC-5. Thus RCC-5 contains  $2^5$  relations. A relation which contains all base relations is called the **universal relation**. The empty set of relations is called the **empty relation**. Following operations for the relations are possible: union, intersection, inversion and composition. RCC-5 relations are closed under composition and form a relation algebra [9]. A special compositions table is used to compute composition in RCC-5.

### 2.2 Reasoning with RCC-5

Using RCC-5 we can represent knowledge about entities in the form of constraints. A **constraint network** is formed from a set of variables  $V$  over the domain  $D$  and a set of constraints on the variables of  $V$ . A network is **consistent** if it has a solution which is an assignment of values of  $D$  to the variables of  $V$  in a way that all constraints are satisfied [9].

A path-consistency serves as a common approximation of the constraint satisfaction problem. A given constraint network is **path-consistent** iff for any consistent instantiation of any two variables there exists an instantiation of any third variable such that all three variables taken altogether are consistent. To enforce path-consistency, the following operation is used for all constraints between vertices  $i$  and  $j$  [9]:

$$\forall_k R_{ij} \Leftarrow R_{ij} \cap (R_{ik} \circ R_{kj})$$

The operation is used until a fixed point is reached. If an empty set results from the operation, then the network is not path-consistent. Otherwise it is path-consistent. In the general case path-consistency does not imply consistency. However, if the network is not path-consistent, it is not consistent either.

### 2.3 Knowledge Database

Spatial knowledge may be represented either quantitatively or qualitatively. The former way (e.g. by using absolute geographical coordinates) is certainly useful

in most engineering applications. However, for the sake of QA, the qualitative representation - storing relationships between spatial entities - is by far more useful, as this is how spatial information is represented in natural language [10].

We store entities and facts in the knowledge database. An **entity** is a notion of an individual entity in the real or abstract world. One entity belongs to exactly one **type**. A **fact** (or a **predicate**) is a triple: subject entity, predicate name and object entity. Predicate name is a name of the relation that holds between subject and object. Predicates are typed to form a taxonomy. An example of the fact that *Kraków is located in Poland* is a triple: (*city Kraków*, *is located in*, *country Poland*).

We acquired a comprehensive qualitative spatial knowledge database by integration of several on-line data sources. Our knowledge database contains 300 000 facts concerning objects such as: world cities, countries, rivers, lakes, mountains, forests, famous buildings, administrative divisions and touristic attractions. It is used as a naive knowledge data source. Detailed description of the knowledge database and its acquisition is given in [11].

## 2.4 Reasoning Module

In order to reason about space we create a **constraint network**. A constraint network is created for the input fact  $f$  (for which we want to check consistency) and a set of additional knowledge facts  $F_e$  (which may be obtained during answer extraction process).

In the first step of reasoning we add all facts from  $F_e$  to the network and the subject entity of  $f$  (but not the  $f$  itself). Our aim is to obtain the object of  $f$  with constraint equal to spatial predicate of  $f$  by adding a fact from the spatial knowledge database (which introduce new entities in the network).

We make the following assumptions in the reasoning process:

- If we obtain an edge in the network, whose constraint is exactly equal to the constraint corresponding to spatial relation of  $f$  (e.g. PP relation is equal to *is located in* relation), then we assume that  $f$  is TRUE.
- If we obtain an inconsistent network, then we assume that  $f$  is FALSE.
- Otherwise we assume that  $f$  is UNKNOWN.

In each algorithm's step we retrieve all facts from the knowledge database, which correspond to the entities from the current network and we add retrieved facts to the network. The process is repeated until the network is stable.

We check if the obtained network contains an object of  $f$ . If it does then we run path-consistency algorithm on the obtained constraint network. If the network is not path-consistent then we return that  $f$  is FALSE. Otherwise we check if the constraint obtained between subject of  $f$  and object of  $f$  is equal to the input constraint of  $f$ . If it is, then we assume that the fact  $f$  is TRUE.

If the network does not contain the object of  $f$ , then we add  $f$  to the network and run path-consistency algorithm once more. If the network with  $f$  is inconsistent then the result is  $f$  is FALSE. Otherwise  $f$  is UNKNOWN. For details of the reasoning process see: [11].

For example, suppose we want to decide if the fact  $f$ : (castle Wawel, is located in, country India) is true. An initial constraint network is created with only two vertices: castle Wawel and country India. The network is expanded using knowledge database and the following network is obtained:

```
castle Wawel --PP--> city Krakow --PP--> country Poland
```

The entity: country India was not reached. Hence the fact  $f$  is added to the network. The following network is obtained:

```
castle Wawel
  ' '--PP--> city Krakow
  '           '--PP--> country Poland
  '           '----DR ---->
  '-----PP----> country India
```

Note that DR relation between countries was inferred from semantics (see: [11]). This network is not path-consistent, hence  $p$  is FALSE.

### 3 Answering Algorithm

We divided yes/no questions into two groups:

- Questions which can be represented as a predicate (e.g. the question: *Is Paris located in France?*, can be represented as a predicate: (city Paris, is located in, country France)). Answering such a question is equal to proving that the predicate is true (or false). Check predicates are the model for the processed question.
- Questions which are represented as statements with additional constraints (e.g. the question: *Did Eric die in Africa?*, can be represented as a question: *Did Eric die?* and a constraint (\*, is located in, continent Africa)). Answering such a question is equal to finding an answer for the statement and then checking whether the constraints are satisfied.

We introduce two properties to the QQuery representation:

- **Check predicates set** – which corresponds to the first group of the questions. All check predicates have to be proven in order to answer the question.
- **Constraint predicates set** – which corresponds to the second group of the questions. Constraint predicates have either subject or object undefined (marked as an asterisk in the example above). All constraints have to be satisfied to accept the answer.

The difference between check and constraint predicates is that a check predicate is a full representation of the question, where constraint predicates are only conditions, which have to be fulfilled by the extracted answer.

### 3.1 Answering a Question with the Check Predicates Set

Predicate checker processes each of QQuery's check predicates.

We assume that the answer is:

- YES if any of check predicates is true,
- NO if any of check predicates is false.
- UNKNOWN otherwise

The answer is **defined** if its either YES or NO. Otherwise it is **undefined**. Further question processing is suppressed if a defined answer is obtained.

Each of the check predicates  $p$  from the given QQuery with a set of paragraphs  $P$  is proven using the following algorithm:

1. Try to find predicate  $p$  in the naive knowledge database. If it is in the database then return YES and STOP.
2. Use the reasoning module on a single  $p$  predicate and create the constraint network using knowledge database. If the returned answer is defined then STOP.
3. Extract all predicates from paragraphs  $P$  which have equal subject or object as one of the subject or object of  $p$ .
4. For each extracted predicate  $p_e$  from  $P$  do
  - (a) Use the reasoning module on  $p$  as an input predicate and  $p_e$  as an additional knowledge predicate. Create a constraint network using knowledge database.
  - (b) If the reasoning module returns a defined answer then STOP.
5. If an undefined answer is obtained then return UNKNOWN.

Note that the constraint network, which was obtained during reasoning process, can serve as an explanation for the answer returned by the system.

If no answer is obtained then we move to the second answering mechanism, which uses constraint predicates.

Consider the following example: User poses questions which concern the following article extract:

Tulips named after Maria Kaczyńska, [...] will be placed in the tomb of the Presidential Couple in the Wawel castle.

The first question is: *Is Wawel located in Poland?*. The system processes the question and transforms it into the *check predicate* structure: (castle Wawel, is located in, country Poland). We perform *check predicate answering process* presented above. The system lacks the information of the desired predicate in the naive knowledge database (point 1 of the algorithm), hence it tries to use reasoning mechanisms. From the naive knowledge database the system retrieves facts that: (castle Wawel, is located in, city Kraków) and (city Kraków, is located in, country Poland). The following constraint network is created:

```
Wawel -- PP --> Krakow -- PP --> Poland
`----- PP ----->
```

The network is path-consistent. We obtained the PP relation between *Wawel* and *Poland* vertices which is equal to *is located in* predicate. So the returned answer is: Yes.

A second exemplary question is: *Is the tomb located in Warsaw?*. The system transforms the question into the following *check predicate* structure: (*entity tomb, is located in, city Warsaw*). Notice that the *tomb* entity has only general entity type identified. We perform *check predicate answering process*. The first two steps of the algorithm fail. Next the system extracts the fact that: (*entity tomb, is located in, castle Wawel*). It uses this fact to obtain the following constraint network:

```
the tomb -- PP --> Wawel -- PP --> Krakow
`----- PP ----->
```

The reasoning module did not prove the check predicate (the desired predicate was not obtained in the process, see section: 2.4), hence it adds check predicate to the network. The following network is obtained:

```
the tomb
  '--PP--> Wawel
  '--PP--> Krakow
  '-----DR --->
  '-----PP----> Warsaw
```

Adding the check predicate fact to this network (*the tomb PP Warsaw*) leads to network inconsistency (since two cities are discrete, DR relation was inferred from semantics). Hence the system returns the answer: No.

### 3.2 Answering a Question with the Constraint Predicates Set

This method uses our baseline yes/no question answering module, which is based on the shallow NLI methods described in [4]. The module returns candidate answers with source sentences attached. Source sentences are sentences from the paragraphs which prove the answer chosen by the system. Candidate answers are verified by our constraint predicate verifier algorithm.

Constraint predicate verifier tries to satisfy all constraint predicates for the given candidate answer. We assume that the constraint predicate is satisfied if it is either TRUE or FALSE with reasoning using predicates extracted from the source sentence and its neighborhood (remark that falsification also satisfies constraint).

Deciding whether a given constraint predicate  $c$  is satisfied by the source sentence of the candidate answer  $s_a$  for the QQuery  $q$  is carried out using the following algorithm (we call this procedure *constraints verification procedure*):

1. Extract all predicates from  $s_a$ .
2. For each of the extracted predicates  $p_s$  do:
  - (a) Attach constraint predicate  $c$  to  $p_s$  (link undefined subject/object of  $c$  with the corresponding entity of  $p_s$ ).
  - (b) If constraint  $c$  equals predicate  $p_s$  then return YES and STOP
  - (c) Otherwise use the reasoning module on  $c$  with  $p_s$  as an additional knowledge predicate. Create a constraint network and check its path-consistency.
  - (d) If the result is obtained then return it and STOP
3. If no result is obtained then return UNKNOWN (constraint is not satisfied).

Using the verification result we obtain the final answer value:

- If the verification is **positive** (all constraints are satisfied with YES result), then the answer is left unchanged.
- If the verification is **negative** (at least one constraint is satisfied with NO result and all of them are defined), then the initial answer is negated.
- If the verification is not satisfied (at least one constraint verification process returned UNKNOWN), then the answer is removed from the candidate set (we are unable to verify is it a correct answer or not, hence it will not be shown as a result).

Consider the following example: the user asks a question (concerning above mentioned text extract): *Will tulips be placed in Kraków?* The system transforms the question to a QQuery representation with the topic: *tulip* and one constraint predicate: (\*, *is located in*, *city Kraków*) (note: an asterisk marks undefined subject). The baseline yes/no answerer extracts the answer *Yes* as a candidate answer (mostly due to the lexical overlapping for the topic of the question).

Now we perform constraint verification. We have one constraint to satisfy. The subject is attached to the constraint predicate obtaining constraint  $c$  equal to (*entity tulip*, *is located in*, *city Kraków*). Next the system extracts the following predicate from the input paragraph: (*entity tulip*, *is located in*, *castle Wawel*). For this predicate the following constraint network is created:

```
tulip -- PP --> Wawel -- PP --> Krakow
      '----- PP ----->
```

The network is path-consistent. The constraint  $c$  was obtained during reasoning so it is satisfied with the TRUE value. The verification is positive, so the candidate answer is left unchanged: Yes.

## 4 Evaluation

Evaluation of question answering systems is a complex task. Several features of the QA systems can be considered in the evaluation process, such as: query language difficulty, content language difficulty, question difficulty, usability, accuracy, confidence, speed and broad domain [12]. Moreover evaluation of the

specific method leads to additional problems such as incompleteness of the indexed articles database (one can ask a question for which the answer is not present in any of the indexed articles).

In order to evaluate effectiveness of our algorithm we carried out an experiment which involved the creation of the testing corpus. Due to the lack of the QA system for the Polish language with similar capabilities as our QA project, we were unable to compare our results with the other systems.

#### 4.1 Obtaining Data for the Evaluation

We decided to create a corpus of the questions from the predefined set of documents. From our knowledge database we randomly chose 40 articles. Each article was given to a tester, whose task was to create questions considering the article. The tester was instructed to pose only questions which contained a spatial relation and were the yes/no questions (still we did not define what the spatial relation is, leaving it to the tester's intuition). Each tester created a list which consists of questions, their correct answer and documents' ids from which the question was extracted.

#### 4.2 Experiments and Results

We prepared two versions of our system:

- **BASE** – without the algorithms for reasoning (only the baseline version of yes/no questions answering was turned on);
- **FINAL** – with all algorithms turned on.

We carried out evaluation in two experiments:

- In *semi-supervised answering* experiment, only the answering module was run. The input to the system was a question with the corresponding document id, from which the answer should be extracted.
- In *full answering* experiment, the answering module was run as a part of the full answering process. An additional step of question processing was carried out by the system, which was a retrieval of the relevant documents, from which an answer could be obtained.

The first experiment checks whether the system is able to extract correct answer assuming that the document from which the answer should be extracted was properly retrieved (since the document was given as an input to the system). The second experiment checks the full answering process.

We computed *precision* as a fraction of the correct answers among all for which any answer was returned and *recall* as a fraction of the correct answers among all answers in the test set.

The first experiment is focused on evaluation of the answering process *in the sandbox*. This task is similar to the task of the NLI. This follows from the observation, that the document which is given as an answer source can be treated as a set of premises to the hypothesis included in the question.

**Table 1.** Results of evaluation experiments

	Base	Supervised	Final	Supervised	Base	Full	Final	Full
# of all questions	640		640	640	640		640	
# of correct	218		248	127		206		
# of processed questions	359		335	419		396		
Precision	0.607		0.740	0.233		0.520		
Recall	0.341		0.387	0.198		0.322		
F-score	0.436		0.509	0.214		0.398		

The second method measures the effectiveness of the developed method in the working QA system prototype. In fact results obtained in the second experiment can be misleading. When the testers prepared the test set of questions, they often assumed some context of the question posed (e.g. by using only the first name of the person, or using ambiguous phrases like: *the president* – the president of what?). Hence there exists a significant number of questions for which the expected answer can differ from the answer obtained by the system, but still both may be correct. To avoid this problem we required each answer provided by the system to be checked by the researcher (expected answers given by testers were not used). The researcher checked the answer and its explanation (e.g. paragraph retrieved or constraint network). Results of both experiments are given in Table 1.

Experiments show that our reasoning module increases precision of the system. In both experiments precision increased due to application of the reasoning module. As a consequence total number of the processed questions decreases (a processed question is a question for which a defined answer was returned). We report an increase in F-score for the system in both experiments.

Full experiment results show that the context is important issue in the process of evaluation. As expected, some answers were marked as incorrect in the automatic evaluation, but they were in fact correct. On the other hand, there were some questions which were marked as incorrect by the researcher due to bad explanation provided by the system.

## 5 Conclusions and Further Work

In the paper we presented a method for finding an answer for yes/no spatial questions. We applied automatic reasoning module into the state-of-art answering mechanism.

We plan to extend our method to process temporal questions. This goal can be achieved by means of application of a temporal reasoning mechanism (for which we plan to use Allen's calculus). The algorithm will be also used in other answering modules of our system (which process other types of questions, e.g. where?, when?, who? questions). It will be used to identify answers (for any spatial questions) that contradict.

## References

1. Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M., Lin, C.Y.: Question answering in webclopedia. In: TREC (2000)
2. Uribe, T.E., Chaudhri, V., Hayes, P.J., Stickel, M.E.: Qualitative spatial reasoning for question-answering: Axiom reuse and algebraic methods. In: AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases. AAAI, Stanford (2002)
3. Vetulani, Z., Marciniak, J., Obrebski, J., Vetulani, G., Dabrowski, A., Kubis, M., Osiński, J., Walkowska, J., Kubacki, P., Witalewski, K.: Language resources and text processing technologies. In: The POLINT-112-SMS System as Example of Application of Human Language Technology in the Public Security Area. Adam Mickiewicz University Press (2010)
4. MacCartney, B.: Natural language inference, Ph. D. Dissertation (2009)
5. Harabagiu, A., Hickl, A., Lacatusu, F.: Negation, contrast and contradiction in text processing. In: Proceedings of AAAI 2006 (2006)
6. Walas, M., Jassem, K.: Named entity recognition in a polish question answering system. In: Kopotek, M.A., et al. (eds.) Intelligent Information Systems, pp. 181–192. Publishing House of University of Podlasie (2010)
7. Randell, D.A., Cui, Z., Cohn, A.G.: A Spatial Logic based on Regions and Connection. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning (1992)
8. Bennett, B.: Spatial reasoning with propositional logics. In: Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR 1994), pp. 51–62. Morgan Kaufmann (1994)
9. Renz, J.: Qualitative Spatial Reasoning with Topological Information. LNCS (LNAI), vol. 2293. Springer, Heidelberg (2002)
10. Renz, J., Rauh, R., Knauff, M.: Towards Cognitive Adequacy of Topological Spatial Relations. In: Habel, C., Brauer, W., Freksa, C., Wender, K.F. (eds.) Spatial Cognition 2000. LNCS (LNAI), vol. 1849, pp. 184–197. Springer, Heidelberg (2000)
11. Walas, M., Jassem, K.: Spatial reasoning and disambiguation in the process of knowledge acquisition. In: Vetulani, Z. (ed.) Proceeding of the 5th Language and Technology Conference, pp. 420–424 (2011)
12. Ferrucci, D., Nyberg, E., Allen, J., Barker, K., Brown, E.W., Chu-Carroll, J., Ciccolo, A., Duboue, P.A., Fan, J., Gondek, D., Hovy, E., Katz, B., Lally, A., McCord, M., Morarescu, P., Murdock, J.W., Porter, B., Prager, J.M., Strzalkowski, T., Welty, C., Zadrozny, W.: Towards the open advancement of question answering systems. Technical Report RC24789, IBM Research, Hawthorne, NY (2008)

# Question Answering and Multi-search Engines in Geo-Temporal Information Retrieval

Fernando S. Peregrino, David Tomás, and Fernando Llopis Pascual

Department of Software and Computing Systems, University of Alicante  
Carretera San Vicente del Raspeig s/n - 03690 Alicante, Spain

**Abstract.** In this paper we present a complete system for the treatment of both geographical and temporal dimensions in text and its application to information retrieval. This system has been evaluated in both the *GeoTime* task of the 8th and 9th *NTCIR* workshop in the years 2010 and 2011 respectively, making it possible to compare the system to contemporary approaches to the topic. In order to participate in this task we have added the temporal dimension to our *GIR* system. The system proposed here has a modular architecture in order to add or modify features. In the development of this system, we have followed a *QA*-based approach as well as multi-search engines to improve the system performance.

**Keywords:** Geographical Information Retrieval, Geo-Tagging, Spacial Information, Temporal Information.

## 1 Introduction

Information retrieval (*IR*) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)[1].

*GIR* is a specialization of *IR* with geographic metadata associated. *IR* systems usually see the documents as a collection or “bag of words”. By contrast, *GIR* systems require semantic information, i.e. they need a place name or geographical feature associated with the document. Because of this, in *GIR* systems, it is common to separate the analysis and text indexing from the geographic indexing.

Temporal information is available in every document either explicitly, i.e., in the form of temporal expressions, or implicitly in the form of metadata. Recognizing such information and exploiting it for document retrieval and presentation purposes are important features that can significantly improve the functionality of search applications. Temporal Information Retrieval (*TIR*), analogously to *GIR*, is a specialization of Information Retrieval with temporal metadata associated.

The objective of this work is to adopt a first approach in the geo-temporal *IR* field, including the observation of how a basic *IR* system can be improved by embedding geo-temporal *IR* intelligence, and to identify what methods used in them have a better performance.

We have evaluated this approach according to the *GeoTime* task included in both the *NTCIR-8* and *NTCIR-9*<sup>1</sup> workshop. *GeoTime* for the *NTCIR Workshop* is an evaluation of Geographic and Temporal Information Retrieval “*NTCIR GeoTime*”. The focus of this task is on searching with Geographic and Temporal constraints[2].

To that end, we have elaborated this paper to be structured as follow: In section 2, we provide a general description of the system, describing the topic storage architecture as well as the system operation. Subsequently, section 3 will outline the experiments and evaluations conducted. Finally, in section 4, we describe the conclusions and future work in this area.

## 2 System Description

For the creation of this *Geo-Temporal IR* system, we have chosen to implement it in a modular fashion with the intention of adding new components, testing and improving the existing ones.

Figure 1 shows the architecture of our system, its component modules and the data flow. This system works in three different phases: the first phase is represented by the solid lines which show the data flow that takes place in preprocessing time. On the other hand, broken lines represent the data flow which takes place in execution time. The second phase is represented by the thicker broken lines, those that process the topic, and the third phase is outlined by thinner broken lines, those which execute the query.

### 2.1 System Operation

As it was mentioned above, the system operation is divided into three phases: pre-processing and indexing the corpus, processing queries, and running queries.

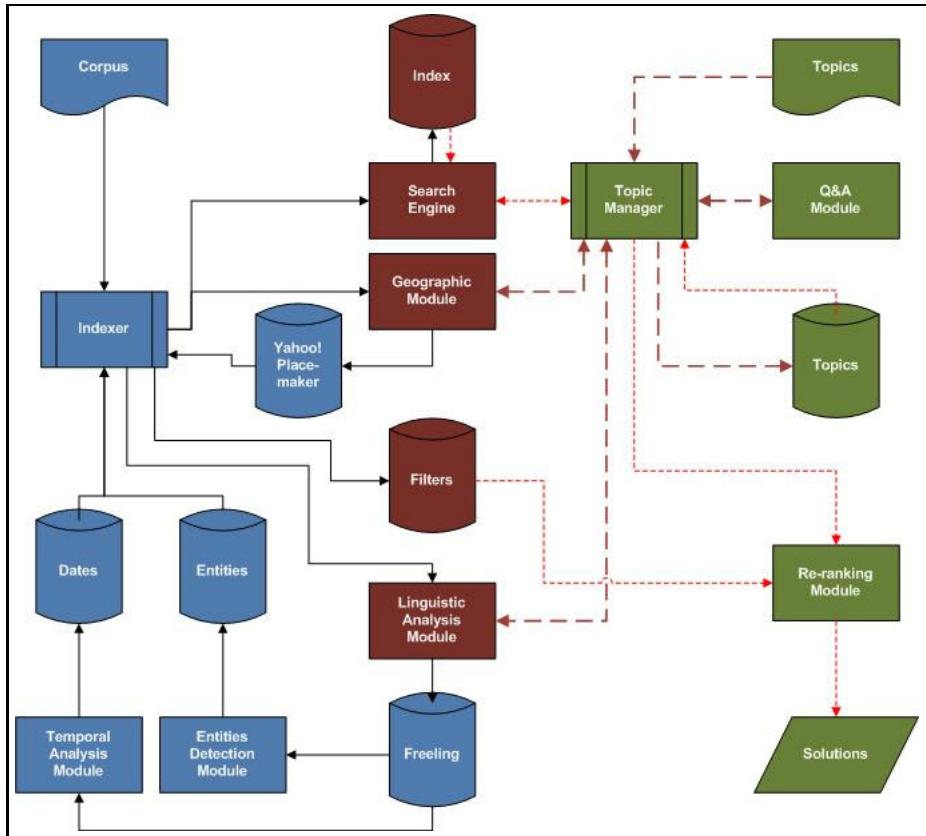
**Corpus Pre-process.** Firstly, in this phase the lemmatized corpus is indexed in the search engine module. This module has two functionalities: to index the whole corpus, and to retrieve a set of relevant documents for a given query.

Initially, the search engine chosen for this system was *Lucene*<sup>2</sup>. We have included characteristics to this search engine, such as a stemming and stopword removal. The ranking function *Okapi BM25*[3] has been used to rank the results according to their relevance. Finally, it has been chosen to retrieve up to 1,000 relevant documents per query.

On the other hand, whilst the search engine is indexing, *Yahoo! Placemaker* obtains the geographic entities, and *FreeLing* gets the temporal expressions and the rest of named entities of the corpus. With all this information a new *XML* file is made for each corpus article. These *XML* documents will be useful to know the article relevance with respect to the query in the query runtime phase.

<sup>1</sup> <http://metadata.berkeley.edu/NTCIR-GeoTime/description.php>

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



**Fig. 1.** Diagram of the workflow in the *GIR* system implemented

**Query Process.** In this second phase, the topics are sent to the linguistic analysis module and to the *QA* module. Afterwards, our system sends every geographic reference obtained from the two previous modules to the geographic module in order to transform them to the *Yahoo!* unique identifier (*WOEID*). Finally, this data are stored, making a new *XML* file for each topic (this *XML* file is different to that one created for each corpus article). An example of this topic file can be seen in Figure 2, where the following sections can be observed:

- Search terms ( $<search>$ ): all search term without stopwords.
  - Lemmatized search terms ( $<search\_lemma>$ ): lemmatized search terms section.
  - Filters ( $<filters>$ ):
    - Descriptive part ( $<description>$ ): dates, place names, and entities found in the descriptive part of the query.
    - Narrative part ( $<narrative>$ ): Analogous to the previous one and, in addition, it has the geographical and temporal constraints.

```

<query id="GeoTime-0040">
  <search>Concorde crash</search>
  <search_lemma>concorde crash</search_lemma>
  <filters>
    <description>
      <entities>
        <item>concorde</item>
      </entities>
    </description>
    <entities>
      <item>concorde</item>
    </entities>
    <commons>
      <item>crash</item>
      <item>airliner</item>
    </commons>
    <narrative>
    </narrative>
  </filters>
  <yahoo>
    <dates>
      <item weight="1.0">[??-??/??/2000:??-??-??]</item>
      <item weight="0.08434785">[??-??/7/2000:??-??-??]</item>
      <item weight="0.08478205">[??-??/7/2000:??-??-??]</item>
      <item weight="0.15217301">[??-??/??/2002:??-??-??]</item>
      <item weight="0.17608695">[??-??/??/1976:??-??-??]</item>
      <item weight="0.17608695">[??-??/??/1969:??-??-??]</item>
      <item weight="0.065217301">[??-??/9/2001:??-??-??]</item>
      <item weight="0.065217301">[??-??/2/2010:??-??-??]</item>
      <item weight="0.043478205">[??-??/6/2000:??-??-??]</item>
      <item weight="0.043478205">[??-??/4/2003:??-??-??]</item>
    </dates>
    <dates_year>
      <item weight="1.0">[??-??/??/2000:??-??-??]</item>
      <item weight="0.08986595">[??-??/7/2000:??-??-??]</item>
      <item weight="0.059051415">[??-??/7/2001:??-??-??]</item>
      <item weight="0.059051415">[??-??/7/2001:??-??-??]</item>
      <item weight="0.055118115">[??-??/??/1969:??-??-??]</item>
      <item weight="0.039370005">[??-??/??/1976:??-??-??]</item>
      <item weight="0.0236226475">[??-??/2/2000:??-??-??]</item>
      <item weight="0.019685645">[??-??/??/2011:??-??-??]</item>
      <item weight="0.0078740165">[??-??/??/1985:??-??-??]</item>
      <item weight="0.0078740165">[??-??/??/1979:??-??-??]</item>
    </dates_year>
    <dates_month>
      <item weight="1.0">[??-??/7/2000:??-??-??]</item>
      <item weight="0.068490555">[??-??/7/2010:??-??-??]</item>
      <item weight="0.054794525">[??-??/8/2010:??-??-??]</item>
      <item weight="0.0479452055">[??-??/9/2009:??-??-??]</item>
      <item weight="0.0479452055">[??-??/10/2009:??-??-??]</item>
      <item weight="0.0479452055">[??-??/7/2001:??-??-??]</item>
      <item weight="0.034246575">[??-??/12/2010:??-??-??]</item>
      <item weight="0.027397205">[??-??/6/2011:??-??-??]</item>
      <item weight="0.027397205">[??-??/4/2003:??-??-??]</item>
    </dates_month>
    <locations>
      <item weight="1.0">615702</item>
      <item weight="0.493127114">23424819</item>
      <item weight="0.367697615">13130000</item>
      <item weight="0.131300005">13130000</item>
      <item weight="0.105841955">23424077</item>
      <item weight="0.079037875">44418</item>
      <item weight="0.066728525">2384019</item>
      <item weight="0.044673545">2384019</item>
      <item weight="0.0399278355">24865575</item>
      <item weight="0.027491415">2459115</item>
    </locations>
  </yahoo>
  </filters>
</query>

```

Fig. 2. XML topic document sample

- \* Query expanding (*<commons>*). It has expanded entries of the most representative terms of the query to a possible future query expansion.
- *QA* (*<yahoo>*): It has the following extracted data from *Yahoo!*: dates, year and month dates, year dates, and toponyms. It has normalized the 10 more representative values for all four piece of date aforementioned. This data is obtained from the module of *Question Answering* which tries to obtain from the web geographic and temporal expressions that are relevant to the query. The process to get the expressions is:
  1. The query is sent to *Yahoo! Search BOSS*<sup>3</sup>.
  2. *Yahoo! Search BOSS* collects the first 1,000 snippets from the returned results.

<sup>3</sup> *Yahoo! Search BOSS* (Build your Own Search Service) is *Yahoo!*'s open search and data services platform to build web-scale search products that utilize *Yahoo! Search* technology and data (<http://developer.yahoo.com/search/boss/>)

3. All dates and places from these snippets are then extracted. In order to do this task, the open source language analysis tool *FreeLing*<sup>4</sup> is used.
4. The total number of occurrences is computed and normalized, obtaining the 10 most relevant for each of the following categories:
  - (a) Completed or uncompleted dates (<dates>).
  - (b) month and year (<dates\_month>).
  - (c) year (<dates\_year>).
  - (d) place names (<locations>). *FreeLing* assigns the same label to both a place name and other named entities, and in order to distinguish between them, we use a list of toponyms obtained from *GeoNames*<sup>5</sup>. Once we have separate the grain from the chaff, the locations are sent to *Yahoo! PlaceMaker* to get the *WOEID*.

**Query Runtime.** In this third and last phase, the system sends the query, which is the content of the tag <*search\_lemma*> in the *XML* topic file (see Figure 2), to the search engine. The search engine returns 1,000 relevant ranked documents. The re-ranking module obtains the *XML* corpus files for each document returned by the search engine and this module re-ranks the documents matching the former rank from the search engine with the *XML* corpus, according to a weight function (the operation of this function is not going to be described here as this exceeds the scope of this paper).

### 3 Experimentation and Evaluation

In this section, on the one hand, we will describe both the metrics used to evaluate this system and the framework in which the evaluation was carried out. On the other hand, we will analyse the impact of the search engine and *QA* modules on the final results.

#### 3.1 Metrics and Evaluation Framework

In this section, it how the system has been evaluated will be shown and the choice of an evaluation metric will be reasoned.

**Evaluation Framework.** Firstly, this system was assessed with the document collection of the task *GeoTime* included in the *NTCIR 2010*, which can be seen in [4]. The English collection used in this task consisted of 315,417 *New York Times* stories for 2002-2005. Regarding topics, there were 25 which included geographical and temporal constraints, with both a descriptive and a narrative part (e.g. Descriptive part: “*When and where did a volcano erupt in Africa*

<sup>4</sup> <http://nlp.lsi.upc.edu/freeling/>

<sup>5</sup> <http://www.geonames.org/>

during 2002?”. Narrative part: “*The user would like to know the date in 2002 in which a volcano erupted in Africa. What was the name of the volcano and in which country is it located?*”).

Secondly, the system was assessed in the following year task, i.e. the *NTCIR 2011 GeoTime*, which was similar to the previous one, with 25 new topics, and adding three more corpora for 1998-2001: *Mainichi Daily*, *Korea Times* and *Xinhua English*, for a total of 797,216 articles which cover the period 1998 to 2005.

**Evaluation Metrics.** To assess the result of this geo-temporal *IR* system we have chosen one of the metrics used in *NTCIR-GeoTime*, the *nDCG*<sup>6</sup> (*normalized Discounted Cumulative Gain*) [5]. We have chosen this metric because it is capable of doing gradual assessments, it means that not only can it tag a document as a relevant or irrelevant, but it gives a relevance degree. At *NTCIR-GeoTime* this metric was used with three different bases: 10, 100, and 1,000. We have chosen the base 1,000 for this metric (the same as the number of documents that we retrieve for a topic), which means that the function is not taking into account the position of a relevant document, but whether this document is retrieved (*Cumulative Gain*). This has been done because we are focusing on obtaining the biggest percentage of relevant documents rather than in getting an accurate ranking, such as will be shown in future work which will be carried out in the rest of the modules of the system.

### 3.2 Impact of the Components

In the next sections, the impact that the search engine and the *QA* modules have in the system will be shown and how they can obtain a considerable improvement.

**Search Engine.** As mentioned in the Section 2.1, this system highly depends on the search engine performance and, therefore, the first experiment carried out dealt with this module. The experiment took place in the *NTCIR 2011* framework, and it was observed that the coverage achieved by *Lucene* was just 55.7892%, so that lead to an experiment to test what would had happened if it had reached a wider coverage, the results of which are shown in Table 1. These results have been classified into three groups:

1. Topics which get a recall between 0% and 100%, all of them.
2. Topics which get a recall between 50% and 100%, 12 out of 25.
3. Topics which get a recall between 75% and 100%, 10 out of 25.

---

<sup>6</sup> *nDCG* measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks.

In each of these three groups, the percentage of document recall by each topic, and the *nDCG-1000* score achieved for the query can be observed. Finally, the average recall and score for the topics that fall into each of the three groups is obtained. The objective of this experiment was to see what would happen if there had been more recall by the search engine module, and the substantial improvement that could have been achieved can be appreciated in the last two rows of Table 1 (from a score of 0.3959 to 0.5607 or 0.6081, according to the minimum recall required).

**Table 1.** Recall and *nDCG-1000* scores achieved using only *Lucene* for each *NTCIR 2011* topic

Topic	0% - 100%		50% - 100%		75% - 100%	
	Recall	nDCG	Recall	nDCG	Recall	nDCG
GeoTime-0026	93.2945%	0.7730	93.2945%	0.7730	93.2945%	0.7730
GeoTime-0027	85.7143%	0.2576	85.7143%	0.2576	85.7143%	0.2576
GeoTime-0028	85.4839%	0.5846	85.4839%	0.5846	85.4839%	0.5846
GeoTime-0029	43.3566%	0.2806	-	-	-	-
GeoTime-0030	66.6667%	0.3467	66.6667%	0.3467	-	-
GeoTime-0031	36.6667%	0.2905	-	-	-	-
GeoTime-0032	35.0877%	0.3367	-	-	-	-
GeoTime-0033	74.4186%	0.5660	74.4186%	0.5660	-	-
GeoTime-0034	86.3636%	0.4655	86.3636%	0.4655	86.3636%	0.4655
GeoTime-0035	28.5714%	0.1031	-	-	-	-
GeoTime-0036	31.9149%	0.2849	-	-	-	-
GeoTime-0037	0.0000%	0.0000	-	-	-	-
GeoTime-0038	1.6908%	0.0317	-	-	-	-
GeoTime-0039	84.1202%	0.6174	84.1202%	0.6174	84.1202%	0.6174
GeoTime-0040	82.0755%	0.7887	82.0755%	0.7887	82.0755%	0.7887
GeoTime-0041	98.9362%	0.7117	98.9362%	0.7117	98.9362%	0.7117
GeoTime-0042	1.2739%	0.0145	-	-	-	-
GeoTime-0043	91.4894%	0.5294	91.4894%	0.5294	91.4894%	0.5294
GeoTime-0044	28.5714%	0.1920	-	-	-	-
GeoTime-0045	75.0000%	0.6110	75.0000%	0.6110	75.0000%	0.6110
GeoTime-0046	92.3077%	0.7454	92.3077%	0.7454	92.3077%	0.7454
GeoTime-0047	6.6667%	0.0174	-	-	-	-
GeoTime-0048	47.9167%	0.4963	-	-	-	-
GeoTime-0049	60.0000%	0.6509	60.0000%	0.6509	-	-
GeoTime-0050	57.1429%	0.2031	57.1429%	0.2031	-	-
<b>Average Recall</b>	<b>55.3770%</b>		<b>80.9295%</b>		<b>87.4785%</b>	
<b>Average Score</b>	<b>0.3959</b>		<b>0.5607</b>		<b>0.6081</b>	

Given that the coverage obtained by *Lucene* barely reached 50%, as can be seen in the penultimate row of the Table 1, and based on the work done by [6], we decided to give the system an additional search engine, *Terrier*<sup>7</sup>. The

<sup>7</sup> <http://terrier.org/>

*Bose-Einstein (Bo1)* query expansion model has been added to *Terrier*. In order to obtain a final normalized score for each document returned by both search engines, it was done as follow for each topic:

1. The maximum *Lucene* score value is obtained among all documents returned by it.
2. All documents scores returned by *Lucene* are divided between the value indicated in the previous step.
3. Similarly, the previous two steps are repeated for *Terrier*.
4. If there are documents returned either by *Lucene* and *Terrier*, both scores must be added.
5. Finally, the score of each document returned by the search engines mentioned above is divided by two, thereby obtaining a normalized value between 0 and 1.

Using both search engines, the recall improved from 55.377% to 87.0165% (Table 2). This recall increases the *nDCG-1000* score of the system from 0.3959 to 0.5921 by using only the *IR* module of the system.

Although *Terrier* alone achieved a recall comparable to the combination with *Lucene*, employing both search engines provided an improvement in 7 out of 25 topics. In the case of topic 44, this improvement was clearly significant (from 28.5714% to 46.9387%). Thus, this combination of search engines offers a more robust approach in order to retrieve the relevant documents that will be employed in the rest of the modules of the *GIR* system [7].

**Question Answering Module.** We performed a study on this module and noted that the *XML* documents created after the treatment of the topics (see Figure 2), in the part concerning to this module, which operation has been explained in Section 2.1 in the page 345, in the vast majority of cases, the temporal and/or the geographical part of the query were answered. For this reason it was decided to carry out an experiment where the 10 terms from the dates section (*<dates>*), complete or incomplete ones, and the 10 terms from the place names section (*<locations>*), all of them with their respective weights (*weight*), were added to the query which is run on the *Lucene* search engine. Later, the documents retrieve by *Lucene* would be joined to the *Terrier* ones, as explained in the Section 3.2 in the search engines experiment mentioned in page 348. As a result of this experiment the *nDCG-1000* score was increased from 0.5921 to 0.6206.

**Table 2.** Recall achieved using two search engines (*Lucene* and *Terrier*) for each *NTCIR 2011* topic

Topic	Lucene	Terrier	Lucene+Terrier
GeoTime-0026	93.2944%	98.5422%	98.8338%
GeoTime-0027	85.7142%	100%	100%
GeoTime-0028	85.4838%	99.1935%	99.1935%
GeoTime-0029	43.3566%	87.4125%	90.2097%
GeoTime-0030	66.6667%	85.7142%	85.7142%
GeoTime-0031	36.6667%	86.6667%	86.6667%
GeoTime-0032	35.087%	89.4736%	89.4736%
GeoTime-0033	74.4186%	100%	100%
GeoTime-0034	86.3636%	95.4545%	95.4545%
GeoTime-0035	28.5714%	76.1904%	76.1904%
GeoTime-0036	31.9148%	91.489%	91.489%
GeoTime-0037	0%	2.8571%	2.8571%
GeoTime-0038	1.6908%	68.5990%	68.8405%
GeoTime-0039	84.1201%	98.7124%	98.7124%
GeoTime-0040	82.075%	99.0566%	99.0566%
GeoTime-0041	98.9361%	100%	100%
GeoTime-0042	1.2738%	87.261%	87.8980%
GeoTime-0043	91.489%	100%	100%
GeoTime-0044	28.5714%	28.5714%	46.9387%
GeoTime-0045	75%	100%	100%
GeoTime-0046	92.3076%	96.1538%	98.7179%
GeoTime-0047	6.6667%	80%	80%
GeoTime-0048	47.9167%	77.0833%	79.1667%
GeoTime-0049	60%	100%	100%
GeoTime-0050	57.1428%	100%	100%
<b>Average</b>	<b>55.3770%</b>	<b>86.1557%</b>	<b>87.4330%</b>

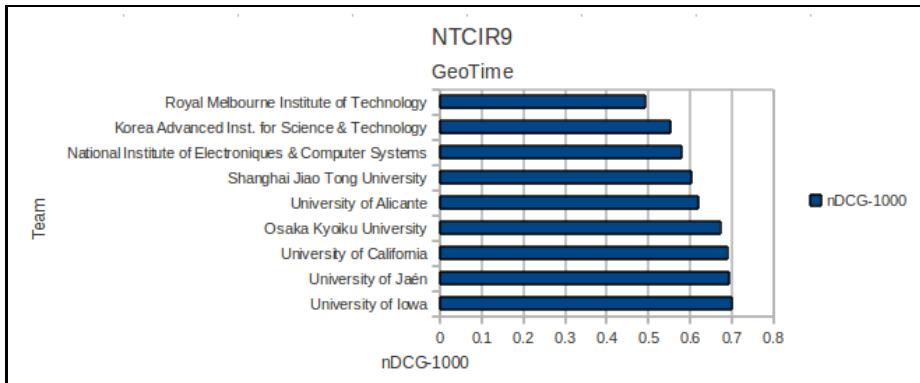
## 4 Conclusions

In this first approach to geo-temporal *IR* systems, we have started from a *IR* system and we have added geographical intelligence. In addition, we have used a naive implementation to tackle the temporal dimension. In spite of this, we can draw the following conclusions.

In the future, the linguistic analysis module should be improved to have the ability to extract and/or filter better the information from the narrative part of the topics. Despite this, our system (*University of Alicante*) with only two search engines and *QA* techniques is able to obtain outstanding scores in the *NTCIR 2011 GeoTime* task, such as can be seen in [2] and in Figure 3<sup>8</sup>.

As we have mentioned before, the *QA* module obtains a remarkable enrichment, therefore, we are exploring different *QA* techniques to use in the future.

<sup>8</sup> The scores from non completely automatic runs have been omitted.



**Fig. 3.** Best *NTCIR 9* teams score

In addition, given that good results were achieved by applying *QA* on *Lucene* query terms, as was seen in the Section 3.2 in the *QA* experiment in page 349, in a future experiment we will introduce the *QA* as *Terrier* query terms as well.

Focusing on the geographical module, currently we have two active fronts. On the one hand, we are exploiting more metadata from *Yahoo! Placemaker*, such as the general geographical scope of the document. On the other hand, we intend to fully develop the geographic module to be independent of applications which are subject to the restrictions of third parties.

Regarding Temporal Information Retrieval (*TIR*), a *TIR* system (*TIPSem*<sup>9</sup>) developed in our research group will be joined to this geo-temporal system in order to provide more temporal intelligence.

In future work, the usefulness of the rest of the components of the system such as the *entities detection module*, or the *Re-ranking module* will be analysed.

**Acknowledgments.** This research has been partially funded by the Spanish Government under project TEXTMESS 2.0 (TIN2009-13391-C04-01), and by the University of Alicante under project GRE10-33.

## References

1. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press (2009)
2. Gey, F., Larson, R.R., Machado, J., Yoshioka, M.: NTCIR9-GeoTime Overview - Evaluating Geographic and Temporal Search: Round 2 (2011)
3. Robertson, S.E., Walker, S., Hancock-Beaulieu, M.: Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive, pp. 199–210. NIST (1998)
4. Gey, F., Larson, R.R., Kando, N., Machado, J., Sakai, T.: NTCIR-GeoTime Overview: Evaluating Geographic and Temporal Search, pp. 147–153 (2010)

<sup>9</sup> <http://gplsi.dlsi.ua.es/demos/TIMEE/>

5. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.* 20, 422–446 (2002)
6. Perea-Ortega, J.M.: Recuperación de información geográfica basada en múltiples formulaciones y motores de búsqueda. *Procesamiento del Lenguaje Natural* 46, 131–132 (2011) ISSN 1135-5948
7. Peregrino, F.S., Tomás, D., Llopis, F.: University of Alicante at NTCIR-9 GeoTime. In: *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies* (2011)

# Using Graph Based Mapping of Co-occurring Words and Closeness Centrality Score for Summarization Evaluation

Niraj Kumar, Kannan Srinathan, and Vasudeva Varma

IIIT-Hyderabad, Hyderabad-500032, India

niraj\_kumar@research.iiit.ac.in, {srinathan,vv}@iiit.ac.in

**Abstract.** The use of predefined phrase patterns like: N-grams ( $N \geq 2$ ), longest common sub sequences or pre defined linguistic patterns etc do not give any credit to non-matching/smaller-size useful patterns and thus, may result in loss of information. Next, the use of 1-gram based model results in several noisy matches. Additionally, due to presence of more than one topic with different levels of importance in summary, we consider summarization evaluation task as topic based evaluation of information content. Means at first stage, we identify the topics covered in given model/reference summary and calculate their importance. At the next stage, we calculate the information coverage in test / machine generated summary, w.r.t. every identified topic. We introduce a graph based mapping scheme and the concept of closeness centrality measure to calculate the information depth and sense of the co-occurring words in every identified topic. Our experimental results show that devised system is better than/comparable with best results of TAC 2011 AESOP dataset.

**Keywords:** Summarization Evaluation, GAAC, Closeness Centrality, Sentence Clustering, AESOP.

## 1 Introduction

Human evaluation for text summarization is time consuming, costly, and prone to human variability [3]; [4]. Thus, it creates the demand of automatic evaluation of machine generated summary.

Evaluation of machine generated summaries has been of importance both in TAC (Text Analysis Conference) and previously DUC (Document Understanding Conference). The main goal is to produce two sets of numeric summary-level scores.

**(a) All Peers case:** a numeric score for each peer summary, including the model summaries. The "All Peers" case is intended to focus on whether an automatic metric can differentiate between human and automatic summarizers.

**(b) No Models case:** a numeric score for each peer summary, excluding the model summaries. The "No Models" case is intended to focus on how well an automatic metric can evaluate automatic summaries.

## 1.1 Related Work

Current state-of-the-art techniques such as manual pyramid scores [1] or automatic ROUGE metric (considers lexical N-grams as the unit for comparing the overlap between summaries [2]) use human summaries as reference.

[5], [6] proposed basic elements based methods (BE), it facilitates matching of expressive variants of syntactically well-formed units called Basic Elements (BEs). The ROUGE/BE toolkit has become the standard automatic method for evaluating the content of machine-generated summaries, but still there is a significant gap in quality between human evaluation and these automated metrics.

## 1.2 Problem Setup and Motivation

**Exploiting Topics Covered in Document:** It is important to note that summaries may contain more than one topics and each topic may have different levels of importance. The use of variability in the granularity of the analysis of summary for summary evaluation by [1] Pyramid method also support our view. We use the group average agglomerative clustering (GAAC) to cluster the sentences of document. To calculate the importance of identified clusters, we use page rank score of words on reverse directed word graph of sentences. The page rank score on reverse directed word graph of sentences effectively capture our writing strategies as, we describe the terms after writing it.

**Importance of Sense of Co-occurring Words.** The co-occurring words or sequences may have different role in model and machine generated summary. Thus, neglecting the sense of co-occurring words/ sequences may misguide the evaluation scheme. It will clear from following reference sentences, taken from [2].

- S1. Police killed the gunman.
- S2. Police kill the gunman.
- S3. The gunman kill the police.
- S4. The gunman kill police.
- S5. Gunman the killed police.

In these five sentences more than one word are common, but their roles are not always same. Now, let us analyze the problem:

- i. Suppose we take S1 as the reference and S2 and S3 as candidate summary sentences, then ROUGE-2 [2], gives same score to S2 and S3. This is just because, both sentences have common bigram “the gunman”. However, S2 and S3 have very different meaning.
- ii. To solve this (discussed above) problem ROUGE-S (Skip-Bigram co-occurrence statics) is used by [2]. But, the potential problem with ROUGE-S is that, it doesnot give any credit to a candidate sentences, if the sentence doesnot have any word pair co-occurring with its reference. i.e. it cannot properly handle the candidate sentence (S5). To solve this problem an extension of ROUGE-S is proposed (i.e. ROUGE-SU) by the addition of unigram as counting unit. Now, ROUGE-2 and ROUGE-SU4 is used as benchmark (By TAC “Text Analysis

Conference") in automatic evaluation of machine generated summary. But, we believe that there should be a single metric to handle all such issues.

**Information Loss due to Phrase Length Related Restriction:** as, discussed earlier, the predefined phrase patterns creates problem of information loss due to phrase length related restrictions. For example, ROUGE-L (which uses LCS "Longest Common Sub-Sequence"), suffers with this disadvantage. It only counts main in-sequence words, therefore, other LCSes and shorter sequences are not reflected in the final score. In the example, in sentence "S4", using "S1" as reference, LCS counts either "the gunman" or "police killed", but not both; therefore "S4" has same ROUGE-L score as "S3".

**Use of Closeness Centrality Measure:** In this paper we introduce the use of closeness centrality measure to:

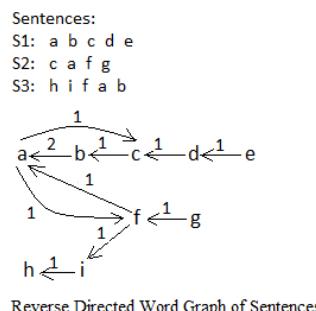
- Identify the sense of co-occurring words and
- Remove the chances of information loss due to fixed length sequences.

For this, we use word graph of sentences, which helps in maintaining the inter-word cohesion. Then we use closeness centrality measure, which helps in calculating the information propagation strength of words (as, words are treated as nodes in word graph of sentences). The information propagation strength of word (i.e. closeness centrality) is a global measure w.r.t. local sequence matching. Thus, it gives better prediction about role of co-occurring words in every identified topic. The experimental results on TAC-2011 dataset also support our way to solve the problem.

## 2 Calculating Importance of Words

To calculate the importance/weight of words, we prepare reverse directed word graph of sentences and then calculate the page rank score of every distinct word. To clean documents, we remove the noisy entries and stopwords and stem the entire text by using porter stemmer. Finally, we filter the sentences. The rest of the process to calculate the importance of words is given below:

The way to prepare reverse directed word graph of sentences and calculation of page rank is given below:



**Fig. 1.** Reverse directed word graph of sentences, Here S1, S2 and S3 represents the sentences of document and 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' and 'i' represents the distinct words

**Preparing Reverse Directed Word Graph of Sentences:** For a given set of sentences i.e.  $S = \{S_1, S_2, \dots, S_n\}$ , we add a reverse directed link for every adjacent word pair of every given sentence. See Figure-1. We denote  $G = (V, E)$  as a directed graph, Where,  $V = \{V_1, V_2, \dots, V_n\}$  denotes the vertex set and  $V \in C$  and link set  $(V_j, V_i) \in E$  if there is a link from  $V_j$  to  $V_i$ .

**Calculating Page Rank Score:** we use [10] to calculate the page rank score of every word. For any given vertex  $V_i$ , let  $IN(V_i)$  be the set of vertices that point to it (predecessors), and let  $OUT(V_i)$  be the set of vertices that vertex  $V_i$  points to (successors). Then the page rank score of vertex  $V_i$  can be defined as:

$$S(V_i) = \frac{(1 - \lambda)}{N} + \lambda \sum_{j \in IN(V_i)} \frac{S(V_j)}{OUT(V_j)} \quad (1)$$

Where:

$S(V_i)$  = Rank/score of word/vertex  $V_i$ .

$S(V_j)$  = rank/score of word/vertex  $V_j$ , from which incoming link comes to word/vertex  $V_i$ .

$N$  = Count of number of words/vertex in word graph of sentences.

$\lambda$  = Damping factor (we use a fixed score for damping factor i.e., “0.85” as used in [10]).

### 3 Identifying Topics and Calculating Their Importance

To identify the topics covered in document we use Group average agglomerative clustering scheme (GAAC). In our case the topic is considered as set of sentences related to same concept. Among three major agglomerative clustering algorithms, i.e. single-link, complete-link, and average-link clustering. Single-link clustering can lead to elongated clusters. Complete-link clustering is strongly affected by outliers. Average-link clustering is a compromise between the two extremes, which generally avoids both problems. This is the main reason of use of group average agglomerative clustering algorithm for clustering the sentences.

GAAC, uses average similarity across all pairs within the merged cluster to measure the similarity of two clusters. In this scheme average similarity between two clusters (say,  $c_i$  and  $c_j$ ) can be computed as:

$$sim(c_i, c_j) = \frac{1}{|c_i \cup c_j|(|c_i \cup c_j| - 1)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j); \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y}) \quad (2)$$

Where,

$sim(\vec{x}, \vec{y})$  = count of co-occurring words in  $\vec{x}$  and  $\vec{y}$

To apply the GACC on sentences we use a sentence vector representation of entire document. Here, each row represents a sentence and each column represents a term. We use the threshold “0.4” in the entire evaluation.

**Calculating Importance of Sentence Clusters or Topics:** To calculate the weighted importance of any Sentence cluster or topic, we calculate the Sum of weighted importance of all words in the given sentence cluster. It can be given as:

$$W(C) = \sum W_{wd} \quad (3)$$

Where

$W(C)$  = weight of given sentence cluster ‘C’

$\sum W_{wd}$  = weight of all words in given sentence cluster. (see eq-1 to calculate the weight of words).

Next, we calculate the percentage of weighted information of every identified sentence cluster. It can be calculated as:

$$\%W(C) = \left( \frac{W(C)}{\sum W(C)} \times 100 \right) \quad (4)$$

Where:

$\%W(C)$  = percentage weight of given sentence cluster ‘C’.

$\sum W(C)$  = sum of weight of all identified sentence cluster.

$W(C)$  = weight of given sentence cluster ‘C’.

## 4 Preparing Evaluation Sets

At this stage, we prepare evaluation sets. Every evaluation set consists of two sets, i.e., (1) Set-1: contains set of sentences from identified sentence cluster (also denoted as topic) and (2) Set-2: contains the uniquely matching sentences from target or candidate document, which matches with Set-1. Thus, the number of evaluation set depends on the number of identified sentence clusters in reference/model summary. Briefly, each evaluation set contains an identified topic (i.e. sentence cluster from reference or model summary) and uniquely mapped set of sentences from target summary/ machine generated summary). See Figure-2 for a sample evaluation set.

## 5 Using Closeness Centrality Measure

We, use closeness centrality measure to predict the sense of co-occurring words in both sets (i.e. Set-1 and Set2) of every given evaluation set. Due to (1) topic boundary of sentences in every evaluation set (as achieved in Set-1 and Set-2 of evaluation set), and (2) global nature of closeness centrality score, the proposed system effectively predicts the sense of co-occurring words w.r.t. local measures based on sequences. Now, we remove the words which do not co-occur in both sets (i.e. Set-1 and Set-2) of given evaluation set (See Figure-3, which contains co-occurring words of Set-1 and Set-2 of evaluation set given in Figure-2). Next, we calculate the closeness centrality score of (1) co-occurring words of Set-1 and (2) Co-occurring Words of Combined Graph of Set-1 and Set-2. The details are given below:

## 5.1 Calculating Closeness Centrality Score of Co-occurring Words of Set-1

We prepare a Bi-directional graph of co-occurring words of Set-1 of given evaluation set and calculate their closeness centrality score. To prepare Bi-directional graph we add a bidirectional link for every adjacent word pair in every sentences of Set-1 (see figure-4).

**Graph Theoretical Notation:** We denote  $G = (V, E)$  as a directed graph of Set-1 of given evaluation set. Where,  $V = \{V_1, V_2, \dots, V_n\}$  denotes the vertex set and link set  $(V_j, V_i) \in E$  if there is a link from  $V_j$  to  $V_i$ .

**Path Length:** In this scheme we use link strength to calculate the path length between any two nodes. As, the link strength of any link between two nodes in word graph of sentences depends upon number of times the adjacent words co-occur in the given text. The Link Strength between two adjacent nodes can be calculated as:

$$\text{Link\_Strength} = 2 \times \{\min(\# \text{Forward\_Link}, \# \text{Backward\_Link})\} \quad (5)$$

Where,

$\# \text{Forward\_Link}$  = count of forward links between two adjacent nodes

$\# \text{Backward\_Link}$  = count of backward links between two adjacent nodes.

*Note:* As, we consider only Bi-directional links, so we use “multiply by 2” in above equation.

In general case, with the increase of count of co-occurrences, the similarity between words increases. We believe that with the increase of similarity between words the edge weight between words should decrease. We use this fact in calculation of path length. Now according to this scheme, path length can be calculated as (e.g. see figure-4, for path length of different paths):

$$\text{Path\_Length} = \frac{1}{\text{Link\_Strength}} \quad (6)$$

**Calculating Closeness Centrality Score:** The closeness centrality of any node  $V_i$  is defined as the mean geodesic distance (i.e., the shortest path) between a node  $V_i$  and all of the nodes reachable from  $V_i$  as follows:

$$C_C(V_i) = \frac{(n-1)}{\sum_{t \in V/V_i} d_G(V_i, t)} \quad (7)$$

Where,

$n$  = is the size of the connected component reachable from  $V_i$  and ( $n >= 2$ )

$C_C(V_i)$  = closeness centrality of node / vertex  $V_i$

$d_G(V_i, t)$  = sum of geodesic distance from  $V_i$  to 't', we use the path length obtained from above step in calculation of all geodesic distances (see Figure-6 for sample calculation of geodesic distances and Figure-8, for closeness centrality scores).

**NOTE:** In some cases the path from node  $V_i$  to 't' may not exist. In such cases, we consider that word at node  $V_i$  is not related to word at node 't'. So, in such cases we consider the geodesic distance from  $V_i$  to 't' as the count of total number of nodes in the graph.

Thus, in that case,  $d_G(V_i, t)$  = count of total number of nodes in given graph.

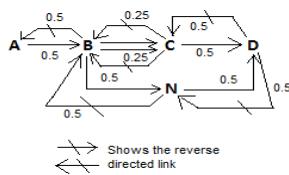
By using this scheme, we calculate the closeness centrality of every node / word of given graph.

Evaluation Set					
Set-1:			Set-2:		
A	B	C	D	M	N
B	N	D		B	D
				C	O
				D	
				A	B
				C	

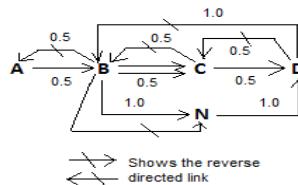
**Fig. 2.** Evaluation Set Containing Set-1 and Set-2, here upper case letters represent the words

Common words from both sets					
Set-1			Set-2		
Common Words	Set-1	Common Words	Set-2	Common Words	Set-2
A	B	C	D	N	B
B	N	D		C	D
				A	B
				C	

**Fig. 3.** Only common words from both sets are taken



**Fig. 4.** Bi-directional word graph of sentences of co-occurring words of Set-1, given in Figure-3



**Fig. 5.** Combined directed graph of co-occurring words of Set-1 and Set-2, given in Figure-3

	A	B	C	D	N
A	x	0.5	0.75	1.25	1.0
B	0.5	x	0.25	0.75	0.5
C	0.75	0.25	x	0.5	0.75
D	1.25	0.75	0.5	x	0.5
N	1.0	0.5	0.75	0.5	x

**Fig. 6.** Pair wise geodesic distances of nodes of graph shown in Figure-4

	A	B	C	D	N
A	x	0.5	1.0	1.5	1.5
B	0.5	x	0.5	1.0	1.0
C	1.0	0.5	x	0.5	1.5
D	1.5	1.0	0.5	x	2.0
N	2.5	2.0	1.5	1.0	x

**Fig. 7.** Pair wise geodesic distances of nodes of graph shown in Figure-5

Closeness Centrality scores	Closeness Centrality scores	(% Difference)
CC(A) = 0.89	CC(A) = 1.14	%Diff = 27%
CC(B) = 1.33	CC(B) = 0.5	%Diff= 166%
CC(C) = 1.14	CC(C) = 1.78	%Diff=56%
CC(D) = 0.80	CC(D) = 1.33	%Diff=66%
CC(N) = 0.57	CC(N) = 1.45	%Diff=154%

**Fig. 8.** Closeness centrality scores of nodes of graph shown in Figure-4, Figure-5 and % difference

## 5.2 Calculating Closeness Centrality Score of Co-occurring Words of Combined Graph of Set-1 and Set-2

For this, we prepare a combined directed graph by using co-occurring words of Set-1 and Set-2 of given evaluation set. Here the main aim is to exploit the differences in information flow of adjacent common words of both sets of given evaluation set in predicting the sense of co-occurring words. The details are given below:

**Graph Construction:** We take every adjacent pair of words in every sentence of Set-1 of given evaluation set and adds a forward directed link. Next, we take every adjacent pair of words of every sentence of Set-2 of given evaluation set and add a backward directed link to existing graph. Thus the constructed graph represents the flow of information of pair of adjacent words from both sets (e.g. See Figure-5).

**Path Length:** Similar to scheme discussed in sub-section 5.1, we use link strength to calculate the path length between any two nodes. But, due to the use of combined graph, Bi-directional links may not exist for some nodes. For this, we make some change in strategy. We calculate the link strength by using two additional cases i.e.

Now, the link strength between two adjacent nodes when Bi-directional links between nodes do not exist can be calculated as:

- If only Forward link(s) exist.

$$\text{Link\_Strength} = \# \text{Forward\_Links} \quad (8.1)$$

- If only backward link(s) exist.

$$\text{Link\_Strength} = \# \text{Backward\_Links} \quad (8.2)$$

Rest of the process for calculation of the path length is same, as discussed in sub-section 5.1 (see equation-6). We use this path length in calculation of geodesic distances between every pair of nodes. See Figure-7 for sample calculation of pair wise geodesic distances for graph shown in Figure-5.

**Calculating Closeness Centrality Score:** The process of calculating the closeness centrality is same as discussed in previous sub-section 5.1 (see equation-7). (e.g. See Figure-8)

### 5.3 Identifying Co-occurring Words Having Similar Sense

At this stage, we identify the co-occurring words in Set-2, having same sense w.r.t. Set-1. As, earlier it is discussed that there may be some words co-occur in both sets i.e. Set-1 and Set-2 (of evaluation set), but the role of these words in Set-2 may be different w.r.t. Set-1. To check this, we use closeness centrality scores of common words of Set-1 and Set-2 (see sub-section 5.1 and 5.2, for calculation of centrality score of node / word of graphs). Now, we calculate the “%” difference in centrality scores between every co-occurring word (node) of *Word graph of Set-1* and *combined word graph of Set-1 and Set-2* of given evaluation set. The scheme is given below:

$$\%Diff(V_i) = \left( \frac{|C_C(V_i) - C_C(V'_i)|}{C_C(V_i)} \right) \times 100 \quad (9)$$

Where,

$C_C(V_i)$  = Closeness centrality score of node/word  $V_i$  in “Word graph of Set-1”.

$C_C(V'_i)$  = Closeness centrality score of node/word  $V'_i$  in “combined word graph of Set-1 and Set-2”.

Here ( $V_i = V'_i$ ), i.e. It denotes the same word representing node on different graphs.

For example, see Figure-8 (“%Diff”). Now, we calculate the median of “%Difference” of scores of all nodes / (distinct words) and put a minimum threshold as:

$$\text{“% Diff” should be} = \begin{cases} < 50\% & \text{if } (median < 50\%) \\ < median & \text{if } (median \geq 50\%) \end{cases} \quad (10)$$

We use this threshold value in identification of words in Set-2 (which are common to both sets of given evaluation set) and (2) whose role in Set-2 is similar to the corresponding word of Set-1. We use this information in scoring in next section. The threshold used in equation-10, is fixed after a lot of observations on TAC-2009 and TAC-2010 dataset.

**Additional Note:** If in any case all the nodes of graph shows  $\%Diff(V_i) \geq 50\%$  , then it means, common words of both sets show very high diversities in roles. So, we will not consider such nodes (words) of that graph as valid nodes.

**Example:** For the evaluation set given in Figure-2, our system predicts that only two words i.e. ‘A’ and ‘C’ have similar role in both sets, i.e. Set-1 and Set-2.

Similarly for sample sentences given in sub section-1.2, our system predict “3” valid matches when comparing S2 w.r.t. S1 (i.e. 75% score on [0-100] scale), “1” valid match when comparing S3 w.r.t. S1 (i.e. 25% score), “2” valid matches when comparing “S4”, w.r.t. “S1” (i.e. 50% score) and “No” valid matching when comparing “S5” w.r.t. “S1” (i.e. 0% score). The comparison of these scores with ROUGE scores also supports our view. To evaluate these sentences we applied above discussed rules and do not use stemming and stopwords removal, as in [2].

## 6 Final Scoring

At this step, we take every evaluation set one by one and check, if Set-2 is not null, then we calculate the score for every such evaluation set. Now we apply following formula to calculate the weighted score in any given evaluation set  $S_i$ .

$$Score(S_i) = \left( \frac{\sum Count_{match}(word)}{\sum Count(word)} \times 100 \right) \quad (11)$$

Where:

$Score(S_i)$  = Evaluation score for given evaluation set  $S_i$ .

$\sum Count_{match}(word)$  = count of co-occurrences of all such words in Set-1, (1) which co-occur in both sets i.e. Set-1 and Set-2 and (2) pass the minimum frequency threshold related criteria. As described earlier, (see sub-section 5.3, equation 9 and 10).

$\sum Count(word)$  = Count of all words in Set-1 of given evaluation set.

*Note:* In any given evaluation set, if there does not exist any mapped sentences in Set-2, then we set the evaluation score of that evaluation set to zero.

$$Score(S_i) = 0; \quad (12)$$

**Calculating Final Score:** For this we just add the evaluation scores of all evaluation sets.

## 7 Pseudo Code

**Input:** (1) reference / model summary, (2) candidate / machine generated summary, both in ASCII format.

**Output:** %score, which can be further normalized to “0-1” scale.

**Algorithm:**

1. Apply pre-processing and input cleaning for reference / model summary and candidate / machine generated summary and calculate the weight of every word of reference / model summary. (See section-2).
2. Identify the sentence clusters in reference / model document and calculate the weighted importance of every identified sentence cluster (See section-3).
3. Prepare separate evaluation sets by using every identified sentence cluster of reference / model summary by uniquely mapping the sentences from candidate / machine generated summary (See section-4).
4. Use closeness centrality measure to identify the co-occurring words having same sense in both sets of given evaluation set (See section-5).
5. By using (1) co-occurring words having same role in both sets of every evaluation set and (2) importance of sentence cluster (i.e. Set-1 of given evaluation set), calculate the final evaluation score of given machine generated summary (See section-6).

## 8 Experiments

We, use TAC-2011, AESOP dataset to evaluate our system. The details of dataset, evaluation strategies, metrics, baselines and results are given below.

**Evaluation Strategies, Metrics and Baselines:** For automatic evaluation of summary quality, we consider two cases i.e. (1) All peer and (2) No-Model cases with both parts, i.e. Initial summary and update summary. Thus we have total for evaluation results, i.e. (1) All-Peer evaluation with initial summary, (2) All-Peer evaluation with update summary, (3) No-Model evaluation with initial summary and (4) No-Model evaluation with update summary.

**Metrics:** To judge the summary quality, we calculate (a) Pearson's, (b) Spearman's, and (c) Kendall's correlations with (1) Pyramid, (2) Overall Responsiveness and (3) Readability.

**Baselines:** We consider TAC-2011 AESOP baseline and TAC-Best Score, for comparison purpose. The details are given below:

- i. **Baseline-1:** ROUGE-2, with stemming and keeping stopwords.
- ii. **Baseline-2:** ROUGE-SU4, with stemming and keeping stopwords.
- iii. **Baseline-3:** Basic Elements (BE). Summaries were parsed with Minipar, and BE were extracted and matched using the Head-Modifier criterion.
- iv. **TAC-Best Score:** It contains best TAC-2011 scores on every evaluation metric by TAC-2011 participants/benchmarks.

### Results on AESOP Test Dataset

Results are given in Table-1, 2, 3 and 4. (1) Table-1, shows All-Peer evaluation with initial summary, (2) Table-2 shows All-Peer evaluation with update summary, (3) Table-3 shows No-Model evaluation with initial summary and (4) Table-4 shows No-Model evaluation with update summary.

In all four tables, the first three rows headed as “Baseline-1”, “Baseline-2”, and “Baseline-3” represents the corresponding baseline scores as obtained from TAC-2011 results. “TAC-Best score” shows the best TAC-2011 scores. The last row of all four tables contains the score of our devised system.

The correlation score with (1) Pyramid, (2) Overall Responsiveness and (3) Readability given in Table 1, 2, 3 and 4, show that our devised system (1) performs better than all three baseline systems and (2) comparable with TAC-Best Scores. In all four tables highest scores are represented by bold font.

**Table 1.** AESOP-ALL Peers (Initial Summary), correlation with Pyramid, Responsiveness, Readability

System	Correlation with Pyramid			Correlation with Responsiveness			Correlation with Readability		
	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
Baseline-1	0.572	0.864	0.703	0.725	0.779	0.609	0.663	0.498	0.374
Baseline-2	0.763	0.886	0.723	0.733	0.810	0.629	0.682	0.533	0.400
Baseline-3	0.781	0.878	0.720	0.752	0.784	0.590	0.683	0.531	0.387
TAC-Best Score	0.975	0.933	0.799	<b>0.972</b>	0.894	0.740	0.926	<b>0.674</b>	0.519
<b>OUR SYSTEM</b>	<b>0.976</b>	<b>0.935</b>	<b>0.799</b>	0.968	<b>0.895</b>	<b>0.743</b>	<b>0.926</b>	0.673	<b>0.519</b>

**Table 2.** AESOP-ALL Peers (Update Summary), correlation with Pyramid and Responsiveness

System	Correlation with Pyramid			Correlation with Responsiveness			Correlation with Readability		
	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
Baseline-1	0.775	0.851	0.684	0.717	0.869	0.710	0.712	0.550	0.399
Baseline-2	0.730	0.883	0.720	0.675	0.903	0.743	0.686	0.558	0.405
Baseline-3	0.740	0.848	0.686	0.649	0.808	0.637	0.611	0.415	0.287
TAC-Best Score	0.953	<b>0.891</b>	0.731	<b>0.975</b>	0.911	<b>0.762</b>	<b>0.934</b>	<b>0.663</b>	0.507
<b>OUR SYSTEM</b>	<b>0.956</b>	0.887	<b>0.733</b>	0.954	<b>0.921</b>	<b>0.762</b>	<b>0.934</b>	0.645	<b>0.510</b>

**Table 3.** AESOP-NO Models (Initial Summary), correlation with Pyramid and Responsiveness

System	Correlation with Pyramid			Correlation with Responsiveness			Correlation with Readability		
	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
Baseline-1	0.961	0.894	0.745	0.942	0.790	0.610	0.752	0.398	0.292
Baseline-2	0.981	0.894	0.737	0.954	0.790	0.602	0.784	0.395	0.292
Baseline-3	0.939	0.903	0.746	0.915	0.768	0.567	0.717	0.405	0.291
TAC-Best Score	<b>0.981</b>	0.903	0.758	<b>0.954</b>	<b>0.845</b>	0.675	<b>0.819</b>	0.497	0.366
<b>OUR SYSTEM</b>	<b>0.981</b>	<b>0.904</b>	<b>0.760</b>	0.945	<b>0.845</b>	<b>0.681</b>	0.815	<b>0.499</b>	<b>0.369</b>

**Table 4.** AESOP-NO Models (Update Summary), correlation with Pyramid and Responsiveness

System	Correlation with Pyramid			Correlation with Responsiveness			Correlation with Readability		
	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
Baseline-1	0.903	0.802	0.632	0.917	0.840	0.678	0.658	0.373	0.263
Baseline-2	0.885	0.838	0.665	0.912	0.876	0.706	0.672	0.363	0.254
Baseline-3	0.906	0.838	0.684	0.876	0.796	0.625	0.545	0.245	0.162
TAC-Best Score	<b>0.911</b>	0.838	<b>0.684</b>	0.927	0.877	<b>0.716</b>	0.742	<b>0.482</b>	0.361
<b>OUR SYSTEM</b>	0.910	<b>0.840</b>	0.668	<b>0.930</b>	<b>0.881</b>	0.689	<b>0.745</b>	0.450	<b>0.363</b>

## 9 Conclusion and Future Work

In this paper we presented a graph based flexible mapping approach to identify the matching word patterns of any type/length and also devised a closeness centrality based scheme, which identify the role of matching words in the entire context of information (globally). In other words, it calculates the role of matching sequences / matching words in both i.e. set of reference sentences and set of candidate sentences. We use both scheme in automatic evaluation of machine generated summary.

The experimental results on TAC-2011, AESOP dataset shows that, our devised system performs better than TAC benchmarks and is better/comparable with TAC-Best Scores. It is remarkable to note that our devised system does not require heavy linguistic resources and truly unsupervised in nature.

Due to flexible structure, our devised system can be extended to evaluate the short answers, text passages etc (similar to the extended use, given in [9]).

**Acknowledgment.** For this work, partial financial support is provided by MHRD, INDIA.

## References

1. Nenkova, A., Passonneau, R., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.* 4(2), 4 (2007)
2. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: *Proceedings of HLT-NAACL 2003* (2003)
3. Teufel, S., van Halteren, H.: Evaluating Information Content by Factoid Analysis: Human Annotation and Stability. In: *Proceedings of the NLP 2004 Conference*, Barcelona, Spain (2004)
4. Nenkova, A., Passonneau, R.: Evaluating Content Selection in Summarization: The Pyramid Method. In: *Proceedings of the HLT-NAACL 2004 Conference* (2004)
5. Hovy, E.H., Lin, C.Y., Zhou, L.: Evaluating DUC 2005 using Basic Elements. In: *Proceedings of DUC-2005 Workshop* (2005)
6. Hovy, E.H., Lin, C.Y., Zhou, L., Fukumoto, J.: Automated Summarization Evaluation with Basic Elements. In: Full paper. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy (2006)
7. Porter Stemming Algorithm for suffix stripping, web -link,  
[http://telemat.det.unifi.it/book/2001/wchange/download/stem\\_porter.html](http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html)
8. Kumar, N., Srinathan, K., Varma, V.: Evaluating Information Coverage in Machine Generated Summary and Variable Length Documents. In: *COMAD-2010*, Nagpur, India (2010)
9. Kumar, N., Srinathan, K., Varma, V.: An Effective Approach for AESOP and Guided Summarization Task. In: *TAC 2010 Workshop* (2010), <http://www.nist.gov/tac>
10. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford digital library technologies project (1998)

# Combining Syntax and Semantics for Automatic Extractive Single-Document Summarization

Araly Barrera and Rakesh Verma

University of Houston, Houston TX USA,  
Computer Science Department  
`abarrera7@uh.edu, rmverma@cs.uh.edu`

**Abstract.** The goal of automated summarization is to tackle the “information overload” problem by extracting and perhaps compressing the most important content of a document. Due to the difficulty that single-document summarization has in beating a standard baseline, especially for news articles, most efforts are currently focused on multi-document summarization. The goal of this study is to reconsider the importance of single-document summarization by introducing a new approach and its implementation. This approach essentially combines syntactic, semantic, and statistical methodologies, and reflects psychological findings that pinpoint specific selection patterns as humans construct summaries. Successful summary evaluation results and baseline out-performance are demonstrated when our system is executed on two separate datasets: the Document Understanding Conference (DUC) 2002 data set and a scientific magazine article set. These results have implications not only for extractive and abstractive single-document summarization, but could also be leveraged in multi-document summarization.

## 1 Introduction

The Internet age brings forth an alarming rate of text documents (from news articles to electronic books to scientific papers, etc.), making it difficult for people to cope. Since as early as the 1950s, automated summarization of documents has been studied in an effort to alleviate an information overload problem considered to exist even then. The goal of this area of research is simply to reduce the vast amounts of information into compact summaries so that users can locate the most important pieces of information more easily from the “haystack.”

Two main methods of summarization are [14]: 1) *abstractive* - the construction of original sentences from one’s own thoughts, understanding, and experiences, and 2) *extractive* - the selection of most salient source sentences. Due to the complex linguistic and real-world knowledge required for truly abstractive summaries, extractive summarization has become a more popular choice for computation and is the focus of this study.

Although summarization has been studied for almost 50 years now, there has been a decline in recent research on single document summarization. In 2001-02 the Document Understanding Conference<sup>1</sup> (DUC) proposed the task of creating

---

<sup>1</sup> <http://duc.nist.gov>

100-word summaries of individual news articles but soon after dropped single document summarization competitions to move on to multi-document extraction and update summarization. This, according to [21,18], was due to the fact that no system [in DUC 2001-2002] could outperform the baseline with statistical significance. The baseline, an extract consisting of the first portions of a document, has been generally accepted as a good representation of a news-article summary. Outperforming baseline standards essentially indicates a summarizer of high-quality, but for many researchers, the notion of single-document summarization remains that of an underperformer and essentially a more difficult task than multi-doc summarization [21,17].

In this work we revisit the important problem of single-document summarization and reconsider the performance of the baseline in a different context, viz., scientific magazine articles. We design a new and robust approach for single document summarization that ranks an article's sentences based on semantics, overall word popularity, and sentence position. We subject it to intensive experiments using two datasets: scientific magazine articles, and the DUC 2002 news collection from NIST. We compare our approach and its implementation against the baseline(s), the popular MEAD summarizer available on the internet [19], TextRank sentence extraction [16], and, for news data, the systems that participated in the DUC 2002 competition. We show that: (i) our system outperforms all the systems including the baselines, and (ii) for scientific article dataset, our system beats the baselines by a wide, statistically significant margin. For news articles, our system beats the baseline, but not by a statistically significant margin. Hence, our results also demonstrate that the baseline's presumed superiority *so far* only holds for news data.

The organization of the rest of this paper is as follows. Section 2 presents our system and its implementation and Section 3 describes the data sets used for system trials. Section 4 provides the evaluation methodology, Section 5 the results and Section 6 some perspective on the results. Section 7 discusses the related work and Section 8 concludes the paper.

## 2 Method and System Overview

As a whole, our system is designed to handle both syntactic and semantic qualities of a document's text. It implements part-of-speech (POS) tagging<sup>2</sup>, named entity recognition<sup>3</sup>, stopword removal<sup>4</sup>, TextRank word extraction[16] for word popularity ranking, SenseLearner<sup>5</sup> for word disambiguation, a parser for heading recognition and filtering<sup>6</sup> and the popular WordNet [6] database tool for deeper word analysis. Figure 1 illustrates the entire process of our system.

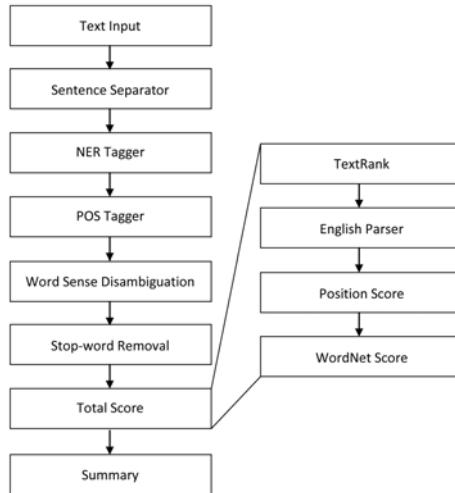
<sup>2</sup> Stanford POS Tagger: <http://nlp.stanford.edu/software/tagger.shtml>

<sup>3</sup> Stanford NER Tagger: <http://wwwnlp.stanford.edu/software/CRF-NER.shtml>

<sup>4</sup> <http://search.cpan.org/creamyg/Lingua-StopWords-0.09/lib/Lingua/StopWords.pm>

<sup>5</sup> R. Mihalcea and A. Csomani. SenseLearner: Word Sense Disambiguation for all Words in Unrestricted Text. ACL, 2005.

<sup>6</sup> Link Grammar Parser: <http://www.link.cs.cmu.edu/link/>



**Fig. 1.** Block Diagram of our system

The focus of this section is on our system’s sentence scoring algorithm, which has a major influence on the extraction of a document’s sentences for the construction of a summary. This method consists of assigning a score to each sentence that is the aggregate of three key weighted scores: 1) A **TextRank score**, based on the TextRank keyword extraction algorithm [16] to rank popularity of words within a document and to exploit the presence of these words in document sentences. 2) A **WordNet score** utilizing the WordNet [6] lexical database in three different models proposed for semantic prioritization. 3) A **Position score** which exploits a sentence’s relative position within the text, a feature that humans naturally use for extraction.

## 2.1 Total Sentence Score

A final sentence score is assigned to each sentence as a linear combination of the Position ( $P$ ), WordNet ( $WN$ ), and TextRank ( $T$ ) scores using the equation:

$$TotalScore(S_i) = w_1P(S_i) + w_2WN(S_i) + w_3T(S_i) \quad (1)$$

where  $w_1 + w_2 + w_3 = 1$ . Essentially, resulting top scoring sentences are selected and used as the document’s final summary, whose size would depend on a compression rate constraint<sup>7</sup> specified for the task at hand.

<sup>7</sup> See Section 5 for the different compression rates used for the datasets analyzed in this study.

## 2.2 TextRank Score

Our system implements the TextRank algorithm [16] to extract important keywords from a text document and also to determine the word's weight of importance within the entire document. The TextRank keyword extraction algorithm is a graph-based ranking model for graphs generated from text and is primarily based on PageRank [4]. Those words containing most co-occurring connections to other words in a graph are thus ranked with greater weights and thus considered most popular. Optimal results have been found when using only nouns and adjectives in this implementation. *The primary purpose of using this function is by giving higher weight to sentences containing a larger quantity of these popularly-used words as a means of selecting more thematic information.* The TextRank score ( $T$ ) as mentioned in equation (1) used in our system for a sentence  $S_i$ , which is a multiset (or bag) of words,  $w$ , is the following:

$$T(S_i) = \frac{\sum_{w \in \text{Nouns}(S_i) \cup \text{Adjectives}(S_i)} I(w)}{|S_i|} \quad (2)$$

where  $I(w)$  computes the word's importance, as detailed in [16]. The TextRank score for each sentence is normalized by dividing  $T(S_i)$  with the maximum TextRank score of all sentences.

## 2.3 Position Score

Our system explores three position models. *The first position model is based on the assumption that sentences near the beginning and end of a document are more likely to be included in effective summaries.* This assumption is accomplished through the following cosine position score model,  $P_{cos}$ , which in previous empirical testing yielded superior results:

$$P_{cos}(S_i) = \frac{\cos \frac{2\pi x}{k-1} + \alpha - 1}{\alpha} \quad (3)$$

where  $\alpha$  is the *dent factor* ( $\alpha = 2$  was used in the evaluations described below based on optimal results obtained in prior experiments). The idea is that as  $\alpha$  increases, the  $P_{cos}$  score becomes more equally distributed, and as  $\alpha$  decreases,  $P_{cos}$  becomes more concentrated to value one at the beginning and end of a document. Here,  $k$  represents the total number of sentences in the document and  $x$  is the position of sentence  $S_i$  within the document. The first sentence in the text document would have an  $x$  value of 0 and the last sentence an  $x$  value of  $k - 1$ .

The following function was the linear position score model used,  $P_{lin}$ , for an individual sentence in a document. *The assumption in using this model is based on efforts to prioritize sentences closer to only the top portions of a document.* This is accomplished through the following scoring model:

$$P_{lin}(S_i) = 1 - \frac{x}{k} \quad (4)$$

where  $x$  and  $k$  represent the same values as in the cosine position score equation (3) above. Essentially, as the  $x$  value increases, the score decreases, giving higher weight to the sentences at top portions of a document.

A third position score function was designed for our system based on a correlation and regression analysis performed by us (we omit this for lack of space here) on data obtained from a previous cognitive experiment [13]. Essentially, a set of four scientific articles articles (that either contained heading or not) were assigned to a group of people who were asked to make short summaries from these. Of all different factors analyzed from this data, sentences closer to preceding *signaling devices* such as headings or titles, were found to be mostly correlated with human sentence extraction.

*The purpose of this scoring algorithm is therefore to prioritize sentences closer to topic headings, a condition that as we've seen, has a strong effect in sentence extraction decisions made by humans.* Of all four article analyzed, we found the following equation to model this correlation best (we omit details of this analysis but present here the equation used as a means of making closer human extractions):

$$P(S_i) = -19 \ln(d_i) + 51.926 \quad (5)$$

where  $d_i$  represents the positional distance of sentence  $S_i$  from a previous signaling device, such as a title, or heading encountered in the document. The position scores of each sentence are normalized by dividing with the maximum respective position score.

## 2.4 WordNet Score

The WordNet score method (so named due to the use of WordNet [6]) is the major word analysis component of the our system. One approach to a sentence's WordNet score is to determine the combination of a sentence's noun and verb score as a means of selecting a document's most *thematic* sentences. Informally, the noun and verb scores ( $NS$  and  $VS$ ) determine the location of nouns and verbs, respectively, within the WordNet hypernymy graphs. *Hypernyms* here are words that by definition, are general representatives of other words. For instance, the word *dog* would be considered a hypernym to the word *poodle*, and *poodle* a *hyponym* to *dog* since *poodle* is a *type* of dog. *The purpose of this scoring algorithm is to prioritize sentences containing nouns and verbs closer to the their root forms since these could lead to the most thematic sentences.* The first WordNet score ( $WN$ ) model for an individual sentence is presented as follows:

$$WN(S_i) = 1 - \frac{VS + NS}{(|Nouns(S_i)| + |Verbs(S_i)|)^2} \quad (6)$$

Here  $Noun(S_i)$  (resp.  $Verbs(S_i)$ ) denotes the set of nouns (resp. verbs) in sentence  $S_i$ .  $VS$ , here, represents a total verb score given to the individual verbs of the sentence and their distances to their own root forms within the hypernymy tree structure.  $NS$ , similarly, represents a total score given to the nouns of the

sentence and their distances to their roots forms (for details on NS and VS calculations, see [22]). Essentially, the more general the nouns and verbs of a sentence are (determined by a simple traversing mechanism using WordNet's hypernymy tree structures), the higher the WordNet score weight is for this model. The denominator is squared based on results from prior experiments.

*The second model presented is intended to give higher priority to sentences containing words close in meaning to the article's popular keywords (computed by TextRank keyword extraction) and any other heading keywords within the entire text document.* The reason for its use is also based on the importance of keywords in sentence extraction, but this with the intention of examining keyword semantics through WordNet synonym lists.

The computation of this WordNet model revolved around the collection of a *thematic word list*, the combination of all document headings and the top five percent popular words generated by the TextRank keyword extraction algorithm [16]. Each document sentence,  $S_i$  is assigned a score based on its individual bag of words with the following equation for each word  $w$ :

$$score(w) = \frac{1}{2^l}$$

where  $l$  is the minimum level determined when  $w$  is compared in meaning to words in *thematic word list* through WordNet's synonym lists, known as synsets. For instance, if  $w$  is a word found in the thematic word list then  $score(w) = 1$  (level  $l = 0$ ). Otherwise,  $l$  is increased by one to ( $l = 1$ ) and  $w$  is now compared to the entire WordNet [6] synset list of the preceding level of the thematic word list. If no match is found, synsets of preceding synsets are determined with up to a maximum of 4 levels. Say  $w$  is found at  $l = 3$ , then  $score(w) = \frac{1}{8}$ .  $WN_{syn}$  score for  $S_i$  in *SynSem* therefore became:

$$WN_{syn}(S_i) = \sum_{w \in S_i} score_{syn}(w) \quad (7)$$

Higher WordNet scores are achieved as the closer a sentence word,  $w$ , is to the *thematic list*'s synonyms, where  $score_{syn}(w)$  represents the  $score(w)$  computed using the WordNet synset relation. A third method is also presented based on the same procedure listed above, except hypernyms sets are used in place of the synsets described in Step 4. Equation for WN using hypernyms is similarly:

$$WN_{hyp}(S_i) = \sum_{w \in S_i} score_{hyp}(w) \quad (8)$$

Higher WordNet scores are similarly achieved as the closer a sentence word,  $w$ , is to the *thematic list*'s hypernym, or general word-forms, list. Here,  $score_{hyp}(w)$  represents the  $score(w)$  computed using the WordNet hypernymy relation. All three scores,  $WN$ ,  $WN_{syn}$  and  $WN_{hyp}$  scores for each sentence are normalized by dividing with the maximum respective WordNet score over all sentences.

### 3 Data Sets

Two separate datasets were used for evaluating our and other systems' performances: 1) Cognitive experiment data originating from [13] (inspiration for equation (5) of 2.3) and composed of scientific-type magazine articles along with corresponding human-generated summaries and 2) DUC 2002 (sponsored by the National Institute of Standards and Technology, NIST) newspaper article set. Note that, 2002 is the last year in which participating systems in DUC were assigned to produce single document summaries, the very task analyzed for this study. Most systems that participated in DUC 2002 are not available for download, however, the summaries they produced for the DUC 2002 competition are available to us through NIST. The other systems used for comparison in this study include MEAD [19] and TextRank sentence extraction [16].

1. **Dataset A.** *Scientific article dataset* - The data obtained from the cognitive experiment contains the original four article versions distributed to the experiment's participants and the corresponding summaries constructed by the participants. Essentially, there were two different versions *A* and *B* of an article titled, "Energy Problems and Solutions" assigned to readers. These, however, were distributed as article versions that contained headings (which we refer to as *YA* and *YB*) and versions that lacked headings (which we refer to as *NA* and *NB*). The idea of that study was to determine the effects that headings had on human extraction, which were found to have major impact. We test our system on this dataset along with the human-constructed summaries as model extracts and evaluate our system's performance with respect to them.
2. **Dataset B.** *DUC02* The data provided by DUC 2002 contains a total of 533 unique news articles.<sup>8</sup> *Different variations of our system were executed among this set as well, constructing total summary sizes of exactly 100 words per article.* Note that DUC02 data is composed of articles containing no topic headings and only one title. Hence, the position scoring method (position score equation (5)) presented could not fully exploit its intended use in this particular set.

### 4 Evaluation

ROUGE [10] evaluation scores were used to compare our system extractions (using varying scoring models presented in this paper) to each other and to those produced by MEAD and TextRank. This fully automated evaluator essentially measures content similarity between system-developed summaries and

---

<sup>8</sup> Since multiple document summarization was also a task in DUC 2002 the files are grouped together in sets and these sets overlap - a total of 34 files are repeated in the collection, which brings the number down from 567 to 533. For single document summarization it does not make sense to repeat articles and can bias the results, so we have eliminated duplicate articles.

corresponding model summaries, usually developed by humans. Of all forms of measures it utilizes, ROUGE n-gram co-occurrences between system summaries and model summaries have been of most interest to our experiments. The n-grams, in this case, would specify the number of n consecutive word units that would have to overlap between a summary sentence and a model sentence in order to be counted as a match.

Each dataset required a set of model summaries for proper evaluations to take place. For Dataset A, the top 15 selected sentences for each article version were used as models for the evaluations. In the case of Dataset B, two manually produced 100-word reference summaries (these are abstractive, not extractive) are provided for each article in the data and used for the evaluation. All the ROUGE evaluations use all the words in the summaries, i.e., we do not use stemming (word generalization) or stopword elimination.

## 5 Results

The following results illustrate various executions of our system, TextRank sentence extraction [16], and MEAD [19] on the pair of datasets used in our analysis. We compare the evaluations of those summaries to the datasets' baselines. *In the case of Dataset B (DUC02), the baseline consisted of a summary of the first 100 words of each article and for Dataset A, the baseline consisted of the first 15 sentences from the source article version.* In addition, compression rates were established as follows. *All systems were required to produce 15-sentence summaries for Dataset A and 100-word summaries for Dataset B.* These were determined based on the original summary requirements corresponding to each dataset.

Table 1 shows ROUGE uni-gram evaluations when the systems were executed on Dataset A. For lack of space, we show results for only *YB* and *NB* (article version *B* containing headings [*YB*] and the one lacking headings [*NB*.]) In the case of executions made on *YB* (Table 1(a)), our system's ROUGE scores manage to outperform those of MEAD, TextRank sentence extractions, and the baseline, most importantly. Best results were achieved in our system using topic heading filtering<sup>9</sup>, position equation (4) and WordNet equation (6) with an F-measure of 0.71937. Executions made on the *NB* article (shown in Table 1 (b)) demonstrate similar outperformance and a highest F-measure resulting from our system of 0.65209. The same model combinations executed on both versions of the article resulted within the top 7 scoring systems of Tables 1 (a) and (b).

Table 2 illustrates ROUGE uni-grams scores on various executions made by our system for Dataset B. For lack of space, we only show a few top scoring model combinations. Optimal results here were achieved using the position model prioritizing sentences closer to distances-to-preceding headings (position equation 5), and the WordNet model exploiting synonym linkage to thematic content (WordNet equation 7), or in the case of DUC02, article titles and popular words.

---

<sup>9</sup> An option to filter headings and the inclusion of these in a final extract is an additional aspect of our system.

**Table 1.** Basic ROUGE evaluation scores for the baseline, our system, MEAD, and TextRank sentence extraction on Dataset A, showing results for articles *YB* and *NB*

Parameter Key for (Our System)			
$N$	Removal of topic headings in summary	$P_m$	Position Score $P$ model ( $m$ )
$H$	Inclusion of topic headings in summary	$WN_m$	WordNet Score $WN$ model ( $m$ )
(a) <i>YB</i>		(b) <i>NB</i>	
<b>Execution on <i>YB</i></b>		<b>Execution on <i>NB</i></b>	
<i>(Conditions)</i>		<i>Recall</i>	<i>Precision</i>
$(N, P_4, WN_6)$		.74897	.69202
$(H, P_4, WN_6)$		.70782	.67717
$(N, P_5, WN_7)$		.55144	.62617
$(N, P_5, WN_6)$		.63786	.54007
<i>Baseline</i>		.39506	.61146
MEAD		.52263	.42617
TextRank		.59671	.36341
			<i>F-meas.</i>
			.71937
			.69216
			.58643
			.58491
			.48000
			.46950
			.45172
<i>(Conditions)</i>		<i>Recall</i>	<i>Precision</i>
$(H, P_4, WN_6)$		.65079	.65339
$(N, P_4, WN_6)$		.65476	.63218
$(N, P_5, WN_7)$		.59921	.66520
$(N, P_5, WN_6)$		.58730	.66667
MEAD		.50794	.42953
<i>Baseline</i>		.49603	.43103
TextRank		.55556	.34913
			<i>F-meas.</i>
			.65209
			.64327
			.63048
			.62447
			.46546
			.46125
			.42879

**Table 2.** Basic ROUGE evaluation scores for our system on Dataset B – DUC02

<b>DUC02 Dataset</b> <i>(Conditions)</i>	<b>ROUGE uni-gram Scores</b>			
	<i>Recall</i>	<i>Precision</i>	<i>F-measure</i>	<i>95% conf. int.</i>
$(N, P_5, WN_7)$	.48159	.45062	.46549	[.45753-.47260]
$(N, P_3, WN_7)$	.48111	.44995	.46491	[.45715-.47252]
$(N, P_5, WN_6)$	.47965	.45145	.46491	[.45774-.47236]
$(N, P_5, WN_8)$	.47920	.45195	.46488	[.45738-.47257]
$(N, P_4, WN_7)$	.48091	.44965	.46466	[.45689-.47236]
$(N, P_4, WN_6)$	.47941	.45113	.46462	[.45691-.47218]
$(N, P_4, WN_8)$	.47930	.45098	.46450	[.45724-.47228]

Table 3 presents the top 7 out of 13 participating DUC02 systems compared with our system (the highest scoring from Table 2), MEAD, TextRank, and the baseline, all whose summaries contain up to 100 words only.<sup>10</sup> Our system using topic filtering, the closest distance-to-preceding headings position model, and the WordNet method exploiting synonyms obtains higher ROUGE F-measure scores than the baseline and all other participating systems but S28, a system which failed to produce one summary. Our system was ranked second in F-measure but according to [2,11], the recall metric can be prioritized since precision scores can be manipulated by adjusting the length of a candidate, or system, summary. If recall is only taken into consideration, then our system would rank first.

<sup>10</sup> Both manual abstracts and the system summaries are truncated to exactly 100 words whenever they exceed this limit.

**Table 3.** Basic ROUGE evaluation scores for our system, top 7 DUC02 systems, MEAD, TextRank sentence extraction, and the baseline

DUC02 System	ROUGE uni-gram Scores			
	Recall	Precision	F-meas.	95% conf. int.
S28	.47813	.45779	.46729	[.45986-.47418]
<b>Our System</b>	<b>.48159</b>	<b>.45062</b>	<b>.46549</b>	<b>[.45753-.47260]</b>
S19	.45563	.47748	.46309	[.45427-.47202]
<i>Baseline</i>	.47788	.44680	.46172	[.45413-.46944]
S21	.47543	.44635	.46029	[.45209-.46802]
TextRank	.46165	.43234	.44640	[.44004-.45348]
S29	.46100	.44557	.45269	[.44585-.45982]
S23	.43188	.47585	.45018	[.44191-.45900]
S27	.45485	.44808	.45014	[.44227-.45862]
MEAD	.44506	.45290	.44729	[.43961-.45508]
S15	.44805	.43323	.44014	[.43203-.44799]

## 6 Discussion

From the experimental results presented here, it is clear that our system succeeds in identifying important sentences in a text using information that is present only in the text and to do this within a summary that manages to outperform the documents' baseline. It is an unsupervised system with one caveat, the issue of weight selection, and requires no training data. When only a single article on a topic is available, we have devised unweighted schemes that deliver performance very close (F-measure to within 2-3%) to the optimal weighted schemes. For lack of space we omit these schemes and their performance here.

When a set of related articles is available, selection of weights can be done by adding a tuning module that uses a random subset of the data to find the best weight combination for the subset and then using it for the entire collection. To test this hypothesis, we conducted two experiments. In the first we took ten random samples of  $\lceil \sqrt{D} \rceil$  articles from the  $D = 533$  articles in DUC02 dataset and found the optimal combination of weights for each sample using ROUGE. All ten combinations of weights for the samples were in the top ten (F-measure) weight combinations for the entire DUC02 dataset. Of these ten optimal weight combinations for the samples, the two weight combinations with the highest frequencies, three times each, are the second and fifth best for the entire collection. This means that a small number of square-root size samples can give a near optimal combination of weights for the entire corpus. In a second experiment, we took 30 random samples of  $\lceil \log_2 D \rceil$  documents from the 533 DUC02 documents, but here the results were not as good (a few optimal weight combinations for the sample were not in the top ten combinations for the entire dataset) as for the square-root size samples.

Our system outperforms MEAD and TextRank sentence extraction in all experiments of evaluation and is consistently higher than the baseline. Its ROUGE

scores are also statistically significantly higher (through ROUGE 95% confidence intervals) than the baseline for the scientific magazine article set, where it is able take advantage of the headings in the article for its position score and the summary size restriction is on the number of sentences. When there are no headings in the articles and summary needs to be shorter (100 words versus 15 sentences), as for instance in the DUC dataset, it still beats the baseline.

## 7 Related Work

Sentence position has been considered important to summarization and information extraction ever since the late 1950s [3]. Many researchers have proposed using it for automatic summarization, e.g., see [5], [10], [20] and [14]. The importance of sentence position in *book length* documents was studied by [15], which are outside the scope of our study. Most researchers use sentence position based on their opinion of the language in which the document is written. Many use a linear function of the sentence position [9], [8] or sentence position with respect to a centroid sentence [20], others use either the first few sentences in a paragraph or the document. To our knowledge, this is the first objective study that analyzes human summary data for a “newspaper-length” article without requiring any key words<sup>11</sup>. Moreover, our work shows the importance of considering derived variables from the sentence position, not just the raw sentence position, and we observe a logarithmic relationship.

The importance of keywords or key phrases for summarization is also well-recognized since at least [5]. Many researchers have proposed using it, e.g., ([9], [16]) among others. Although WordNet was used before in summarization, e.g, in SUMMARIST [7] for the task of topic interpretation, the usage is quite different from that of our methods and our WordNet scoring methodology is new to the best of our knowledge.

## 8 Conclusions

In this paper we have described the implications of basing a single-document summarization system on combining new syntactic and semantic techniques for sentence scoring. Results have demonstrated topic heading relevance to the overall position, and semantic linkages have produced effective summaries when experimented on both the DUC02 newswire and the scientific magazine article sets.

Our approach is easily adapted to specific domains that have ontologies available. There are several interesting directions for future work: the incorporation of heuristics that optimize the score of a summary given a size constraint, sentence compression (e.g., [1]), and criteria for measuring inter-sentence redundancy and its minimization. We have recently extended this approach to multi-document summarization and are currently evaluating it. Extensions of our approach on stronger semantic summary evaluations are other avenues for the future.

---

<sup>11</sup> Lin and Hovy’s work on optimum position policy [12] requires a corpus along with key words.

## References

1. Angheluta, R., Mitra, R., Jing, X., Moens, M.-F.: K.U. Leuven Summarization System at DUC 2004. Available on the Web (2004)
2. Arora, R., Ravindran, B.: Latent Dirichlet Allocation and Singular Value Decomposition based Multi-Document Summarization. In: ICDM 2008: Proceedings of the 2008 Eighth IEEE Int'l Conf. on Data Mining, pp. 713–718 (2008)
3. Baxendale, P.: Machine-made Index for Technical Literature - An Experiment. IBM Journal of Research Development 2(4), 354–361 (1958)
4. Brin, S., Page, L.: The Anatomy of Large-scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems 30, 1–7 (1998)
5. Edmundson, H.: New Methods in Automatic Extraction. Journal of ACM 16(2), 264–285 (1969)
6. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
7. Hovy, E., Lin, C.: Automatic Text Summarization in SUMMARIST. In: Mani, Maybury, M. (eds.) Adv. in Text Summarization, vol. 1. MIT Press (1999)
8. Ishikawa, K.: Trainable Automatic Text Summarization Using Segmentation of Sentence. In: Proceedings of the Third NTCIR Workshop (2003)
9. Li, S., Wang, W., Wang, C.: TAC 2009 Update Summarization Task of ICL. In: Text Analysis Conference 2008 (2008)
10. Lin, C.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization Post-Conference Workshop (ACL 2004), Barcelona, Spain (2004)
11. Lin, C., Hovy, E.: Automatic Evaluation of Summaries Using n-gram Co-occurrence Statistics. In: HTL-NAACL (2003)
12. Lin, C.-Y., Hovy, E.H.: Identifying topics by position. In: ANLP, pp. 283–290 (1997)
13. Lorch, R., Lorch, E.: Effects of Headings of Text Recall and Summarization. Contemporary Educational Psychology 21, 261–278 (1996)
14. Mani, I., Maybury, M.: Advances in Automatic Summarization. MIT Press, Cambridge (1999)
15. Mihalcea, R., Ceylan, H.: Explorations in Automatic Book Summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2007), Prague (2007)
16. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 (March 2004)
17. Nenkova, A.: Automatic Text Summarization of Newswire: Lessons Learned from the document understanding conference. In: AAAI, pp. 1436–1441 (2005)
18. Nenkova, A.: A General Introduction to Automatic Summarization (2009), <http://webcast.jhu.edu/mediaside/Viewer/?peid=8cd235b1699a457f9c776c12d4925408>
19. Radev, D., Allison, T.: Mead - a Platform for Multidocument Multilingual Text Summarization. In: LREC (2004)
20. Radev, D., Jing, H., Stys, M., Tam, D.: Centroid-based Summarization of Multiple Documents. Information Proc. and Mgmt. 40, 919–938 (2004)
21. Svore, K.M., Vanderwende, L., Burges, C.J.C.: Enhancing Single-document Summarization by Combining RankNet and Third-Party Sources. In: EMNLP-CoNLL, pp. 448–457 (2007)
22. Verma, R., Filozov, F.: Document Map and WN-Sum: A new framework for automatic text summarization and a first implementation. Technical Report UH-CS-10-03, University of Houston Computer Science Dept. (2010)

# Combining Summaries Using Unsupervised Rank Aggregation

Girish Keshav Palshikar, Shailesh Deshpande, and G. Athiappan

Tata Research Development and Design Centre (TRDDC),

Tata Consultancy Services Limited,

54B, Hadapsar Industrial Estate, Pune 411013, India

{gk.palshikar, shailesh.deshpande, athiappan.g}@tcs.com

**Abstract.** We model the problem of combining multiple summaries of a given document into a single summary in terms of the well-known rank aggregation problem. Treating sentences in the document as *candidates* and summarization algorithms as *voters*, we determine the *winners* in an election where each voter selects and ranks  $k$  candidates in order of its preference. Many rank aggregation algorithms are supervised: they discover an optimal rank aggregation function from a training dataset of where each "record" consists of a set of candidate rankings and a model ranking. But significant disagreements between model summaries created by human experts as well as high costs of creating them makes it interesting to explore the use of unsupervised rank aggregation techniques. We use the well-known Condorcet methodology, including a new variation to improve its suitability. As voters, we include summarization algorithms from literature and two new ones proposed here: the first is based on keywords and the second is a variant of the lexical-chain based algorithm in [1]. We experimentally demonstrate that the combined summary is often very similar (when compared using different measures) to the model summary produced manually by human experts.

## 1 Introduction

The Web, digital libraries, online newspapers and enterprise document repositories all provide easy access to enormous information. Document summary is an important method which people use to deal with this information overload. Hence automatic techniques for document summarization are receiving increasing attention from researchers. We focus on *single document extract summarization* which identifies the most informative (important) sentences in a given document. The information in a summary should be (a) complete (should cover all important topics discussed in the document); (b) correctly ordered (c) coherent and consistent (e.g., must not include any ambiguities or redundancies); and (d) understandable for humans. Given such nebulous goals, it is not surprising that summary *evaluation* is also a complex task, difficult even for expert humans. There are considerable differences and disagreements among experts about what sentences in any given document are important enough to be included in a summary and why. For example, [2] report an average correlation of 0.2 between two randomly chosen summaries from 50 summaries of the same document. The average overlap for the 542 single document summary pairs in DUC-2002 was only about 47%. [3] report

a much higher level of agreement of news-like documents, though even then the agreement level decreases for longer summaries. We empirically demonstrate that there are considerable disagreements even among different summarization algorithms (not just humans). Suppose we assume that individual summarization algorithms are reasonably "good" (but not always "perfect"), in the sense that the summary produced by each algorithm (for a given document) contains at least some of the important sentences of the document. In that case, it is interesting to ask if the summaries produced by individual summarization algorithms can be systematically and effectively combined to produce another summary which is demonstrably better than the constituent summaries. Such *expert advice combination* (or *ensemble*) techniques are used classification (e.g., bagging, boosting). Informally, a summary combination method should combine the "best" parts of the constituent summaries, thus leveraging strengths of the corresponding summarizations algorithms.

The contributions of this paper are as follows. First, we propose two new algorithms for single-document extract summarization. The first is based on identifying *keywords* in the given document and then selecting sentences which are *rich* in keywords. The second is a variant on the lexical-chain based algorithm in [1]. Next, we select a set of summarization algorithms from the literature (including the above two) and propose a method to produce a combined summary from the individual summaries produced by these algorithms. We map this problem to the well-known rank aggregation problem, where by treating sentences in the document as *candidates* and summarization algorithms as *voters*, the goal is to determine the *winners* in this election where each voter selects and ranks  $k$  candidates in order of its preference;  $k$  is a user-defined fixed *summary length* (number of sentences). Many rank aggregation algorithms developed in IR community are supervised: they discover an optimal rank aggregation function from a training dataset of where each "record" consists of a set of candidate rankings and a model ranking. But significant disagreements between model summaries created by human experts (mentioned above) as well as high costs of creating them makes it interesting to explore the use of unsupervised rank aggregation techniques. We propose the use of Condorcet methodology, well-known in politics and sociology, along with our variation to improve its suitability for summary combination. We experimentally demonstrate that the combined summary is "good" i.e., it is often very similar to the model summary produced manually by human experts, when compared using different similarity measures. This paper is organized as follows. Section 2 outlines the related work. Section 3 contains our variant of the lexical-chain based summarization algorithm. Section 4 contains a keyword-based approach to summarization. In section 5, we discuss the use of Condorcet methodology to combine multiple summaries into a winner summary. Section 6 discusses some evaluation of the proposed algorithms. Section 7 contains conclusions and outlines further work.

## 2 Related Work

Many different approaches for text summarization have been proposed for single document extract summarization - see [4] for an overview - we review only some relevant ones. Some algorithms [5], [6], [7] use the presence of *cue phrases*, such as

significantly, as an indication that the sentence may be important. [6] train a Bayesian classifier from a corpus of documents and associated summaries, using sentence position and cue phrases (among others) as features. Presence of words which are *somewhat* frequent is also a characteristic of important sentences [8]. Our keyword-based summarization algorithm can use any suitable technique for identifying keywords [9], [10], [11]; we used [12]. [13] presents a keyword-based summarization algorithm that uses a sentence-score based on the number and percentage of keywords in a sentence. [14] represent a document as a bipartite graph and combine keyphrase extraction and summarization using the principle of mutual reinforcement and sentence clustering; see also [15].

Several summarization approaches use some explicit measure of coherence in the text, usually based on lexical chains. A *lexical chain* [16] is a sequence of related words (usually nouns) occurring in a sequence of sentences in the document. Treating each chain as a concept or topic, one can compute importance of sentences based on the chains that pass through them, selecting the most important ones as a summary; e.g., [17], [18], [19].

Intrinsic evaluation techniques (see [20] for an overview) measure the amount of information from the original document retained in a summary. [2] use factoids, which are atomic facts, as information units to compare model and peer summaries. [21] use Jensen-Shannon divergence as a measure of distance between the two probability distributions over words in the model and peer summaries. [22] extend this approach to evaluate a peer summary (without any model summary) by comparing it with the input document. [23] automatically identify and use (for summary evaluation) summary content units (SCU) which are small sequences of words whose weights depends on the frequency of their appearance in multiple summaries; see [24] for a similar approach called basic elements. [25] propose the relative utility method for summary evaluation, in which each sentence in a document is given a score by a set of human judges about including that sentence in a summary.

Several unsupervised rank aggregation methods have been developed; e.g., median-based [26] and Markov chain based [27]. There is a large amount of work on supervised rank aggregation methods; see the tutorial<sup>1</sup> by S. Agarwal for an overview.

### 3 Keyword-Based Summarization

In this section, we propose a new keywords-based method for single document summarization. Given a set of characteristic keywords for a document, the idea is to select those sentences which contain maximum number of keywords. A document - e.g., an article, a research paper or a news item - is typically characterized by a set of *keywords* or *key phrases*. Each keyword indicates an important aspect (e.g., concept or topic) of the subject matter discussed in a document. Typically, only a few keywords ( $\leq 20$ ) are associated with a document, unlike the large number of *index terms* used in information retrieval to index a document in a collection. Moreover, the keywords are usually *ordered* in decreasing order of their importance, or in increasing order of their generality (more specific keywords first).

<sup>1</sup> <http://web.mit.edu/shivani/www/Events/SDM-10-Tutorial/sdm10-tutorial.pdf>

No.	#Keywords	Keywords
1	4	ship, fire, cruise, Mexico
2	9	ship, fire, coast, cruise, port, aboard, Wednesday, engine, room
3	7	ship, coast, guard, Diaz, scandinavian, star, Miami
4	3	ship, fire, Diaz
5	4	fire, coast, guard, aboard
6	2	fire, Diaz
7	2	ship, Wednesday
8	2	ship, Miami
9	6	fire, cruise, scandinavian, star, Wednesday, Mexico
10	3	fire, engine, room
11	3	Diaz, engine, room
12	2	coast, guard
13	1	fire
14	1	fire
15	0	
16	6	ship, fire, port, scandinavian, star, Miami
17	0	
18	3	coast, guard, cruise

(b)

**Fig. 1.** (a) Algorithm KWSummary (b) Keywords in example sentences

Any keyword extraction algorithm can be used - we have used [12] - to extract a set  $W$  of  $m$  keywords from a given document  $D$ , where  $m$  is a user-specified positive integer. The idea is to represent the given document as an undirected graph, whose vertices are words in the document and the edges are labeled with a dissimilarity measure between two words, derived from the frequency of their co-occurrence in the document. The central vertices in this graph, identified using centrality measures such as eccentricity or betweenness, are returned as keywords. Algorithm KWSummary (Fig. 1(a)) simply ranks (in descending order) the sentences in  $D$  in terms of the number of keywords in  $W$  that they contain and returns the first  $k$  sentences. In case of ties, the algorithm can be modified to apply a criterion such as sentence length or can choose the sentence having most different keywords than those in the sentences chosen so far. Since different keywords have different importance (i.e., the keywords can be ranked or weighted), the algorithm can be easily modified to use (as sentence score) the sum of the weights (or ranks) of the keywords occurring in a sentence.

Suppose the set of  $m = 15$  keywords for the document in Figure 2 is  $W = \{\text{ship, fire, coast, guard, Diaz, cruise, port, Scandinavian, star, aboard, Miami, Wednesday, engine, room, Mexico}\}$ . We treat any variation of these keywords (e.g., plural, past tense etc.) as the occurrence of the same keyword. Keywords occurring in each sentence are shown in Fig. 1(b). Algorithm KW-Summary identifies sentences 2, 3, 16, 9 as the summary since they contain the highest number of keywords. Algorithm KWSummary is expected to be sensitive to (a) the set of keywords  $W$ ; and (b) the number  $k$  of keywords used. While this is broadly true, experimentally, we found that the algorithm is quite robust to reasonable changes in both the set  $W$  and the number  $k$ . For example, for the same keyword extraction algorithm and the same set of documents, we found that the summary sentences chosen by KWSummary do not vary much over values of  $k$  from 10 to 20.

1. Fire Disables Cruise Ship in Gulf of Mexico
2. An engine room fire Wednesday disabled a cruise ship off the Mexican coast, but the crew extinguished the blaze and the ship with 715 people aboard was towed to port.
3. The Navy cutter Vigilant escorted the 465-foot ship, the Scandinavian Star, as a precaution, according to Chief Petty Officer Luis Diaz of the U.S. Coast Guard in Miami.
4. "The fire is out and the ship is being towed by the Mexican navy to Cancun," said Diaz.
5. There were no injuries from the fire, but a 71-year-old St. Petersburg man who suffered a heart attack was in stable condition and on his way home aboard a Coast Guard plane.
6. A second person who suffered a spinal cord injury was also evacuated with the heart attack victim, but Diaz said that injury also was apparently not related to the fire.
7. The ship was expected to arrive in Cancun late Wednesday or Thursday.
8. The 449 passengers will be flown to St. Petersburg, said Jill DeChello, spokesman for the ship's owner, SeaEscape Ltd. of Miami.
9. The Scandinavian Star was on a 3-day cruise to Cozumel, Mexico, and was returning to St. Petersburg when the fire broke out about 1 a.m. EST Wednesday, she said.
10. The crew sealed off the engine room and pumped in carbon dioxide to put out the fire.
11. "They used all their CO<sub>2</sub> and closed off the engine room," said Diaz.
12. "Then they requested more assistance from the Coast Guard."
13. The fire was already extinguished when the Vigilant arrived about 10 a.m., he added.
14. Ms. DeChello said the company did not know what sparked the fire.
15. "It will take a week before we know the cause and the extent of the damage," said Ms. DeChello.
16. In August 1984, a fire struck the Star's sister ship, the Scandinavian Sun, in the port of Miami, killing a passenger and a crewman.
17. "I feel they are two totally different incidents that are in no way related," said Ms. DeChello.
18. She said the cruise line was inspected by the Coast Guard and rigorously enforced all safety rules.

**Fig. 2.** Document D21d\_AP880316-0208.txt from DUC-2001 corpus

## 4 Summarization Using Lexical Chains

A *lexical chain* [16] is a sequence of related words that occurs across sentences in a given document. Two consecutive words in a chain may be related by the *strong relation* if both words are the same or related through synonymy, hypernymy, meronymy or holonymy; e.g., *fire* and *fire*, *fire* and *flame*, or *fire* and *blaze*. They have a *medium-strength relation* if they have a common ancestor in WordNet hierarchy and the path between them is at most of length  $K$  (for some constant  $K$ ); e.g., *apple* and *orange* have the common ancestor *edible fruit* and are connected through the paths *apple* ISA *edible fruit* and *orange* ISA *citrus fruit* ISA *edible fruit*. Several algorithms have been developed for identifying lexical chains in a given document; we use the one in [1]. Often, only nouns are included in a lexical chain, because the semantic hierarchy is much better developed for nouns. Several summarization algorithms are based on lexical chaining [17], [18], [19]. The general idea is to compute the chains for the given document, score the chains based on their cohesiveness, score the sentences based on the score of the chains passing through them, and then choose the most important sentences as the summary. We now discuss our algorithm for lexical chain based summarization.

Consider a chain  $C = \{u_1, u_2, \dots, u_n\}$  consisting of  $m$  words. Let  $DOM(C) = \{w_1, w_2, \dots, w_n\}$  denote the set of  $n$  distinct words in  $C$  ( $C$  may contain multiple occurrences of some words). The score  $G_i$  of the  $i$ -th word  $w_i$  is computed as  $G_i = n_{i,1} \cdot a_1 + n_{i,2} \cdot a_2 + \dots$  where  $n_{i,j}$  denotes the number of words in chain  $C$  that are related to  $w_i$  by  $j$ -th relationship ( $j = 1$ :repetition,  $j = 2$ :synonym,  $j = 3$ :hypernym, hyponym, meronym, holonym etc.) and  $a_j$  is the weight for  $j$ -th relationship (we use  $a_1 = 1.0$ ,  $a_2 = 0.9$ ,  $a_3 = 0.7$ ). We have assumed that medium strength relation is not used when forming lexical chains. Score of the entire chain is then defined as the sum of the scores of the words in it:  $H_C = G_1 + G_2 + \dots + G_n$ . The intuition is that a chain that includes more occurrences of stronger bonds between two words should score more than a chain that has many weaker bonds. We do not consider the length of the chain explicitly as it is already taken into account when calculating the word scores. Higher chain score indicates a more important and more cohesive chain. As an example, consider the chain  $C = \{\text{wind, squall, wind, wind}\}$  ( $m = 4$ ,  $DOM(C) = \{\text{wind, squall}\}$ ,  $n = 2$ ). The score for the word **wind** in the chain's domain is  $G_1 = \text{number of repetitions} * 1.0 + \text{number of synonyms} * 0.9 + \text{number of secondary relations} * 0.7 = 2 * 1 + 0 * 0.9 + 1 * 0.7 = 2.7$ . Similarly, the score for the word **squall** is  $G_2 = 0 * 1 + 0 * 0.9 + 1 * 0.7 = 0.7$ . The score of the entire chain is  $H_C = 2.7 + 0.7 = 3.4$ . We treat chains as concepts and chain score as the importance of a particular concept in the given document. The score of a sentence is obtained by adding the scores of all chains passing through that sentence. We sort the sentences in descending order of their score and then select top  $k$  sentence for the extract summary. The document in Figure 2 has the following 4 chains (sentence number is the subscript for each word in the chain):

$C_1 : \text{fire}_1, \text{fire}_2, \text{blaze}_2, \text{fire}_4, \text{fire}_5, \text{attack}_5, \text{attack}_6, \text{fire}_6, \text{fire}_9, \text{fire}_{10}, \text{fire}_{13}, \text{fire}_{14}, \text{fire}_{16}$   
 $C_2 : \text{ship}_1, \text{ship}_2, \text{ship}_2, \text{ship}_4, \text{ship}_7, \text{ship}_8$   
 $C_3 : \text{coast}_2, \text{coast}_3, \text{coast}_5, \text{coast}_{12}, \text{coast}_{18}$   
 $C_4 : \text{guard}_3, \text{guard}_5, \text{guard}_{18}, \text{safety}_{18}$

The scores for these 4 chains are 12.8, 6, 5, 4.4 (the seed word of the chain - e.g., **ship** in  $C_2$  - is treated as an instance of the extra-strong relation). The sentence scores can now be easily computed; e.g., the score for sentence 2 is  $13.8 + 6 + 5 + 4.4 + \dots = 41.8$  since  $C_1, C_2, C_3$  and  $C_4$  (among others) pass through it. This algorithm chooses sentences 2, 9, 5, 3 as the summary. Our algorithm for finding lexical chains is very similar to that used by [1] with following differences. We do not consider medium strength relations while forming lexical chains, because such relations tend to form spurious chains and thereby dilute the unity of a concept. We also do not consider co-occurrence relations for reducing computational complexity of the chaining algorithm. We do not consider the sentence distance constraint as well. Sentence constraints are useful in finding topics boundaries or intentional reoccurrence of the topics. For summarization, we do not want to split the chains (concepts) into multiple chains because we want to find a concept spans multiple regions (segments) in the document. If we use sentences boundaries, we need to combine similar chains again which is wasteful for summarization. [18] has also shown that chaining improves when the entire document is considered, rather

ID	Method	Sentences	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	AlphaGreedy	3, 5, 2, 17	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	AlphaSVD	3, 5, 2, 17	2	4	0	3	5	3	5	5	5	5	5	5	5	5	5	5	5	5
3	LexRank	16, 15, 14, 8	3	4	2	0	5	4	5	5	5	4	5	5	5	5	5	4	5	5
4	Microsoft Word 2007	1, 2, 16, 3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Lexical Chains based	2, 9, 5, 3	5	3	2	1	3	0	3	3	2	3	3	3	3	3	3	3	3	3
6	KWSummary	2, 3, 16, 19	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	0	3	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	4	2	0	5	4	5	5	5	4	5	5	5	5	5	5	5	4	5	5
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	2	1	3	0	3	3	2	3	3	3	3	3	3	3	3	3	3	3
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1
9	2	0	1	2	2	2	2	2	0	2	2	2	2	2	2	1	2	2	2
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
16	2	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3
17	2	0	0	2	0	2	2	2	2	2	2	2	2	2	2	2	2	0	2
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

**Fig. 3.** (a) Ranked output of summarization methods. (b) Rank matrix for 6 methods and 18 sentences.

than document segments. Our chain scoring mechanism is almost the same as the one in [28], except that we do not consider the penalty part when computing the sentence score.

## 5 Combining Summaries

Given a document  $D$  and  $M$  extract summaries for it (each containing  $k$  sentences) produced by  $M$  summarization algorithms, we now consider the problem of systematically and effectively combining these  $M$  summaries to produce a final summary for  $D$ .

**Definition 1.** Let a document  $D$  contain  $N$  sentences, numbered from 1 to  $N$ . Let  $\sigma_i = \langle S_{i_1}, S_{i_2}, \dots, S_{i_K} \rangle$  denote the ranked output of  $i$  - th summarization method, where each  $1 \leq i_k \leq N$  denotes a sentence number. The rank of sentence number  $j$  in  $\sigma_i$ , denoted  $r(\sigma_i, j)$ , is the position at which that sentence occurs in  $\sigma_i$ . If sentence number  $j$  does not occur in  $\sigma_i$  then its rank is set to some large number (say, 100). A method  $i$  prefers sentence  $x$  over sentence  $y$  if  $r(\sigma_i, x) < r(\sigma_i, y)$  i.e.,  $x$  appears earlier than  $y$  in the ranked output of method  $i$ . Let  $S = \{\sigma_1, \sigma_2, \dots, \sigma_M\}$  denote the collection of the outputs of  $M$  summarization algorithms.

Fig. 3(a) shows the  $k = 4$  sentences selected by each of  $M = 6$  summarization algorithms applied to the example document in Figure 2 containing  $N = 18$  sentences;  $\sigma_1 = \langle 3, 5, 2, 17 \rangle$ ,  $\sigma_3 = \langle 16, 15, 14, 8 \rangle$  and so on. The rank of sentence 5 in  $\sigma_1$  is  $r(\sigma_1, 5) = 2$ , since 5 occurs at position 2 in  $\sigma_1$ . Method 1 prefers sentence 3 to sentence 5. Note that several sentences are identified by more than one method; e.g., sentences 2 and 3 are identified by 5 methods each, sentences 5 and 16 by 3 methods each and sentences 9 by 2 methods. Thus sentences 2 and 3 have a stronger claim to being in the "true" summary of the given document, over other sentences. The question now is:

how do we systematically combine the sentences selected by these 6 summarization algorithms into a final "winning" summary of 4 sentences?

Borda count is a standard unsupervised rank aggregation method which computes the importance of each sentence based on a weighted average of its ranks in different summaries. As another unsupervised rank aggregation method to combine outputs of different summarization algorithms, we propose to use a well-known method called the *Condorcet algorithm*, which is often used to decide winner in an election. As the first step, we form an  $N \times N$  matrix  $R$ , where  $N$  is the number of sentences in the given document  $D$  ( $N = 18$  in the example). The  $i$ -th row of  $R$  corresponds to the  $i$ -th sentence in  $D$ . We consider each of the  $M = 6$  summarization methods as a voter. Entry  $R_{i,j}$  in  $R$  indicates the total number of voters (out of  $M$ ) who prefer sentence  $i$  over sentence  $j$ . The matrix  $R$  for the example document is shown in Fig. 3(b). For example,  $R_{9,5} = 2$  since 2 methods (5 and 6) prefer sentence 9 over sentence 5.

Condorcet method analyzes the votes of  $M$  summarization methods, as represented in the matrix  $R$ , to decide the *winner* among the  $N$  candidate sentences. For this purpose, it considers  $N \times N$  imaginary contests, pitting every sentence  $i$  against every other sentence  $j$ . Sentence  $i$  is a winner of this contest if the number of voters that prefer  $i$  is greater than the number of voters that prefer  $j$ . In the example, sentence 2 wins against sentence 5 because  $R(2, 5) > R(5, 2)$ . When all possible pairings of the sentences are considered, we pick up the  $k$  sentences that have won the maximum number of contests. In the example, the number of wins for each sentence are: 1:8 2:17 3:16 4:0 5:13 6:0 7:0 8:8 9:12 10:0 11:0 12:0 13:0 14:9 15:10 16:14 17:12 18:0. Thus sentence 9 wins against 12 sentences, viz., 1, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 18. Sentences 2, 3, 16 and 5 constitute a winning summary according to this method. Note that for  $k = 5$ , there would have been a tie for sentences 9 and 17, as both won against 12 sentences each. Ties can be broken by means of well-known variations of the Condorcet method; e.g., ranked pairs or Schulze method.

## 6 Evaluation

We used DUC 2001 dataset of 100 news articles. We selected  $N = 100$  documents for which model summaries were available in this repository. These 100 documents are news stories related to natural hazards, politics, financial results and so on. The average number of sentences in these documents is 34. For each of these  $N = 100$  documents, we created extract summaries with different number of sentences ( $k = 3, 4, \dots, 10$ ), using the following summarization methods: IBM's Many Aspects Alfa-Greedy, IBM's Alfa-SVD [29]<sup>2</sup>, LexRank [30], Microsoft Word 2007<sup>3</sup>, the lexical chain based and keyword-based summarization algorithm in this paper.

Then for each document and each summary length ( $k = 3, 4, \dots, 10$ ), we created two additional summaries by combining the summaries generated by the above summarization algorithms, one using Condorcet and the other using the Borda count method. The main question is: do these two unsupervised rank aggregation methods produce

<sup>2</sup> <http://www.alphaworks.ibm.com/tech/manyaspects>

<sup>3</sup> <http://office.microsoft.com/en-us/word-help/automatically-summarize-a-document-HA010255206.aspx>

“better” summaries than the 5 peer summaries on which they are based? Among various possibilities, we adopt the following simple way to answer this question: we say that one summary is better than another if it is *closer* (i.e., more similar) to the model (reference) summary.

For each document and for each summary length ( $k = 3, 4, \dots, 10$ ), we compare the  $5 + 2 = 7$  peer summaries (prepared by the summarization algorithms) with the model summary for that document using the following similarity measures: Dice, Jaccard and cosine. For summaries of length  $k$ , we retained only the first  $k$  sentences in the model summary. Let  $P_{i,j,k}$  denote the peer summary of a particular document  $1 \leq i \leq N$  ( $N = 100$  here), produced by the summarization method  $1 \leq j \leq 7$  having summary length  $3 \leq k \leq 10$ . Let  $M_{i,j,k}$  denote the corresponding model summary. Let  $sim_{dice}(P_{i,j,k}, M_{i,j,k})$ ,  $sim_{jacc}(P_{i,j,k}, M_{i,j,k})$  and  $sim_{cos}(P_{i,j,k}, M_{i,j,k})$  denote the similarity between  $P_{i,j,k}$  and  $M_{i,j,k}$  computed using the above similarity measures. The performance of a particular summarization method (for a particular summary length) is measured by the average similarity between the model summaries and the summaries produced by that method. Note that this performance measure depends on the similarity method used.

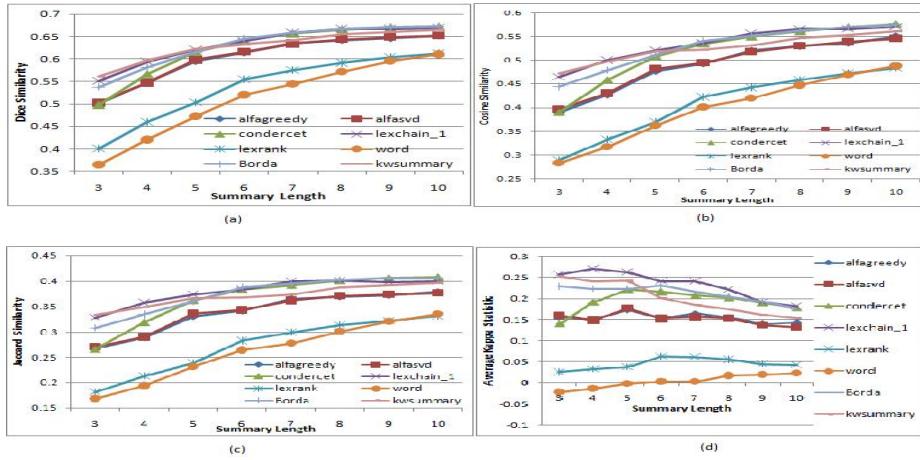
$$a_{dice}(j, k) = \frac{\sum_{i=1}^N sim_{dice}(P_{i,j,k}, M_{i,j,k})}{N}$$

$$a_{jacc}(j, k) = \frac{\sum_{i=1}^N sim_{jacc}(P_{i,j,k}, M_{i,j,k})}{N}$$

$$a_{cos}(j, k) = \frac{\sum_{i=1}^N sim_{cos}(P_{i,j,k}, M_{i,j,k})}{N}$$

Fig. 4(a)-(c) shows how the average performance of each of the 7 summarization methods (as computed using a particular similarity measure) varies with the summary length  $3 \leq k \leq 10$ . For most summarization methods, the performance improves with the summary length. Alfa-Greedy and Alfa-SVD show a slower increase in their performance. On the other hand, Condorcet and Borda count methods show steep rise in performance. Both these voting methods show consistently higher performance compared to other methods for summary lengths 5 or more. The next best performance is shown by the Lexical Chaining algorithm.

*Kappa statistic* is a well-known way to measure agreement between two raters (or annotators), when they both qualitatively evaluate the same set of objects. Suppose each object is given a rating 0 or 1 by each rater. Then the kappa statistic is computed as  $\kappa = \frac{P(a) - P(e)}{1 - P(e)}$  where  $P(a)$  is the observed agreement level (i.e., the fraction of objects for which both raters gave the same rating) and  $P(e)$  is the hypothetical probability of chance agreement. If both raters agree on all objects then  $\kappa = 1$ . If the agreement between raters is as much as expected under chance then  $\kappa = 0$ . Higher values tend to indicate better agreement levels. We adapt the kappa statistic to compute the agreement between the model summaries and the summaries produced by a summarization algorithms. The goal is to investigate whether a combined summary produced by summary combination method (such as Condorcet) has a better agreement with the model summary.



**Fig. 4.** Performance of summarization methods using different measures

We treat the model summary creator as one rater and a particular summarization algorithm as another rater. The document in Figure 2 has 18 sentences. Each rater assigns rating 1 or 0 to each sentence, indicating whether or not the she includes that sentence in the summary. The model summary  $A$  has sentences 2, 3, 5, 9 whereas the Condorcet method's summary  $B$  has sentences 2, 3, 5, 16. Since the two raters agree on 16 out of 18 sentences,  $P(a) = 16/18 = 0.89$ . Both raters assign 1 to 4 sentences and 0 to 14 sentences. Thus the probability that both raters assign 1 by chance is  $\frac{4}{18} \times \frac{4}{18} = 0.049$ . The probability that both raters assign 0 by chance is  $\frac{16}{18} \times \frac{16}{18} = 0.605$ . Then  $P(e) = 0.049 + 0.605 = 0.654$ . The kappa statistics for the agreement between the model summary and the condorcet summary for this document is  $\kappa = \frac{0.89 - 0.654}{(1 - 0.654)} = 0.679$ . We can now get the kappa values for all 100 documents and their average value indicates the average agreement level between the model summaries and the Condorcet summaries. Fig. 4(d) shows the average kappa statistic for all 7 summarization algorithms for different summary lengths ( $k = 3 \dots 10$ ). Both unsupervised summary combination methods (Condorcet and Borda count) tend to have a better agreement with the model summary than the individual summaries, with the exception of Lexical chain and KWSummary algorithms.

## 7 Conclusions and Further Work

This paper's main contributions are as follows. First, we proposed two simple algorithms for single document extract summarization. The first is based on identifying keywords in the given document and then selecting sentences which are *rich* in keywords. The second summarization algorithm is a variant on the lexical-chain based algorithm in [1]. Our second contribution is initiating an investigation into whether summaries produced by individual summarization algorithms can be combined systematically and effectively using some unsupervised rank aggregation method to produce a

summary which tends to be "better" than the individual summaries. Treating sentences in the document as *candidates* and summarization algorithm as *voters*, we formulated this problem as determining the *winners* in an election where each voter selects and ranks  $k$  candidates in order of its preference. We proposed the use of well-known Condorcet methodology for this task. We experimentally demonstrated that the combined summary is often very similar (when compared using different measures) to the model summary produced manually by human experts.

Experimentally, we found that both the new proposed summarization methods (keyword-based and lexical-chain based) tend to produce good summaries. Further, we found that both the unsupervised rank aggregation methods (Condorcet and Borda count) tend to produce good summaries, which are generally better than most of the individual summaries. These methods seem to work reasonably well for documents in a variety of domains. The proposed unsupervised summary combination methods are inherently language-independent (do not use any language-specific knowledge) and hence more widely usable. For further work, we plan to use more voters (i.e., more summarization methods). We are planning a more extensive validation with larger and more varied document repositories. An interesting alternative is to use another combination method, like Dempster-Schafer evidence combination. Another research direction is to extend the use of these methods for multi-document summarization.

## References

1. Stokes, N.: Applications of Lexical Cohesion in the Topic Detection and Tracking Domain, Ph.D. thesis, National University of Ireland, Dublin (2004)
2. van Halteren, H., Teufel, S.: Examining the consensus between human summaries: initial experiments with factoid analysis. In: Proceedings of the HLT-NAACL 2003 on Text Summarization Workshop (HLT-NAACL-DUC 2003), vol. 5, pp. 57–64 (2003)
3. Jing, H., Barzilay, R., McKeown, K., Elhadad, M.: Summarization evaluation methods: Experiments and analysis. In: AAAI Symposium on Intelligent Summarization, pp. 60–68 (1998)
4. Hovy, E.H.: Automated text summarization. In: Mitkov, R. (ed.) *The Oxford Handbook of Computational Linguistics*, pp. 583–598 (2005)
5. Teufel, S., Moens, M.: Sentence extraction as a classification task. In: Proc. Workshop on Intelligent Scalable Summarization ACL/EACL Conference, pp. 58–65 (1997)
6. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proc. 18th Int. ACM Conf. Research and Development in Information Retrieval (SIGIR), pp. 68–73 (1995)
7. Hovy, E., Lin, C.Y.: Automated text summarization in summarist. In: Maybury, M., Mani, I. (eds.) *Advances in Automatic Text Summarization*. MIT Press (1999)
8. Edmundson, H.P.: New methods in automatic extraction. *Journal of the ACM* 16(2), 264–285 (1968)
9. Matsumura, N., Ohsawa, Y., Ishizuka, M.: Pai: Automatic indexing for extracting assorted keywords from a document. In: Proc. AAAI 2002 (2002)
10. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on AI Tools* 13(1), 157–169 (2004)
11. Ohsawa, Y., Benson, N.E., Yachida, M.: Keygraph: automatic indexing by co-occurrence graph based on building construction metaphor. In: Proc. Advanced Digital Library Conference (ADL 1998), pp. 12–18 (1998)

12. Palshikar, G.: Keyword Extraction from Single Document Using Centrality Measures. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 503–510. Springer, Heidelberg (2007)
13. Bouras, C., Poulopoulos, V., Tsogkas, V.: Perssonal's core functionality evaluation: Enhancing text labeling through personalized summaries. *Data and Knowledge Engineering* 64(1), 330–345 (2008)
14. Zha, H.: Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: Proc. 25th Int. ACM Conf. Research and Development in Information Retrieval (SIGIR), pp. 113–120 (2002)
15. Wan, X., Yang, J., Xiao, J.: Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: ACL, pp. 552–559 (2007)
16. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17, 21–48 (1991)
17. Alam, H., Kumar, A., Nakamura, M., Rahman, F., Tarnikova, Y., Wilcox, C.: Structured and unstructured document summarization: Design of a commercial summarizer using lexical chains. In: Proc. Seventh Int. Conf. Document Analysis and Recognition (ICDAR 2003), pp. 1147–1152 (2003)
18. Barzilay, R., Elbadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, pp. 10–17 (1997)
19. Mani, I., Bloedorn, E., Gates, B.: Using cohesion and coherence models for text summarization. In: Using Cohesion and Coherence Models for Text Summarization, pp. 69–76 (1998)
20. Nenkova, A.: Summarization evaluation for text and speech: issues and approaches. In: Ninth International Conference on Spoken Language Processing, INTERSPEECH 2006 (2006)
21. Lin, C.Y., Cao, G., Gao, J., Nie, J.Y.: An information-theoretic approach to automatic evaluation of summaries. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL 2006), pp. 463–470 (2006)
22. Louis, A., Nenkova, A.: Automatic summary evaluation without human models. In: Proc. of Text Analysis Conference, TAC 2008 (2008)
23. Nenkova, A., Passonneau, R., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.* 4 (2007)
24. Hovy, E., Lin, C.Y., Zhou, L., Fukumoto, J.: Automated summarization evaluation with basic element. In: Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006) (2006)
25. Radev, D.R., Tam, D.: Summarization evaluation using relative utility. In: Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM 2003), pp. 508–511 (2003)
26. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proc. of 2003 ACM SIGMOD Int. Conf. on Management of Data, pp. 301–312 (2003)
27. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proc. of 10th Int. World Wide Web Conference, pp. 613–622 (2001)
28. Kang, B.Y.: A novel approach to semantic indexing based on concept. In: Proc. 41st Annual Meeting of Association of Computational Linguistics (ACL 2003), vol. 2, pp. 44–49 (2003)
29. Liu, K., Terzi, E., Grandison, T.: Manyaspects: a system for highlighting diverse concepts in documents. In: Proc. Int. Conf. Very Large Databases (VLDB), pp. 1444–1447 (2008)
30. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of AI Research* 22, 457–479 (2004)

# Using Wikipedia Anchor Text and Weighted Clustering Coefficient to Enhance the Traditional Multi-document Summarization

Niraj Kumar, Kannan Srinathan, and Vasudeva Varma

IIIT-Hyderabad, Hyderabad-500032, India

niraj\_kumar@research.iiit.ac.in, {vv, srinathan}@iiit.ac.in

**Abstract.** Similar to the traditional approach, we consider the task of summarization as selection of top ranked sentences from ranked sentence-clusters. To achieve this goal, we rank the sentence clusters by using the importance of words calculated by using page rank algorithm on reverse directed word graph of sentences. Next, to rank the sentences in every cluster we introduce the use of weighted clustering coefficient. We use page rank score of words for calculation of weighted clustering coefficient. Finally the most important issue is the presence of a lot of noisy entries in the text, which downgrades the performance of most of the text mining algorithms. To solve this problem, we introduce the use of Wikipedia anchor text based phrase mapping scheme. Our experimental results on DUC-2002 and DUC-2004 dataset show that our system performs better than unsupervised systems and better than/comparable with novel supervised systems of this area.

**Keywords:** Multi-document summarization, sentence clusters, weighted clustering coefficient, page rank, and Wikipedia anchor text.

## 1 Introduction

The generic summaries reflect the main topics of the document without any additional clues and prior knowledge. According to [5], generic summaries outperform over (1) query-based and (2) hybrid summaries in the browsing tasks, so the document context of generic summaries help users in browsing.

These days digital libraries and internet etc. contain huge amount of text resources, like: Text articles, web pages, News documents, Educational materials etc. These all again contain huge amount of information and we have less time to go through. It is remarkable to note that all such documents do not always contain human supplied summaries. We believe that an unsupervised approach to generate extract summary by using limited linguistic resources is essential. It improves the quick access of large quantities of such information. Finally, the uses of learning /training based systems make us dependent on corpus or dataset.

That's why we focus our attention towards the development of an unsupervised generic Multi-document summarization system, which can generate high quality extract summary without using heavy linguistic resources and learning/training.

## 1.1 Related Work

A lot of methods have been proposed for multi-document summarization. The most frequently used techniques among all proposed methods are the use of sentence vector representation (where each row represents a sentence and each column represents a term) and graphs based methods (where each node is a sentence and each edge represents the pair wise relationship among corresponding sentences). Finally all these methods rank the sentences according to their scores calculated by a set of predefined features, such as term frequency inverse sentence frequency (TF-ISF) [16]; [14], sentence or term position [20], and number of keywords [20].

Some state of the art methods with key features are: centroid-based methods (e.g., MEAD [16]), graph-ranking based methods (e.g., LexPageRank [10]), non-negative matrix factorization (NMF) based methods (e.g., [11]), Conditional random field (CRF) based summarization [18], and LSA based methods [11].

## 1.2 Problem Setup and Motivation

In this section we present some basic issues and problems related to traditional multi-document summarization and basic motivation behind the techniques used to solve it.

**Using Wikipedia Anchor Texts and Documents Titles to Handle Noisy Terms:** Presence of noisy words in documents generally reduces the performance of most of the summarization algorithms. Because several times noisy words get good score with linguistic, statistical or graph theoretical scoring system. However, the use of Tf-Idf (term frequency and inverse document frequency) and word net etc., shows some improvements, but still it requires some more improvements.

To solve this issue, we use the Wikipedia anchor text and titles of documents. With the help of Wikipedia anchor text and titles of documents, we identify the informative terms from given documents. The anchor texts in Wikipedia have great semantic value, i.e. they provide alternative names, morphological variations and related phrases for target article.

This step has two benefits: (1) It reduces the chances of getting high importance by noisy words and (2) improves the performance of overall system.

**Using Page Rank Score on Reverse Directed Word Graph of Sentences to Rank the Sentence Clusters:** Use of sentence clusters in multi-document summarization is not new. We use GAAC (group average agglomerative clustering algorithm) to cluster the sentences. To rank the identified sentence clusters, we use page rank score of words, calculated on reverse directed word graph of sentences. This scheme helps in effective ranking of words through voting. In general writing behaviour, we describe the term after writing it. The page rank score on reverse directed word graph of sentences effectively captures it.

**Use of Weighted Clustering Coefficient:** Use of weighted clustering coefficient helps us in identifying the strength of ties with strong nodes. Before going into detail, we first describe the clustering coefficient and then describe the requirement of weighted clustering coefficient.

The clustering coefficient is a measure of degree to which nodes in a graph tends to cluster. There are two types of clustering coefficients:

- a) Global Clustering coefficient: It is designed to give an overall indication of the clustering in the network.
- b) Local Clustering Coefficient: It gives the indication of embeddedness of single node.

We use the notion of local clustering coefficient. It can be defined as:

- a) In undirected network the local clustering coefficient  $C(V_i)$  of a node  $V_i$  can be defined as:

$$C(V_i) = \frac{2e(V_i)}{K(V_i)(K(V_i)-1)} \quad (1)$$

Where,

$K(V_i)$ =number of neighbors / degree of  $V_i$  and

$e(V_i)$ =number of connected pairs between all neighbours of  $V_i$

- b) In directed network the local clustering coefficient  $C(V_i)$  of a node  $V_i$  can be defined as:

$$C(V_i) = \frac{e(V_i)}{K(V_i)(K(V_i)-1)} \quad (2)$$

*Main aim behind the use of weighted clustering coefficients:* We believe that each word in document may have different levels of importance (beyond what is captured by degree of node in graph) and therefore we cannot ignore this fact.

The unweighted clustering coefficient obtained by using word graph of sentences, helps us in identifying the embeddedness strength of words with other words in the graph; however, the use of importance of words in clustering coefficients (i.e. weighted clustering coefficient) helps us in identifying the embeddedness strength of words with other important words in the graph. This is a general social networking behaviour, where strength or status of any node or person depends upon (1) strength of that person / node and (2) strength of tie ups with strong friends. By using of page rank of words in calculation of weighted clustering coefficient we tried to achieve both levels of strength.

Our system uses the weighted clustering coefficient score of words to calculate the importance of sentences in sentence cluster. The effective improvements in quality of results also support our view (see sub-section 4.2 for results).

## 2 Framework and Algorithm

### 2.1 Input Cleaning

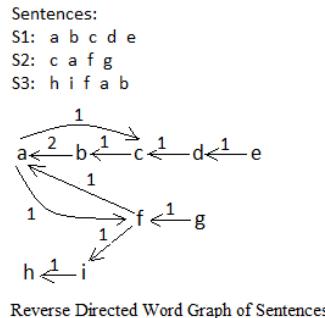
Our input cleaning task includes: (1) removal of noisy entries from entire document collection and (2) sentence filtration. Finally we stem the entire text by using porter stemming algorithm.

## 2.2 Calculation of Importance of Words

The calculation of importance of words is very important, as, we use it to calculate the importance of identified sentence clusters in next step. To calculate the importance of all distinct words of given collection, we concatenate all the documents of given collection and prepare a single file.

Next, we calculate the page rank score of every word on reverse directed word graph of sentences. The way to prepare the reverse directed word graph of sentences and calculation of page rank is given below:

**Preparing Reverse Directed Word Graph of Sentences:** Let, we have a set of sentences i.e.  $S = \{S_1, S_2, \dots, S_n\}$  from given collection. Now, to prepare the reverse directed word graph of sentences, we add reverse directed link for every adjacent word pair of every sentence in the set. See Figure-1. We denote  $G = (V, E)$  as a directed graph, Where,  $V = \{V_1, V_2, \dots, V_n\}$  denotes the vertex set and link set  $(V_j, V_i) \in E$  if there is a link from  $V_j$  to  $V_i$ .



**Fig. 1.** Reverse directed word graph of sentences, Here  $S_1$ ,  $S_2$  and  $S_3$  represents the sentences of document and 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' and 'i' represents the distinct words

**Calculating Page Rank Score:** For any given vertex  $V_i$ , let  $IN(V_i)$  be the set of vertices that point to it (predecessors), and let  $OUT(V_i)$  be the set of vertices that vertex  $V_i$  points to (successors). Then the page rank score of vertex  $V_i$  can be defined as [3]:

$$S(V_i) = \frac{(1 - \lambda)}{N} + \lambda \sum_{j \in IN(V_i)} \frac{S(V_j)}{OUT(V_j)} \quad (3)$$

Where:

$S(V_i)$  = Rank / score of word / vertex  $V_i$ .

$S(V_j)$  = rank/score of word/vertex  $V_j$ , from which incoming link comes to word / vertex  $V_i$ .

$N$  = Count of number of words/vertex in word graph of sentences.

$\lambda$  = Damping factor (we use a fixed score for damping factor i.e., "0.85" as used in [3]).

## 2.3 Preparing Sentence Clusters and Ranking

To identify the topics covered in document we use Group average agglomerative clustering scheme (GACC). In our case the topic is considered as set of sentences related to same concept. Among three major agglomerative clustering algorithms, i.e. single-link, complete-link, and average-link clustering. Single-link clustering can lead to elongated clusters. Complete-link clustering is strongly affected by outliers. Average-link clustering is a compromise between the two extremes, which generally avoids both problems. This is the main reason of use of group average agglomerative clustering algorithm for clustering the sentences.

GACC, uses average similarity across all pairs within the merged cluster to measure the similarity of two clusters. In this scheme average similarity between two clusters (say,  $c_i$  and  $c_j$ ) can be computed as:

$$sim(c_i, c_j) = \frac{1}{|c_i \cup c_j|(|c_i \cup c_j| - 1)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j), \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y}) \quad (4)$$

Where,

$sim(\vec{x}, \vec{y})$  = count of co-occurring words in  $\vec{x}$  and  $\vec{y}$

To apply the GACC on sentences we use a sentence vector representation of documents of entire collection. Here, each row represents a sentence and each column represents a term. In the entire evaluation, we use the threshold “0.4”.

**Calculating Importance of Sentence Clusters or Topics:** To calculate the weighted importance of any sentence cluster or topic, we calculate the sum of weighted importance of all words in the given sentence cluster. The calculation of weighted importance of any sentence cluster can be given as:

$$W(C) = \sum W_{wd} \quad (5)$$

Where

$W(C)$  = weight of given sentence cluster ‘C’

$\sum W_{wd}$  = weight of all words in given sentence cluster. (see sub-section 2.2, eq-3 to calculate the weight of words).

Next, we calculate the percentage of weighted information of every identified sentence cluster. The percentage weighted importance of any identified sentence cluster can be calculated as:

$$\%W(C) = \left( \frac{W(C)}{\sum W(C)} \times 100 \right) \quad (6)$$

Where:

$\%W(C)$  = percentage weight of given sentence cluster ‘C’.

$\sum W(C)$  = sum of weight of all identified sentence cluster.

$W(C)$  = weight of given sentence cluster ‘C’.

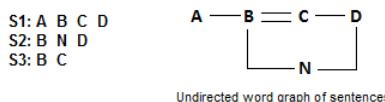
## 2.4 Mapping Phrases by Using Wikipedia Anchor Text

We use Wikipedia anchor text to identify the informative terms in every identified sentence cluster. For this, first of all we fix the phrase boundary. According to scheme defined in [2], we consider stopwords and punctuation marks as phrase boundary. Next, we stem the entire anchor text collection and find the longest matching Wikipedia anchor text sequence in every words sequence within phrase boundary. We repeat this process with every word sequence inside the predefined phrase boundary. We also find the matching words related to titles of entire collection. We remove the rest of the words from every sentence. Thus every sentence in collection contains sequence of Wikipedia anchor texts or words from titles of entire collection. We use this mapping of phrases in calculation of weighted clustering coefficients.

## 2.5 Calculating Weighted Clustering Coefficient

After step 2.4 we have sequence of Wikipedia anchor text words or words from titles of documents, in sentences of every identified sentence cluster. Now, we calculate the weighted clustering coefficient of all such words in every sentence cluster. For this we create undirected word graph of sentences. The sparse nature of word graph of sentences is the main reason behind the selection undirected graph for calculation of weighted clustering coefficient. The process to calculate the weighted clustering coefficient of every distinct word of given sentence cluster is given below:

**Preparing Word Graph of Sentences:** we treat every distinct word as node of graph and prepare undirected word graph of sentences by adding undirected edge for every adjacent words pair.



**Fig. 2.** Undirected word graph of sentences, Here, S1, S2 and S3 denotes the sentences and ‘A’, ‘B’, ‘C’, ‘D’ and ‘N’ denotes the words which are common to Wikipedia anchor text or Title of documents

**Graph Theoretical Notation:** We denote  $G = (V, E)$  as an undirected word graph of sentences. Where,  $V = \{V_1, V_2, \dots, V_n\}$  denotes the vertex set and link set  $(V_j, V_i) \in E$  if there is a link between  $V_j$  and  $V_i$

**Calculating Link Weight:** We use the page rank score of words (See sub-section-2.2, for calculation of weight of every word) in calculation of link weight. The link weight of any edge  $E = (V_i, V_j)$  can be calculated as:

$$W(V_i, V_j) = \left\{ \frac{Score(V_i)}{Degree(V_i)} + \frac{Score(V_j)}{Degree(V_j)} \right\} \times L_c(V_i, V_j) / 2 \quad (7)$$

Where,

$W(V_i, V_j)$  = Link weight of link between nodes  $V_i$  and  $V_j$

$Score(V_i)$  = page rank score of node (word)  $V_i$

$Score(V_j)$ =page rank score of node (word)  $V_j$

$Degree(V_i)$ =degree of node (word)  $V_i$

$Degree(V_j)$ =degree of node (word)  $V_j$

$L_c(V_i, V_j)$ = count of number of links between nodes  $V_i$  and  $V_j$

By using this scheme, we calculate the link weight of every edge of the graph.

**Calculating Weighted Clustering Coefficient:** We use the link weight calculated by using page rank score in calculation of weighted clustering coefficient. In this vein, we maintain the properties of unweighted clustering coefficients on undirected graph (as described in [4]).

- The value of weighted clustering coefficient of any node  $i$  i.e.  $C(\tilde{V}_i) \in [0,1]$ .
- In the unweighted case, the number of triangles at its node determines its clustering property. In the weighted case, clustering should be determined by some weighted characteristic of triangles.
- For each triangle all three edges should be taken into account.
- For each triangle, the weighted characteristic should be invariant to permutation of weight.
- When any of the triangle approaches zero, the weighted characteristic of that triangle should likewise approaches zero. When vertex  $V_i$  participates in the maximum number  $\frac{1}{2}K(V_i)(K(V_i)-1)$  of triangles, where each edge weight is maximal, the weighted clustering coefficient should also be maximal i.e.  $C(\tilde{V}_i)=1$ . To achieve the weighted clustering coefficient [4], replaces  $e(V_i)$  (See Eq-1) by sum of triangle intensities.

Now weighted clustering coefficient of any node  $V_i$  can be defined as:

$$C(\tilde{V}_i) = \frac{2}{K(V_i)(K(V_i)-1)} \sum_{V_j, V_k} \left( \tilde{w}(V_i, V_j) \times \tilde{w}(V_j, V_k) \times \tilde{w}(V_k, V_i) \right)^{1/3} \quad (8)$$

Where,

$$\tilde{w}(V_i, V_j) = \frac{w(V_i, V_j)}{W} \quad (9)$$

$w(V_i, V_j)$ = Link weight of link between nodes  $V_i$  and  $V_j$  (see equation-7).

In these equations 'W' is the maximum of all edge's weight in given graph. The normalization used in above equation and use of sum of triangle intensities fulfil the conditions given in [4].

## 2.6 Ranking Sentences Inside Every Sentence Cluster

To rank the sentences in every sentence cluster, we use the weighted clustering coefficient of words in sentences. We add the weighted clustering coefficient score of words to calculate the weight of sentence. We finally rank the sentences in descending order of their weight. The scheme to calculate the weight of sentences can be given as:

$$Wt(S_r) = \sum WCC(W) \quad (10)$$

Where,

$Wt(S_r)$  = weight of sentence  $S_r$  in given sentence cluster.

$\sum WCC(W)$  = sum of weight of all words (node / vertex) which exist in given sentence  $S_r$  and obtained by using weighted clustering coefficient (see sub-section 2.5, equation-8).

Next, we rank the sentences of given sentence cluster in descending order of their weight.

## 2.7 Generating Extract Summary

To generate the extract summary, we select single top ranked sentence(s) from every identified sentence cluster and arrange them according to the rank of their parent sentence cluster (see sub-section 2.3 for ranking of identified sentence clusters).

If, number of sentence clusters is few, then we use the percentage weight of every sentence cluster to fix the number of required top sentences, which are to be extracted from every sentence cluster. To calculate the percentage weight / importance of any given sentence clusters 'C' we use the following scheme:

$$\%W(C) = \left( \frac{W(C)}{\sum W(C)} \times 100 \right) \quad (11)$$

Where,

$\%W(C)$  = percentage weight of given sentence cluster 'C'.

$\sum W(C)$  = sum of weighted importance of all identified sentence clusters.

$W(C)$  = weight of given sentence cluster 'C'. (see sub-section 2.3, to calculate the weight of any given sentence clusters).

Now, the count of sentences, that is to be extracted from sentence cluster 'C' can be the nearest higher integer value of  $\%W(C) \times$  "Total number of required sentences".

**NOTE:** if the length of sentence is more than 40 words than we discard it and pick the next highest ranked sentence from same sentence cluster.

## 3 Pseudo Code

**INPUT:** ASCII text document.

**OUTPUT:** Required number of extracted sentences as summary. We truncate the final output to meet the required number of words.

**ALGORITHM:**

Step 1. Apply input cleaning (see Subsec-2.1).

Step 2. Calculate the importance/weight of every distinct word of entire text collection (See Subsection-2.2).

- Step 3. Identify all sentence clusters from the given collection and rank every identified sentence cluster in descending order of their importance / score (see sub-section 2.3).
- Step 4. Use Wikipedia anchor text and words from titles of document collection to identify the informative words in every identified sentence cluster (see sub-section-2.4).
- Step 5. Calculate the weighted clustering coefficients of informative words of every identified sentence cluster (See sub-section 2.5).
- Step 6. Use weighted clustering coefficient of informative words to rank the sentences in descending order of their weight, in every identified sentence cluster (See sub-section 2.6).
- Step 7. Apply sentence extraction scheme, to produce the required number of sentences (see sub-section-2.7).

## 4 Evaluation

We have done two different experiments. In first experiment we compare our devised system with state-of-the-art supervised and unsupervised systems. In the second experiment, we test the effect of weighted clustering coefficient. The details of dataset, evaluation metrics and results are given below.

**Details of Dataset:** We use DUC2002 and DUC2004 data sets to evaluate our devised system. DUC dataset is an open benchmark data sets from Document Understanding Conference (DUC) for generic automatic summarization. Table 1 gives a brief description of the dataset.

**Table 1.** Details of DUC 2002, DUC-2004 dataset

	DUC2002	DUC2004
number of document collections	59	50
number of documents in each collection	10	10
data source	TREC	TDT
summary length	200 words	665bytes

**Evaluation metric:** We use ROUGE toolkit (version 1.5.5) to measure the summarization performance. To properly evaluate the summary we use ROUGE-1, ROUGE-2, ROUGE-SU and ROUGE-L based measures. The rest of the details and package is available at [13].

### 4.1 Experiment-1

In this experiment we empirically compare our devised system's result with published results of [6]. The details of system description used in experimental evaluation of [6], is described below:

**Systems Used in Evaluation:** We use the published results of the following most widely used document summarization methods as the baseline systems to compare with our devised system. (1) Random: The method selects sentences randomly for

each document collection (2) Centroid: The method applies MEAD algorithm [16] to extract sentences according to the following three parameters: centroid value, positional value, and first-sentence overlap. (3) LexPageRank: The method first constructs a sentence connectivity graph based on cosine similarity and then selects important sentences based on the concept of eigenvector centrality [10]. (4) LSA: The method performs latent semantic analysis on terms by sentences matrix to select sentences having the greatest combined weights across all important topics [11]. (5) NMF: The method performs non-negative matrix factorization (NMF) on terms by sentences matrix and then ranks the sentences by their weighted scores [12]. (6) KM: The method performs K-means algorithm on terms by sentences matrix to cluster the sentences and then chooses the centroids for each sentence cluster. (7) FGB: The FGB method is proposed in [19]. (8) The published results of BSTM method [6].

**Results:** Results are given in Table-2 and Table-3. Table-2 contains evaluation results on DUC-2002 dataset. Table-3 contains evaluation results on DUC-2004 dataset. The highest evaluation score related to every ROUGE evaluation metric is presented by using bold font. From experimental results (as, given in Table-2 and Table-3), it is clear that our devised system performs better than all unsupervised systems and better/comparable with supervised system like BSTM [6].

**Table 2.** Evaluation results on DUC-2002 dataset

Systems	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU
DUC Best	0.49869	<b>0.25229</b>	0.46803	0.28406
Random	0.38475	0.11692	0.37218	0.18057
Centroid	0.45379	0.19181	0.43237	0.23629
LexPageRank	0.47963	0.22949	0.44332	0.26198
LSA	0.43078	0.15022	0.40507	0.20226
NMF	0.44587	0.16280	0.41513	0.21687
KM	0.43156	0.15135	0.40376	0.20144
FGB	0.48507	0.24103	0.45080	0.26860
BSTM	0.48812	0.24571	0.45516	0.27018
<b>Our System</b>	<b>0.51746</b>	0.24245	<b>0.47252</b>	<b>0.28642</b>

**Table 3.** Evaluation results on DUC-2004 dataset

Systems	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU
DUC Best	0.38224	0.09216	0.38687	0.13233
Random	0.31865	0.06377	0.34521	0.11779
Centroid	0.36728	0.07379	0.36182	0.12511
LexPageRank	0.37842	0.08572	0.37531	0.13097
LSA	0.34145	0.06538	0.34973	0.11946
NMF	0.36747	0.07261	0.36749	0.12918
KM	0.34872	0.06937	0.35882	0.12115
FGB	0.38724	0.08115	0.38423	0.12957
BSTM	0.39065	0.09010	0.38799	0.13218
<b>Our System</b>	<b>0.41413</b>	<b>0.093017</b>	<b>0.39032</b>	<b>0.13846</b>

## 4.2 Experiment-2

We use this experiment to justify the use of weighted clustering coefficient for ranking the sentences in every identified sentence cluster. For this we make simple change and use unweighted clustering coefficient as given in equation-1 in place of equation-8 (see sub-section 2.5) and run the entire system. The comparative results (i.e. with weighted clustering coefficient and with unweighted clustering coefficient) with DUC-2002 and DUC-2004 dataset are given in Figure-3 and in Figure-4 respectively. The results given in Figure-3 and 4, clearly indicates the benefits of using weighted clustering coefficient.

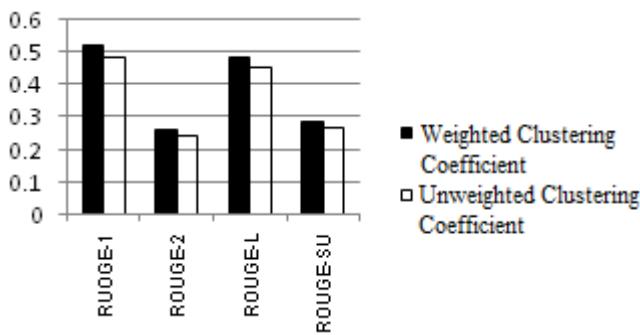


Fig. 3. Experiments using DUC-2002 dataset

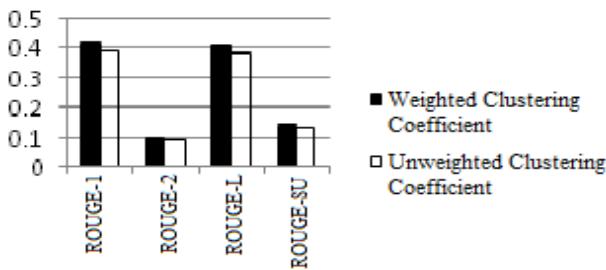


Fig. 4. Experiments using DUC-2004 dataset

## 5 Conclusion and Future Work

In this paper we introduce the use of Wikipedia anchor text and weighted clustering coefficient for multi-document summarization. Additionally, we limit the use of linguistic resources to include only stopwords, stemmers and punctuation marks. The experimental results show that our devised system performs better than unsupervised systems and better/comparable with supervised systems of this area.

As, a future work we are planning to use the relation between Wikipedia anchor texts for improvements in summary quality. We believe that such relation can improve the weighted clustering coefficient score of informative terms and hence, it may improve the summary quality.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: Advances in Neural Information Processing Systems, vol. 14
2. Kumar, N., Srinathan, K.: Automatic keyphrase extraction from scientific documents using N-gram filtration technique. In: Proceeding of the Eighth ACM Symposium on Document Engineering, DocEng 2008, Sao Paulo, Brazil, pp. 199–208 (2008)
3. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford digital library technologies project (1998)
4. Saramaki, J., Onnela, J.-P., Kertesz, J., Kaski, K.: Characterizing Motifs in Weighted Complex Networks
5. McDonald, D.M., Chen, H.: Summary in context: searching versus browsing. ACM Transactions on Information Systems 24(1), 111–141 (2006)
6. Wang, D., Zhu, S., Li, T., Gong, Y.: Multi-Document Summarization using Sentence-based Topic Models. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACL and AFNLP, Suntec, Singapore, pp. 297–300 (August 4, 2009)
7. Ding, C., He, X.: K-means clustering and principal component analysis. In: Prodeedings of ICML 2004 (2004)
8. Ding, C., He, X., Simon, H.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proceedings of Siam Data Mining 2005 (2005)
9. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proceedings of SIGKDD 2006 (2006)
10. Erkan, G., Radev, D.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP 2004 (2004)
11. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of SIGIR (2001)
12. Lee, D.D., Sebastian Seung, H.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, vol. 13
13. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using n-gram cooccurrence statistics. In: Proceedings of NLT-NAACL 2003 (2003)
14. Lin, C.-Y., Hovy, E.: From single to multi-document summarization: A prototype system and its evaluation. In: Proceedings of ACL 2002 (2002)
15. Mani, I.: Automatic summarization. John Benjamins Publishing Company (2001)
16. Radev, D., Jing, H., Stys, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing and Management, 919–938 (2004)
17. Ricardo, B., Berthier, R.: Modern information retrieval. ACM Press (1999)
18. Shen, D., Sun, J.-T., Li, H., Yang, Q., Chen, Z.: Document summarization using conditional random fields. In: Proceedings of IJCAI 2007 (2007)
19. Wang, D., Zhu, S., Li, T., Chi, Y., Gong, Y.: Integrating clustering and multi-document summarization to improve document understanding. In: Proceedings of CIKM 2008 (2008)
20. Yih, W.-T., Goodman, J., Vanderwende, L., Suzuki, H.: Multidocument summarization by maximizing informative content-words. In: Proceedings of IJCAI 2007 (2007)

# Extraction of Relevant Figures and Tables for Multi-document Summarization

Ashish Sadh, Amit Sahu, Devesh Srivastava, Ratna Sanyal, and Sudip Sanyal

Indian Institute of Information Technology, Allahabad, India  
{asheesh.sadh, amit.sahu89, devsri.iiita}@gmail.com,  
{rsanyal, ssanyal}@iiita.ac.in

**Abstract.** We propose a system that extracts the most relevant figures and tables from a set of topically related source documents. These are then integrated into the extractive text summary produced using the same set. The proposed method is domain independent. It predominantly focuses on the generation of a ranked list of relevant candidate units (figures/tables), in order of their computed relevancy. The relevancy measure is based on local and global scores that include direct and indirect references. In order to test the system performance, we have created a test collection of document sets which do not adhere to any specific domain. Evaluation experiments show that the system generated ranked list is in statistically significant correlation with the human evaluators' ranking judgments. Feasibility of the proposed system to summarize a document set which contains figures/tables as their salient units is made clear in our concluding remark.

**Keywords:** Multi-document summarization, Figures, Tables, Ranked list, Local scoring, Global scoring.

## 1 Introduction

Document summarization is a fairly mature research area with wide applicability [1]. The field of summarization encompasses an extreme variety of methodologies, which usually, fall into two categories, namely, extractive and abstractive techniques. Amongst these, extractive techniques have far-reaching potential in emulating domain independent summarization as it refrains from using natural language generation. Thus the system need not be aware of domain specific vocabulary. Generation of a single extractive summary from multiple, topically related documents is a common practice and is frequently applied in various domains. Past several years have resulted in a steady improvement of digital document summarization but failed to achieve the desired effectiveness [2].

The ability to condense information cohesively and coherently is an essential issue of any summarization system and is particularly crucial to the generation of an effective summary. This ability can be further enhanced by incorporating relevant figures/tables at appropriate places in the summarized text. Noticeably, the major

advancements in this field deal with text summarization only [3]. Very few research works address the utility of other document components e.g. tables, pictures, figures, etc. towards the generation of a better summary [4-7].

Figures/Tables are generally introduced into the documents either to elucidate the textual components or to express the information which cannot be well represented in the text form. Human beings tend to understand ideas more easily when expressed in the form of a diagram or a table. Additionally, the figures/tables convey a large chunk of information in relatively condensed form [8]. Hence, these units characterize an excellent choice as components in a summary. Given such significance, one must find a way to extract important figures and tables for effective summarization of digital documents.

As the proliferation of digital content continues, information extraction becomes increasingly complex; in particular, extraction of the most relevant figures/tables [9]. It is further complicated by the fact that the automatic analysis of visual features at very large scale is computationally intensive and more importantly, not much effective [10].

In view of the above stated complexity, we propose a system which extracts important figures and tables from topically related documents by exploiting their association with the textual component. Typically, any figure or table can be associated with its corresponding text using a direct reference or an indirect reference. Direct references (E.g. Fig. 2.1, Table 3.1 etc.) are usually found in scientific documents but not in newspaper articles or magazines. Therefore, in addition to direct references, indirectly referring sentences are also taken into account while computing relevancy score. Our system essentially prepares a ranked list of all the figures/tables, which is ordered, based on their computed relevancy. Given a text-only summary of a document set, the proposed system provides a mechanism to extract the relevant figures/tables from the same set to integrate them into the summary. This integration is assisted by the generated ranked list of the units (figures/tables) and must be done in a way that improves the cohesion and the coherence of the extractive text summary. The system builds on the previous works of text-only multi document summarization [11], further enhancing its capability to summarize documents having figures/tables as their key elements.

The rest of the paper is organized as follows. In Section 2, we discuss related work. We present our proposed method and its implementation in Section 3. In Section 4, we describe evaluation methods. We perform evaluation experiments and discuss the major findings in Section 5. We conclude our paper and discuss future work in Section 6.

## 2 Related Work

Summarization has been a field of vast research [3]. It has recently started exploring importance of non-textual components in regard to document summarization [1]. Robert P. Futrelle, et al [4-5] discussed issues and problems involved in figure summarization. He focused on biological articles and mainly studied content based features of figures. Hong Yu and Minsuk Lee [6] worked on summarization of figures

in documents from the biological domain. In their approach, the abstracts of the biological articles were related with the images present in them. Their hypothesis was that the images can be summarized based on the sentences present in the document. In another approach, Shashank Agarwal and Hong Yu [7] summarized figures by using sentences from each of the four rhetorical categories – Introduction, Methods, Results and Discussion (IMRaD). Ahmet Aker, et al [12], also worked on a domain specific approach for summarizing documents containing information related to geo-referenced images. They used a query based approach for summarization, which performed better than generic ones but lacked information that was selected by human evaluators.

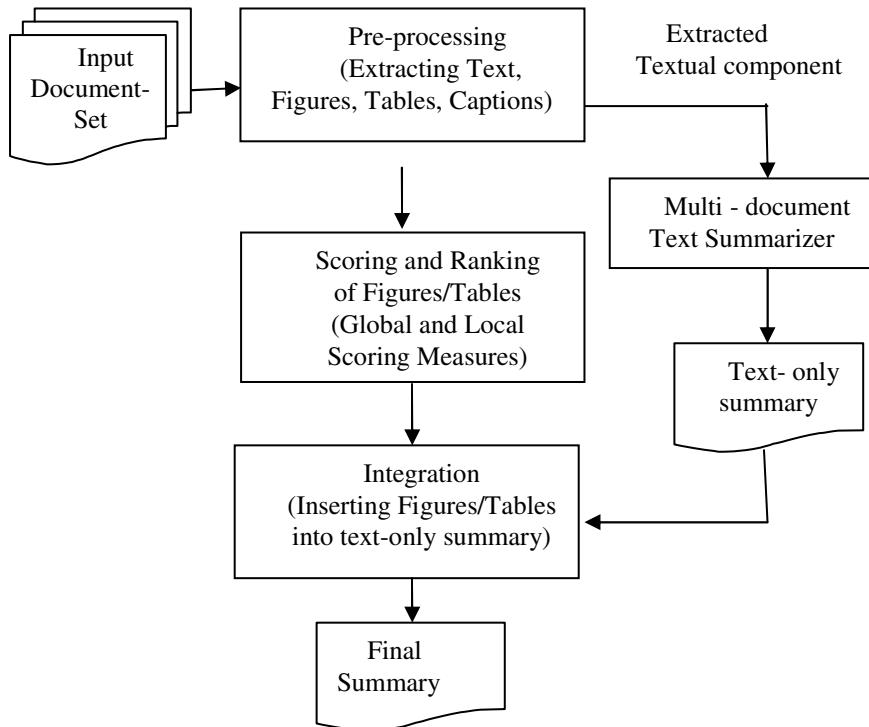
Sumit Bhatia and Prasenjit Mitra [13] referred to the figures and tables present in the documents as document-elements and applied approaches to generate a summary of sentences about these entities (created a synopsis of document elements). Hong Yu, et al [14] developed an approach for figure ranking in full text biomedical articles to help in figure searching. They ranked figures based on their contribution to knowledge discovery. Their hypothesis states that most important figure should be the focus of the article. Hong Yu, et al [15] further explored the applications of figure summarization. Above approaches were mainly directed towards figure summarization. We have applied methods to augment the text summary with figures/tables. Our hypothesis is that importance of figures/tables can be measured in accordance with importance of associated sentences. This hypothesis is inspired from that of Hong Yu and Minsuk Lee [6]. We have also incorporated domain independent methods to extract sentences associated with figures/tables. The related approaches are directed towards extracting information about figures and tables from their document only. We incorporated effects of all the documents in data-set on the importance of figures/tables, from summarization point of view (through global scoring measures).

### 3 Methodology

As the focus of our paper is on the extraction of relevant figures/tables and their integration with the multi document text-summary, our text summarization module is just an implementation of a well-known extractive technique. The technique is outlined in the papers [11,16], which discuss the major principles of a multi-document summarizer named, MEAD. The text summarization module of MEAD consists of three components - a feature extractor, a sentence scorer and a sentence re-ranker. Sentences are added to the summary beginning with the highest scoring sentence. A sentence is added only if its calculated similarity with all the sentences which are already added is above a predetermined threshold.

Given this text only summarizer, we now discuss our method to incorporate figures/tables contained in digital documents into their summary.

In order to gather information about figures/tables of the document set, different components from the documents need to be analyzed. For this purpose, extraction of these components is done as a part of preprocessing, which are further utilized to compute the relevancy score of figures/tables present in that document. A text-only summary is generated using the textual component of the documents. List of figures



**Fig. 1.** Overview of the proposed system

and tables are ranked based on their computed score and finally, integrated to the text summary. Fig. 1 shows a graphical representation of the methodological steps.

### 3.1 Pre-processing

**Document-Components Extraction.** In our implementation of the proposed system, the input documents are of OpenDocument format [17], which is an XML-based open-standard document format. This format allows us to extract figures of a document in a separate directory. The input documents are then converted into html format using an open source utility [18]. This step is done so that text, figures and tables come under different standard tags. Due to this, the extraction of text and search of other components' position within this text, become standard and easier. By looking at the corresponding html tags, text is extracted and the positions of figures and tables are marked in the text. The extracted texts of multiple documents are stored into separate text files, which are used to generate text-only summary using the multi-document text summarizer module.

**Caption Extraction.** The caption of any figure or table carries significant information about it. We now identify the captions for all the figures and tables from the extracted

text in which figure/table positions have already been marked. We look for the presence of words and symbols like fig, figure, diagram, diag, table, ':' inside the preceding and succeeding sentences of the figure/table in the document. Stop-word removal and stemming are then performed on the extracted caption for its future use in subsequent stages.

### 3.2 Figure/Table Scoring and Ranking

To generate a ranked list of figures/tables, we give them a score based on their relationship with the text in the documents and importance of that corresponding text in the summary. The figures/tables to be selected for final summary can be prioritized according to this ranked list. Primarily the stop words are removed from the documents and then the key terms viz. stem of all the remaining words are extracted. Now, the evolved documents are used in the further steps of scoring.

For the scoring of figures/tables, we require two complimentary measures, one based on its local importance within the document and the other based on its global importance within the set of documents.

**Local Measure.** For local scoring of a figure/table, related sentences within the document containing the figure/table are categorized under two labels - direct reference and indirect reference.

*Direct reference.* We analyze the caption to extract the part which is used as a referent to the figure/table. This part is then used to find sentences where the figure/table is cited (using the same referent). These sentences act as direct reference. Their count forms a component (m) of local score.

$$m = \text{number of direct references} . \quad (1)$$

*Indirect reference.* We find the cosine similarity (equation 2) of the caption with each sentence in the document. Those sentences that have high similarity (we have obtained an empirical value of 0.3) act as indirect reference. Indirect references are sentences assumed to be explaining the concept portrayed by the caption of the corresponding figure/table.

Mathematically, cosine similarity could be illustrated as:

$$cs = \cos(\theta) ; \text{ where, } \theta = \cos^{-1} \left( \frac{a \cdot b}{|a||b|} \right) . \quad (2)$$

where, a and b are the d-dimensional vectors representing the two sentences whose cosine similarity is to be calculated; d is the number of terms in vocabulary set of all the source documents.

Let the value of the cosine similarity for  $j^{\text{th}}$  sentence in a total of n indirect references be  $cs(j)$  for the figure F. Sum of cosine similarities (CS) of indirect references with caption form another component for local score, which can be calculated (using equation 2) as:

$$CS = \sum_{j=1}^n cs(j) . \quad (3)$$

**Global Measure.** In our text summarization module, the importance of text in relation to the summary is calculated from the viewpoint of entire document set. In a similar way, while calculating relevancy score of any figure/table to be inserted in this text, we should involve the whole document set, in addition to the local references mentioned earlier. To reflect the same into the relevancy measure, we introduce a global component based on a score (equation 4) calculated for each referring sentence. This score includes the usage frequency of all the terms present in the sentence.

For scoring the sentences, a term frequency matrix is created over the vocabulary of the document set. We believe that the introductory sentences in the documents convey a lot about it, hence the terms appearing in the introductory section are also more important. Frequency of a term present in the introduction is incremented by a weight of 1.2 rather by 1. This matrix is then used to score sentences by adding up the frequency of all the terms appearing in the sentence.

$$scs = \sum_{k=0}^n w_j . \quad (4)$$

where, scs is the score of sentence containing n words with frequency  $w_j$  of jth word. This score is utilized during final calculation of score for indirect (equation 6) and direct (equation 7) references.

### 3.3 Final Score

The final score of a figure/table consists of two scores, one is for direct references and the other is for indirect references denoted by DR and IR respectively. For every figure/table, the sentence scores for directly and indirectly referring sentences are included into the DR and IR as a global measure.

Mathematically, for the figure F, the final score S can be calculated as described in the following. The contribution of the direct references can be calculated as:

$$DR = m * \sum_{i=1}^m scs_i . \quad (5)$$

where m is the number of direct references (equation 1) and  $scs_i$  is the score of  $i^{\text{th}}$  direct reference (equation 4).

The contribution of the indirect references can be calculated as:

$$IR = CS * \sum_{j=1}^n scs_j . \quad (6)$$

where CS is the sum of cosine similarity score of caption with indirect references (equation 3) and  $scs_j$  is the score of  $j^{\text{th}}$  indirect reference (equation 4).

$$\text{Final Score, } S = IR + DR . \quad (7)$$

This process is followed for all the figures and tables in document set.

### 3.4 Ranking

After scoring all the figures and tables on the basis of above method, two ranked lists, one for the figures and the other for the tables, are generated. The ranked lists are in descending order of the scores of units. This ranking reflects the relative importance of a figure/table from the summarization point of view.

### 3.5 Figure/Table Integration

The ranking step is followed by the integration of ranked units into the summary. It is done in two different ways, depending on the type of sentence occurring in the textual summary. All the figures/tables which correspond to any direct reference in the summary-text are unconditionally selected for integration. For the rest of the units, we select a pre-defined percentage (in proportion to text summary produced by text summarization module) of their total count. Figures/Tables referred by indirect references are selected, prioritized by ranked list i.e. higher ranked ones are integrated first till their number exceeds the above percentage.

These units are extracted from different documents and need to be integrated into a single document (i.e. summary). A unique referent is created for each unit. We modified the referent of the direct references in the summary accordingly. Figures/Tables are positioned in the summary after the paragraph that contains its reference (direct or indirect).

## 4 Evaluation

To assess the performance of the proposed system, we devise an experimental evaluation where multiple human evaluators were involved to judge the system ranking of figures/tables. Each human evaluator was asked to generate an expected ranked list which is ordered based on their potential significance to achieve an effective summary after integration.

In essence, our evaluation objective is to evaluate the system generated list based on the gold standard lists proposed by different human evaluators. Kendall's  $\tau$  coefficient and Spearman's rank correlation coefficient are widely used to compare two ordered lists. We calculate the coefficient values for each pair of the system ranking and an evaluator's judgment. However, to effectively reflect agreements and disagreements among multiple gold standards we use the methods named weighted correlation aggregation (WCA), rank-based aggregation (RBA) proposed by the Kim et al. [19]. These two methods address the issue of trustworthiness of different evaluators.

### 4.1 Evaluation Methods

Let  $D = \{d_1, \dots, d_k\}$  be a set of  $k$  figures/tables to be ranked. 'n' number of human evaluators ranked the above  $k$  items in their individual ranked lists, denoted as  $R_1, \dots, R_n$ , where,  $R_i = (d_{i1}, \dots, d_{ik})$  is a ranked-list obtained from  $i^{\text{th}}$  human evaluator. Similarly,  $R = (d_1, \dots, d_k)$  is a system generated ranked-list.

Scoring function  $S(R; R_1, \dots, R_n)$  is used as an evaluation measure for evaluating system ranking  $R$  based on multiple gold standard lists i.e.  $R_1, \dots, R_n$ .

Two scoring functions are used to evaluate our present system as follows:

**Weighted Correlation Aggregation (WCA).** In this approach, the weighted average of correlation values obtained from multiple evaluators is considered as overall score for the system ranking being evaluated. The weight for a particular evaluator is calculated on the basis of agreement with all other evaluator's judgment i.e. average correlation with all other evaluators. Formally,

$$S_{WCA}(R; R_1, \dots, R_n) = \frac{\sum_i^n w_i C(R, R_i)}{\sum_i^n w_i}. \quad (8)$$

$$\text{Where, } w_i = \frac{1}{n-1} \sum_{j=1, j \neq i}^n C(R_i, R_j). \quad (9)$$

Here, the two list correlation measure  $C(R, R_j)$ , can be either Kendall's  $\tau$  coefficient or Spearman's rank correlation coefficient. These two variants are denoted as WCA-  $\tau$  and WCA-Sp respectively.

**Rank-Based Aggregation (RBA).** Ranks assigned by all the evaluators are summarized in the form of consensus order list formed by reordering of elements according to their combined ranking score.

Combined Ranking Score of the  $i^{\text{th}}$  item is given by

$$\text{Rank}_{\text{new}}(d_i) = \sum_{j=1}^n \text{Rank}_j(d_i). \quad (10)$$

where,  $\text{Rank}_j(d_i)$  is the rank of  $d_i$  in  $R_j$  i.e. ranked list of  $j^{\text{th}}$  human evaluator.

The generated Consensus order list can now be evaluated using Kendall's  $\tau$  coefficient or Spearman's rank correlation. Based on which coefficient is being used, we get two variants of the method, denoted by RBA-  $\tau$  and RBA-Sp.

The two scoring methods described in the subsections of 4.1 are basically aggregation of correlation coefficients. We either use Kendall's  $\tau$  coefficient or Spearman's rank correlation, both lie in the range of  $[-1, 1]$ . A correlation value of  $+/-1$  implies perfect correlation, while positive correlation value suggests positive association and negative value indicates negative association. A correlation value nearly zero means no association between the two lists.

## 5 Experimental Results

To support the evaluation experiments, limited scale document collections were prepared from different domains. Five human evaluators were involved in the experiment to generate the gold standards for each document collection.

**Table 1.** Description of the Document Collections

Document Collection	Domain/Topic	No. of Documents	No. of Figures	No. of elements Tables
Doc-Set 1	Scientific (Artificial Neural Network)	5	7	0
Doc-Set 2	Medical (Effect of the Sun on Skin)	3	2	8
Doc-Set 3	Geography (Nile River)	4	12	0

### 5.1 Data-Set

Three document collections which were created to carry out experiments contain on-average 4 articles and are judged by 5 evaluators. Table 1 contains a brief description about these collections. Human evaluators were asked to rank the elements (figures/tables) based on their relative importance for the summarized text. Ranks assigned do not contain tied values i.e. no two units can be ranked at the same level.

The gold standards shown in the tables 2, 3, 4 correspond to Doc-set 1, Doc-set 2 and Doc-set 3 respectively. E1, E2, E3, E4, E5 are the judgments gleaned from the five evaluators.

**Table 2.** Gold standards gathered in response to the Doc-Set 1

Document Number	Figure Number	System Ranks	Evaluators' Rankings				
			E1	E2	E3	E4	E5
5	3	1	6	4	2	1	1
3	1	2	1	3	1	4	3
5	1	3	3	2	4	2	2
2	1	4	2	1	3	6	6
4	1	5	5	6	6	5	4
5	2	6	4	5	5	3	5
1	1	7	7	7	7	7	7
Spearman's rank correlation coefficient			0.39	0.60	0.89	0.67	0.85
Kendall's $\tau$ coefficient			0.33	0.33	0.71	0.52	0.71

**Table 3.** Gold standards gathered in response to the Doc-Set 2

Document Number	Table Number	System Ranks	Evaluators' Rankings				
			E1	E2	E3	E4	E5
1	5	1	1	3	2	1	2
1	1	2	5	1	1	2	1
1	3	3	2	2	3	5	4
1	4	4	6	5	6	4	3
1	7	5	3	4	5	3	6
1	6	6	4	6	4	6	5
1	2	7	7	7	7	7	7
2	1	8	8	8	8	8	8
Spearman's rank correlation coefficient			0.73	0.90	0.88	0.90	0.92
Kendall's $\tau$ coefficient			0.64	0.78	0.71	0.78	0.78

**Table 4.** Gold standards gathered in response to the Doc-Set 3

Document Number	Figure Number	System Ranks	Evaluators' Rankings				
			E1	E2	E3	E4	E5
1	1	1	4	8	7	6	2
3	2	2	2	1	4	4	5
2	1	3	1	3	6	10	7
2	3	4	8	2	3	1	1
4	2	5	7	5	2	3	3
1	4	6	3	7	5	2	8
4	3	7	5	6	1	8	9
4	1	8	6	4	8	7	10
1	2	9	9	11	9	5	8
3	1	10	11	9	10	11	11
1	3	11	10	10	12	9	4
2	2	12	12	12	11	12	12
Spearman's rank correlation coefficient			0.82	0.73	0.66	0.54	0.67
Kendall's $\tau$ coefficient			0.64	0.57	0.45	0.39	0.56

## 5.2 System Performance

Spearman's rank correlation coefficient and Kendall's  $\tau$  coefficient have been calculated [20] for each pair of system ranking and an evaluator's judgment. An aggregated score using these correlation values is calculated for each dataset using methods WCA and RBA as described in the previous section.

The major findings of the experiments are summarized in the table below:

**Table 5.** Scores for the system rankings

Document Collection	WCA- $\tau$ Score	WCA-Sp Score	RBA- $\tau$ Score	RBA-Sp Score
Doc-Set 1	0.767	0.848	0.952	0.982
Doc-Set 2	0.759	0.839	0.878	0.951
Doc-Set 3	0.871	0.935	0.964	0.988

It has been clearly shown in [19] that RBA Scores are more effective and robust than WCA scores. Values of scores obtained for different document sets are considerably close to the perfect correlation value. Our experimental results demonstrate the effectiveness of our proposed system to extract the most relevant figures/tables of the document set from various domains.

## 6 Conclusion

In this paper, we have presented a system that can be used to identify relevant figures/tables from a document set, in order to generate a better summary of it. Evaluation experiments have been performed on document sets from different domains. System performance is reasonably good on all the document sets. This system thus, appears to be especially promising for the summarization of documents having figures/tables as their key elements irrespective of their domain, for example, scientific journals, geographical descriptions etc. However, the resulting set of relevant units still contains some relevant but redundant figures/tables. Furthermore, the figures and the tables can also be pruned to keep only the important portions in them. These are the most challenging issues that need to be resolved and require a new insight and potentially, a new strategy. We plan to address these issues in our future work. It is hoped that the effectiveness of the present system will be improved after resolving these issues.

**Acknowledgements.** The authors gratefully acknowledge the support from the Indian Institute of Information Technology – Allahabad for carrying out this research.

## References

1. Afantinos, S.D., Karkaletsis, V., Stamatopoulos, P.: Summarization from medical documents: a survey. *J. Artificial Intelligence in Medicine* (AIM) 33(2), 157–177 (2005)
2. Lin, C.-Y., Hovy, E.H.: The potential and limitations of automatic sentence extraction for summarization. In: Radev, D., Teufel, S. (eds.) *Proceedings of the HLT-NAACL 2003 on Text Summarization Workshop*, pp. 73–80. ACL, Stroudsburg (2003)

3. Gholamrezaeadeh, S., Salehi, M.A., Gholamzadeh, B.: A Comprehensive Survey on Text Summarization Systems. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) 2nd International Conference on Computer Science and its Applications, CSA 2009, pp. 1–6. IEEE, Jeju (2010)
4. Futrelle, R.P.: Summarization of diagrams in documents. In: Mani, I., Maybury, M. (eds.) Advances in Automated Text Summarization, pp. 403–421. MIT Press, Cambridge (1999)
5. Futrelle, R.P.: Handling figures in document summarization. In: Moens, M.-F., Szpakowicz, S. (eds.) Text Summarization Branches Out. 42nd Annual Meeting of the Association for Computational Linguistics Workshop at ACL, pp. 61–65. ACL, Barcelona (2004)
6. Yu, H., Lee, M.: Accessing bioscience images from abstract sentences. In: Proceedings of 14th International Conference on ISMB, Brazil (2006); *ibid.* *J. Bioinformatics* 22(14), e547–e556 (2006)
7. Agarwal, S., Yu, H.: FigSum: automatically generating structured text summaries for figures in biomedical literature. In: AMIA Annual Symposium Proceedings, pp. 6–10. PubMed Central (2009)
8. Lu, X., Wang, J.Z., Mitra, P., Giles, C.L.: Deriving knowledge from figures for digital libraries. In: Proceedings of the 16th International Conference on World Wide Web, pp. 1229–1230. ACM, Banff (2007)
9. Liu, Y., Mitra, P., Giles, C.L., Bai, K.: Automatic extraction of table metadata from digital documents. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, June 11–15, pp. 339–340. ACM Press, Chapel Hill (2006)
10. Wang, H.H., Mohamad, D., Ismail, N.A.: Image Retrieval: Techniques, Challenge, and Trend. In: The International Conference on Machine Vision, Image Processing, and Pattern Analysis, Bangkok, pp. 25–27 (2009); *ibid.* *J. Waset*, v60–v122 (2011)
11. Radev, D.R., Jing, H., Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility based evaluation, and user studies. In: ANLP/NAACL Workshop on Summarization, vol. 40, pp. 21–29. ACL, Seattle (2000)
12. Aker, A., Gaizauskas, R.: Evaluating automatically generated user-focused multi-document summaries for geo-referenced images. In: Bandyopadhyay, S., Poibeau, T., Saggion, H., Yangarber, R. (eds.) COLING 2008: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 41–48. ACL, Manchester (2008)
13. Bhatia, S., Lahiri, L., Mitra, P.: Generating synopses for document-element search. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 2003–2006. ACM, New York (2009)
14. Yu, H., Liu, F., Ramesh, B.P.: Automatic Figure Ranking and User Interfacing for Intelligent Figure Search. *PLoS ONE* 5(10), e12983 (2010)
15. Agarwal, S., Yu, H.: Figure Summarizer browser extensions for PubMed Central. *J. Bioinformatics* 27(12), 1723–1724 (2011)
16. Radev, D.R., Teufel, S., Saggion, H., Lam, W., Blitzer, J., Qi, H., Celebi, A., Liu, D., Drabek, E.: Evaluation challenges in large-scale multi-document summarization: the mead project. In: Hinrichs, E., Roth, D. (eds.) Proceedings of ACL 2003, pp. 375–382. ACL, Sapporo (2003)
17. The Free and Open Productivity Suite, <http://www.openoffice.org>
18. Odt to html translator, <http://odt2html.gradsoft.ua/Odt2Html.html>
19. Kim, H.D., Zhai, C., Han, J.: Aggregation of Multiple Judgments for Evaluating Ordered Lists. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Ruger, S.M., Rijssbergen, K.V. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 166–178. Springer, Heidelberg (2010)
20. Wessa, P.: Free Statistics Software, Office for Research Development and Education, version 1.1.23-r7 (2011), <http://www.wessa.net>

# Towards Automatic Generation of Catchphrases for Legal Case Reports

Filippo Galgani, Paul Compton, and Achim Hoffmann

School of Computer Science and Engineering  
The University of New South Wales, Sydney, Australia  
`{galganif,compton,achim}@cse.unsw.edu.au`

**Abstract.** This paper presents the challenges and possibilities of a novel summarisation task: automatic generation of catchphrases for legal documents. Catchphrases are meant to present the important legal points of a document with respect of identifying precedents. Automatically generating catchphrases for legal case reports could greatly assist in searching for legal precedents, as many legal texts do not have catchphrases attached. We developed a corpus of legal (human-generated) catchphrases (provided with the submission), which lets us compute statistics useful for automatic catchphrase extraction. We propose a set of methods to generate legal catchphrases and evaluate them on our corpus. The evaluation shows a recall comparable to humans while still showing a competitive level of precision, which is very encouraging. Finally, we introduce a novel evaluation method for catchphrases for legal texts based on the known Rouge measure for evaluating summaries of general texts.

## 1 Introduction

The legal domain has an increasing need for automatic text processing to cope, with the large body of documents that is case law. Due to the importance of precedence in common law, legal research generally is based on searching through case law of applicable jurisdictions looking for facts that are similar to the facts of the current case. Given the large number of court decisions to be scrutinized, information searching can become a very onerous task for legal professionals [12]. Thus natural language processing applications are potentially very useful in the legal domain. While often automatic techniques have been adapted from other domains, there are important differences with the legal domain, which often require techniques specifically developed for this kind of text. For example Brüninghaus and Ashley observed that “*the language used in legal documents is too complex (for NLP techniques to be appropriate). Sentences in the court’s opinion are exceptionally long and often have a very complex structure*” [2].

Among the possible application of language analysis techniques, this paper examines a novel challenge in automatic summarisation: the task of generating and evaluating catchphrases for legal texts. Rather than summaries, case reports often contain a list of catchphrases: phrases that present the important legal points of the case. Catchphrases have an indicative function rather than informative, they present all the legal point considered instead that just summarising the key point(s) of a decision.

Catchphrases give a quick impression on what the case is about: “*the function of catchwords is to give a summary classification of the matters dealt with in a*

case. [...] Their purpose is to tell the researcher whether there is likely to be anything in the case relevant to the research topic” [14]. The presence of catchphrases can improve the performance of retrieval systems and aid research of case law. Examples of legal catchphrases are given in Table 1.

Catchphrases are usually manually drafted by editors or by the authors of the documents, but this varies between courts: while some of them have catchphrases for most cases, others have them only for a portion of cases, and others do not present catchphrases at all. Automatically generating catchphrases is very important both for old documents which do not have any catchphrase as well as to automate the creation of catchphrases for new documents.

**Table 1.** Examples of catchphrases list for three cases

COSTS - proper approach to admiralty and commercial litigation - goods transported under bill of lading incorporating Himalaya clause - shipper and consignee sued ship owner and stevedore for damage to cargo - stevedore successful in obtaining consent orders on motion dismissing proceedings against it based on Himalaya clause - stevedore not furnishing critical evidence or information until after motion filed - whether stevedore should have its costs - importance of parties cooperating to identify the real issues in dispute - duty to resolve uncontentious issues at an early stage of litigation - stevedore awarded 75% of its costs of the proceedings
CORPORATIONS - winding up - court-appointed liquidators - entry into agreement - able to subsist more than three months - no prior approval under s 477(2B) of Corporations Act 2001 (Cth) - application to extend “period” for approval under s 1322(4)(d) - no relevant period - s 1322(4)(d) not applicable - power of Court under s 479(3) to direct liquidator - liquidator directed to act on agreement as though approved - implied incidental powers of Court - prior to approve agreement - power under s 1322(4)(a) to declare entry into agreement and agreement not invalid - COURTS AND JUDGES - Federal Court - implied incidental power - inherent jurisdiction
MIGRATION - partner visa - appellant sought to prove domestic violence by the provision of statutory declarations made under State legislation - “statutory declaration” defined by the Migration Regulations 1994 (Cth) to mean a declaration “under” the Statutory Declarations Act 1959 (Cth) in Div 1.5 - contrary intention in reg 1.21 as to the inclusion of State declarations under s 27 of the Acts Interpretation Act - statutory declaration made under State legislation is not a statutory declaration “under” the Commonwealth Act - appeal dismissed

In this paper we describe our approach towards automatically generating and evaluating catchphrases for legal text. Section 2 overviews related work, Section 3 describes our corpus of legal catchphrases, and Section 4 presents our techniques to extract catchphrases from the text of a case. Section 5 describes how we can automatically evaluate generated catchphrases and Section 6 presents and discusses our first results. Section 7 presents our conclusions and some directions for future research.

## 2 Related Work

Different kinds of language processing have been applied to the legal domain, for example automatic summarisation, retrieval [11], machine translation [6], information extraction [15,1], citation analysis [17,7]. However to our knowledge there has been no previous attempt to automatically generate catchphrases.

Among these tasks, the most relevant to catchphrase extraction is the work on automatic summarisation. Although two different tasks, they both aim at providing a compact representation of a case, presenting only the main points. However, while summaries will focus on the aspects of the case that are considered to be the most important, catchphrases may cover many dimensions of one case: they give a broader representation of a case in that they list all the significant issues considered by the judge, and usually are used as indication of

the relevance of one case in relation to particular issues, rather than a summary as a substitute for the full text.

Automatic summarisation is a well studied application of NLP, but the legal domain has not been investigated as deeply as other popular domains. Comparing extractive summarisation in different domains, Ceylan et al. found (by exhaustive search of possible summaries) that the legal domain is the hardest: i.e. given a golden summary and a baseline, it is more difficult to find sentences that close the gap between the two [3]. Some examples of systems developed for the legal domain are SALOMON [16], a system to summarise Belgian criminal cases, the work of Hachey and Grover [9] to summarise UK House of Lords judgements, and LETSUM [5], a summariser of case reports for the CanLII database (Canadian Legal Information Institute).

## Rouge

Our evaluation method is based on automatic comparisons using Rouge (Recall-Oriented Understudy for Gisting Evaluation) [10]. In Section 5 we describe how we adapt the Rouge score to evaluate sets of catchphrases rather than summaries. Rouge comprises several measures to quantitatively compare system-generated summaries to human-generated summaries, counting the number of overlapping n-grams of various lengths, word pairs and word sequences between two or more summaries. Among the various scores in the Rouge family (for details on how each score is computed see [10]), the results presented here are based on:

- **Rouge-1**, the count of the unigram recall between candidate and reference summaries.
- **Rouge-SU**, based on skip bigrams: a skip bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Rouge-SU counts all in-order matching word pairs, plus common unigrams.
- **Rouge-W**, based on common sequences with maximum length, with a reward for consecutive matches.

## 3 Corpus of Legal Catchphrases

Documents that record decisions are usually stored electronically in different databases. In Australia one example of such database is AustLII<sup>1</sup>, the Australasian Legal Information Institute [8], which is used as the source of data for our system. AustLII is one of the largest sources of legal material on the net, with over four million searchable documents. Analogous databases (LIIIs) exist for many other countries, and the World Legal Information Institute (WordLII<sup>2</sup>) provides a single point for access to all the legal databases found on WordLII's participating LIIIs, with the aim of providing “*free, independent and non-profit access to worldwide law*”.

We accessed case reports from the Federal Court of Australia, for the years 2007 to 2009 and downloaded 5705 documents in html format. Surprisingly we found that for many of these cases no catchphrases are given: only 2816 of the case reports, about half, present a list of catchphrases. We built a parser to analyse these documents and extract the body of the decision and the corresponding

<sup>1</sup> <http://www.austlii.edu.au/>

<sup>2</sup> <http://www.worldlii.org/>

catchphrases for each case. The corpus is submitted as attachment of this paper and we intend to make it available to other researchers.

We extracted some statistics from the texts: the average number of sentences for a document is 221, and every document contains on average 8.3 catchphrases. The average number of words for each document is 7479, while for the catchphrases section the average is 73.6, giving an average compression ratio of 98.4%, this value is higher than compression ratios commonly found both in the legal and other domains (see for example [3]). In total we collected 23230 catchphrases, which form a set of 16566 different catchphrases. 15359 (or 92.7%) of the found catchphrases are unique (appear only in one document in the corpus), while only 1207 appear more than once (7.3%).

Looking at the list of collected catchphrases, we found in fact that some of the catchphrases are very case specific; thus re-use is very limited. Usually there are some high level catchphrases (such as “*Corporations*” or “*Costs*”) and some more generic catchphrases (i.e. “*Federal Court*”) that can be found in a range of cases, while most are longer and quite specific regarding the facts or issues of the case (i.e. “*stevedore not furnishing critical evidence or information until after motion filed*”). Examples from three actual cases were given in Table 1. We also found that for some cases these phrases do not occur in the body of the documents; of the 23230 total catchphrases, 4617 contains words that do not appear at all in the document, and only 14740 (63.5%) the words are in the same sentence of the document (consider an approximate matching which requires at least 70% of their words). This poses some limit to what we can achieve with an extraction based system. We believe that automatic generation of catchphrases could bring as an additional benefit, an increased consistency for catchphrase choice among cases.

## 4 Automatic Extraction of Catchphrases

Our approach towards creating catchphrases is based on extracting sentences from the full text of the case and use them as candidate catchphrases, rather than creating new phrases.

This section presents the different methods which we devised to identify salient sentences in a case. We use the large collection of available catchphrases (from the corpus described in Section 3) to identify relevant words, and experiment with several frequency-based measures to predict which terms indicate important fragments for catchphrase extraction in a given document; then we score sentences based on the presence of these identified terms. Note that we use “term” to refer to single tokens and in the computation of all the methods all terms are stemmed and stopword filtered. We developed seven different scores: **Fcfound**, **Fcfoundfreq**, **Freqmedia**, **TFIDF**, **Thresfreq**, **Thresnocc**, **Myscore**.

The **Fcfound** score of a term  $t$  is the ratio between how many times (that is in how many documents) the term appears both in the catchphrases and in the text of the case, and how many times in the text:

$$Fcfound(t) = \frac{NDocs_{text\&catchp.}(t)}{NDocs_{text}(t)}$$

The Fcfound score of a term is computed using our database of catchphrases from all the corpus (2816 documents).

**Fcfoundfreq** is the previous Fcfound score, multiplied by the number of occurrences of the term in the particular document:

$$Fcfoundfreq(t) = Fcfound(t) \cdot NOccur(t, doc)$$

**Freqmedia** is the ratio between the number of occurrences of the term in the present document and the average number of occurrences of the term in the collection:

$$Freqmedia(t) = \frac{NOccur(t, doc)}{AVG_{alldoc}(NOccur(t))}$$

**TFIDF** is the standard TFIDF measure:

$$TFIDF(t) = Freq(t, doc) \cdot \log \left( \frac{NDocs_{tot}}{NDocs(t)} \right)$$

where  $NDocs_{tot}$  is the total number of documents in the collection, and  $NDocs(t)$  is the number of documents that contains the term  $t$ .

**Thresfreq** and **Thresnocc**: using our data base of catchphrases and documents, for each term, we compute the frequency of the term in documents where the word appears only in the text of the case, and where the words appear both in the text and in the catchphrases. We then select for each term the optimal threshold for the frequency which better separates the two groups. This is done by trying all the observed frequencies of the term in those documents and selecting the one that gives the smallest error on the examples available. Accordingly, for each document we will give a score of one to those terms whose frequency is higher than the corresponding threshold, and zero to the others (Thresfreq considers the frequency, Thresnocc the number of occurrences).

**Myscore** is calculated as follows:

- collect all the sentences that contains any of the 10 most frequent terms in the document. If a sentence contains more than one term, it is collected more than once. Let this set of sentences be  $S$
- take all the terms  $t$  in  $S$ , and count the number of times each word appears in these sentences. We call this number  $NOccur(t, S)$
- the list of candidate terms is formed taking all the terms  $t$  in  $S$ , which appear at least twice in the document
- we give a score to all candidate terms summing their rank on the three following scores: the TFIDF score for the term, the frequency in the document and the ratio  $NOccur(t, S)/NOccur(t, doc)$ .

The rationale behind this method is that some words (the 10 most frequent terms in the document) can indicate which are the “important” sentences of the document, so we look at terms that appear more frequently in those sentences rather than in other parts of the document.

All these methods give scores to terms in a document. Because our goal is to rank sentences for extraction, we wish to assign score to sentences rather than terms, we experimented with both averaging the score for each term of the sentences; or taking the  $n$  top ranked words, looking for sentences that contains them. The first approach gave better results, so all the results presented here are based on it.

## 5 Automatic Evaluation of Catchphrases

An important issue in catchphrase extraction is how to evaluate the generated catchphrases. For human experts, in general, it is not very easy to establish how “good” a candidate catchphrase is, or to rank different candidates, but even more challenging is to run automatic evaluations.

Human expert-based evaluation is considered more accurate and is always important in exploration of possible techniques to use, however two main problems make it too expensive for the exploratory analysis described here: the first is that we would need several experts to obtain a more reliable judgement; the second is that this process become very time consuming for the expert if we need to evaluate a large number of cases. These two factors make human-based evaluation impractical even for a limited number of documents. For this reason we looked for a simple way to evaluate candidate catchphrases automatically by comparing them with the author made catchphrases from our AustLII corpus (considered as our “golden standard”), in order to quickly estimate the performance of a number of methods on a large number of documents. Our goal was not an accurate measure of how well candidate catchphrases matched the target phrases, but to quickly assess methods against each other.

As our system extracts sentences from the full text as candidate catchphrases, so we propose an evaluation method which is based on Rouge scores between extracted sentences and given catchphrases (Rouge was described in Section 2). If we follow the standard Rouge evaluation, we would compare the whole block of catchphrases to the whole block of extracted sentences. However when we are evaluating catchphrases, we do not have a single block of text, but rather several catchphrase candidates. For this reason these should be evaluated individually: the utility of any one catchphrase is minimally affected by the others, or by their particular order. On the other hand we want to extract sentences that contains an entire individual catchphrase, while a sentence that contains only small pieces of different catchphrases is not as useful. An example is given in Figure 1.

<u>Sentences:</u>	<u>Catchphrases</u>
1 The <b>Tribunal</b> should have <b>procedures</b> to guard against such a <b>possibility</b> .	1 <b>denial of procedural fairness</b>
2 The two grounds of this application are that, first, the Tribunal's decision was affected by reasonably apprehended bias and, secondly, <b>denial of procedural fairness</b>	2 decision to issue warrant required forming a view about a <b>possible</b> criminal offence by applicant
	3 <b>Tribunal's</b> decision to be set aside

**Fig. 1.** In this example both sentence 1 and 2 have three words in common with the catchphrases, thus they have the same Rouge score. Using our evaluation methods, however, only sentence 2 is considered a match, as it cover all three terms of catchphrase 1, while sentence 1 has terms from different catchphrases, and thus is not considered a match. This correspond to the intuition that sentence 2 is a better catchphrase candidate.

We therefore devised the following method: we compare each extracted sentence with each catchphrase individually, using Rouge; if the recall (on the catchphrase) is higher than a threshold, the catchphrase-sentence pair is considered a match. For example if we have a 10-word catchphrase, and a 15 words candidate

sentence; if they have 6 words in common, we consider this as a match using Rouge-1 with threshold 0.5, but not a match with a threshold of 0.7 (requiring at least 7/10 words from the catchphrase to appear in the sentence). Using other Rouge scores (Rouge-SU or Rouge-W), the order and sequence of tokens are also considered in defining a match. Once defined the matches between single sentences and catchphrases, for one document and a set of extracted (candidate) sentences, we can compute precision and recall as:

$$Recall = \frac{MatchedCatchphrases}{TotalCatchphrases} \quad Precision = \frac{MatchedSentences}{ExtractedSentences}$$

The recall is the number of catchphrases matched by at least one extracted sentence, divided by the total number of catchphrases, the precision is the number of sentences extracted which match at least one catchphrase, divided by the number of extracted sentences.

This evaluation procedure lets us measure the performance of an extraction system automatically, giving a reasonable measure of how many of the desired catchphrases are generated by the systems, and how many of the sentences extracted are useful. This is different from the use of standard Rouge overall scores, where precision and recall do not relate to the number of catchphrases or sentences, but to the number of smaller units such as n-grams, skip-bigrams or sequences, which makes it more difficult to interpret the results.

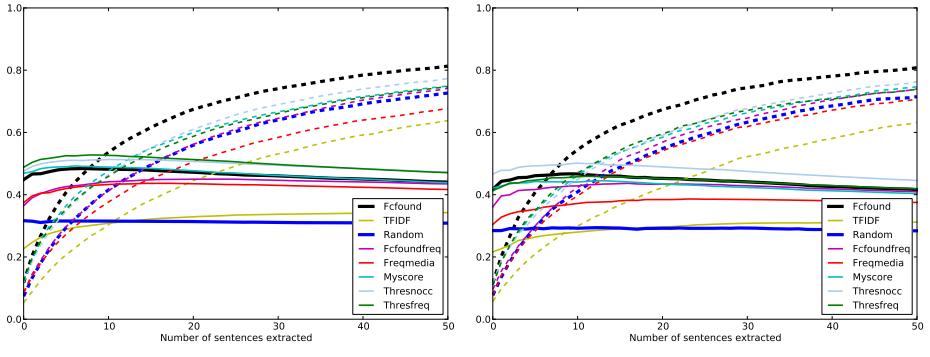
## 6 Experimental Results

We applied the extraction methods described in Section 4 to our corpus of legal cases with catchphrases already given. To obtain an unbiased estimate of the performances, we built a test set, downloading from AustLII new case reports from the Federal Court of Australia decisions in 2010 (of the total 1513 cases, 903 are given with author-created catchphrases, which we used for evaluation). We used the training corpus (2816 case reports) to compute the **Fcfound** scores for each term and to find the thresholds for **Thresfreq** and **Thresnocc**. The other extraction methods do not use any information from catchphrases in the training data, so they can be computed directly on the test set. Then for each document (of the test set, 903 cases), we rank all the sentences according to the methods described. Finally, to compute precision and recall, for each document we calculate which sentences match which catchphrases, using Rouge<sup>3</sup> as described above.

Figure 2 shows precision and recall for the methods, for different number of selected sentences (averaged over all the documents), using Rouge-1 with threshold 0.5 as the matching criterion. For comparison the method **Random** is also included, which is a random selection of sentences from each document (Ceylan et al. [3] already showed how difficult is to improve over random in term of Rouge scores). When comparing the different extraction methods, we can see that **Fcfound** obtains the greater recall and precision, with **Thresfreq** and **Thresnocc** having similar precision but lower recall. An example of a set of sentences extracted by **Fcfound** is given in Figure 3. We can also observe that the performance of the various methods are very similar between the training

<sup>3</sup> We used the Rouge script available from <http://berouge.com>

and the test set, which confirms our intuition that the term-based statistics of catchphrases that we extracted from the corpus comprise most of the relevant word and phrases, and as such can be deemed a general resource and be applied to new data without loss of performances. All the following results are measured on the test set.



(a). Training set: 2816 documents, 2007 to 2009 (b). Test set: 903 documents from 2010

**Fig. 2.** Precision (solid) and Recall (dashed) of extraction methods, for different number of extracted sentences. **Fcfound** and **Random** are represented with thicker lines.

We performed other evaluations, varying the matching criterion: that is using different Rouge scores (Rouge-1, Rouge-SU6 and Rouge-W) with different thresholds (0.5 and 0.7, we tried also other values and the results were comparable) to define a match between a sentence and a catchphrase. More “strict” match conditions give lower values for recall and precision, and vice-versa. However, for any criteria used, the performance of the methods relative to each other and to random are still consistent. Figure 4 plots the difference between **Fcfound** and **Random** for different matching criteria. This shows that the improvement over random of the best performing method is between 10-17% in precision and around 10% in recall.

We also perform what in Section 5 we called the standard Rouge evaluation: comparing the block of sentences extracted to the reference set of catchphrases, as shown in Table 2. Consistently with the data plotted in Figure 2, we can see from the Table that **Fcfound** and the two threshold based methods generally outperform all other methods and **Random**. However we believe that our evaluation relates more directly to the number of sentences (i.e. on average of the first 10 sentences, 5 are “useful”) and catchphrases (with 10 sentences we can cover 60% of the catchphrases), while Rouge tables expresses results in terms of how many “good” words/sequences we obtain.

To better characterize the usefulness of the extraction methods, Figure 5 shows the percentage of documents for which we can generate at least one, half, or all of the catchphrases for each document. We can see that with 10 sentences we can generate at least one of the catchphrases in more than 95% of the documents, and about half of the catchphrases in 60% of the documents.

Sentences:	Catchphrases
1 The extension <b>application</b> is sought under <b>s 1322(4)(d)</b> of the <b>Act</b> . (matches: 9)	1 CORPORATIONS
2 The question then arises whether, on the criteria <b>relevant to the application</b> of <b>s 477(2B)</b> , the <b>Agreement</b> should be <b>approved</b> by the <b>Court</b> . (no matches)	2 winding up
3 The <b>Court</b> may in the exercise of its <b>implied incidental power</b> and its <b>power</b> under s 23 of the <b>Federal Court of Australia Act 1976 (Cth)</b> ( <b>the Federal Court Act</b> ), <b>approve</b> the <b>Agreement</b> . (matches: 12,16,17)	3 court-appointed liquidators
4 The orders sought pursuant to the interlocutory <b>application</b> are as follows: (no matches)	4 entry into agreement
5 If there is no <b>period</b> to <b>extend</b> , then <b>s 1322(4)(d)</b> has no <b>application</b> . (matches: 7, 9)	5 able to subsist more than three months
6 It is hereby <b>declared</b> , pursuant to <b>s 1322(4)(a)</b> of the <b>Corporations Act 2001 (Cth)</b> that the <b>entry</b> by the <b>plaintiffs</b> into the <b>Agreement</b> and the <b>Agreement</b> itself are not <b>invalid</b> by reason of the failure of the <b>plaintiffs</b> to obtain any <b>prior approval</b> required by <b>s 477(2B)</b> of the <b>Act</b> . (matches: 1, 4, 6, 13, 14)	6 no prior approval under s 477(2B) of Corporations Act 2001 (Cth)
7 That pursuant to <b>s 1322(4)(d)</b> of the <b>Law</b> the <b>period</b> for the making of an <b>application</b> under <b>s 477(2B)</b> of the <b>Law</b> for <b>approval of the agreement is extended</b> . (matches: 7, 9)	7 application to extend "period" for approval under s 1322(4)(d)
8 The <b>liquidator</b> may be <b>directed</b> , under <b>s 479(3)</b> of the <b>Act</b> , to <b>act as though</b> the <b>Agreement</b> had been <b>approved</b> . (matches: 11)	8 no relevant period
	9 s 1322(4)(d) not applicable
	10 power of <b>Court</b> under s 479(3) to direct liquidator
	11 liquidator directed to act on agreement as though approved
	12 implied incidental powers of <b>Court</b>
	13 prior to approve agreement
	14 power under s 1322(4)(a) to declare entry into agreement and agreement not invalid
	15 COURTS AND JUDGES
	16 Federal Court
	17 implied incidental power
	18 inherent jurisdiction

**Fig. 3.** The first 8 sentences as extracted by FcFound for a case. Words in bold appear also in the catchphrases. For each sentence the matching catchphrases (if any) is indicated in the brackets. In this case the recall would be  $11/18=0.61$  and the precision  $6/8=0.75$ .

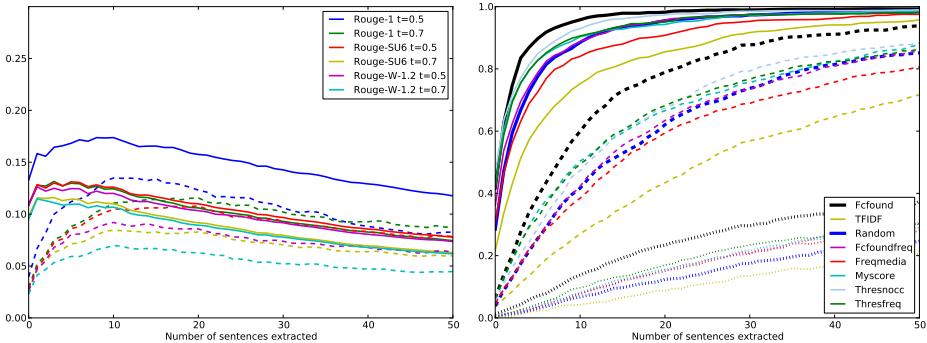
**Table 2.** Rouge evaluation for 10 sentences extracts

	ROUGE-1			ROUGE-SU6			ROUGE-W-1.2		
	Pre	Rec	Fm	Pre	Rec	Fm	Pre	Rec	Fm
Fcfound	0.1495	<b>0.5097</b>	<b>0.2033</b>	0.0537	<b>0.2127</b>	0.0737	0.0997	<b>0.2587</b>	<b>0.1220</b>
Thresfreq	<b>0.1532</b>	0.4766	0.2006	<b>0.0574</b>	0.2084	<b>0.0761</b>	<b>0.1034</b>	0.2441	0.1210
Thresnocc	0.1493	0.4538	0.1933	0.0546	0.1912	0.0712	0.0996	0.2310	0.1152
Myscore	0.1411	0.4781	0.1891	0.0526	0.2067	0.0713	0.0947	0.2435	0.1141
Fcfoundfreq	0.1385	0.4287	0.1802	0.0492	0.1794	0.0648	0.0930	0.2194	0.1083
Freqmedia	0.1130	0.4196	0.1554	0.0389	0.1712	0.0541	0.0767	0.2162	0.0953
Random	0.0969	0.4336	0.1441	0.0268	0.1404	0.0402	0.0635	0.2148	0.0862
TFIDF	0.0949	0.3240	0.1263	0.0285	0.1173	0.0384	0.0648	0.1682	0.0772

We also compared our methods to other human-generated catchphrases. We downloaded catchphrases for all the cases in our corpus from the commercial database LexisNexis CaseBase<sup>4</sup>. These catchphrases are created by professional editors, independently from those created by the original author. The catchphrases given by LexisNexis were compared to the original catchphrases, using our evaluation method and the results are depicted in Figure 6.[3] The human-generated catchphrases show a recall that increases quickly, but only up to a certain value. This confirms our hypothesis that alternative catchphrases can be as good as those provided, or that the same concepts can be expressed with different wording. This also applies to the sentences extracted: if a sentence does not match any catchphrases, that does not mean that it is not useful at all. The evaluation method cannot compare texts with respect to their meaning, but only with respect to the words used.

The precision curve of the human-generated catchphrases shows that the first few catchphrases are very good (catchphrases are taken in the same order as

<sup>4</sup> <http://www.lexisnexis.com/au/>

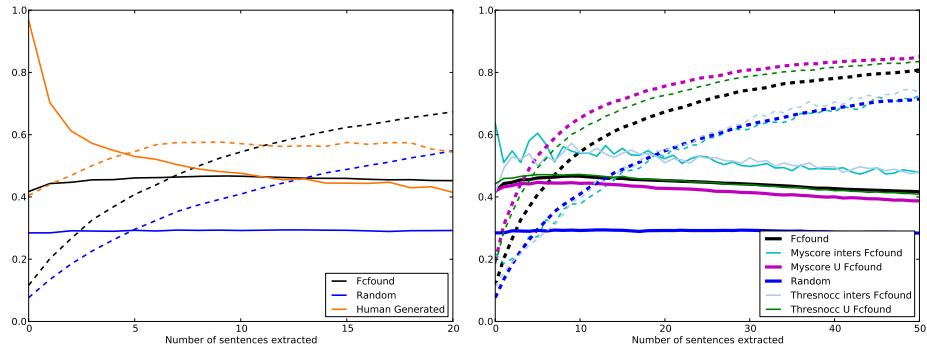


**Fig. 4.** Fcfound plotted as difference from **Fig. 5.** Number of documents with at least Random, for different matching criteria. one (solid line), 50% (dashed line) and Precision is solid, Recall dashed.

100% (dotted line) of the catchphrases matched

given by the expert). Regarding the precision of the other methods, the random line represents the probability that a randomly selected sentence matches at least one catchphrase. For the other extraction methods, the precision appears somewhat “flat”, in that it is not decreasing significantly when we increase the number of sentences. We believe this is due to the fact that there are “easy” catchphrases and more “difficult” ones, and so even low-scored sentences will match the easy catchphrases with some probability.

We expect the performances to improve further by employing more than one method at the same time. We did not examine this issue extensively, but using **Fcfound** as a starting point, we applied it together with the other methods considering their union or intersection. The results are plotted in Figure 7. As expected when we take the intersection of **Fcfound** with another method, the



**Fig. 6.** Comparison of human generated catchphrases, Fcfound and Random. Pre- and Recall dashed.

**Fig. 7.** Union and Intersection of Myscore and Thresnocc with Fcfound. Random and Fcfound (taken alone) are also plotted.

precision increases while the recall decreases, since we are selecting a smaller number of sentences. On the other hand, when we take the union the recall improves around 10%; nevertheless the precision decreases only marginally. We feel that, as these methods looks at different aspects of the text, a better way of combining information should be sought to improve the overall performance of the extraction system. As future research, we are currently investigating a rule based approach which uses the computed scores as attributes for extraction.

## 7 Conclusion and Future Work

In this paper we discuss an application to support legal document search: the automatic extraction of catchphrases from legal cases. We understand from our legal colleagues that catchphrases are considered to be a significant help to lawyers searching through cases to identify relevant precedents and are routinely used when browsing documents. Since legal documents are often missing catchphrases, automatically creating catchphrases for documents is likely to bring substantial benefits.

We collected a corpus of a total of 3719 (training and test sets) case reports with corresponding catchphrases, which is provided with this submission together with the software to reproduce our results. We propose several statistical extraction methods to select sentences from the full text of cases, as candidate catchphrases, using frequency statistics computed from our corpus. We also propose a novel method to evaluate generated catchphrases, based on Rouge, which expresses recall and precision directly in terms of matched catchphrases and useful (i.e. similar to at least one catchphrase) sentences, rather than the less direct units used by Rouge (n-grams or sequences).

When evaluating the extraction methods on an unseen test set, we found that they outperform a random selection of sentences both in precision and recall, in a range of 10-20%, and quickly become comparable to human experts.

As future research we propose to explore different ways of obtaining better catchphrases, by combining information from the several extraction methods. Additionally we propose to explore the use of citation data in relation to this task. As shown for scientific articles (i.e [4,13]) sentences that cite a document can often give a good description of the cited document. We are extending this idea examining, for a legal case, the use of both incoming and outgoing citations, and considering not only citing sentences but also the catchphrases of the cited/citing cases, and relating this information to catchphrase extraction. We are currently creating a corpus with citation information from data available in AustLII.

Finally, evaluation using real users, assessing the utility of the catchphrases produced is left to future research. We believe it is important to have methods that seem likely to be useful via other assessment, before calling on the goodwill of legal practitioners.

## References

1. Ashley, K.D., Brüninghaus, S.: Automatically classifying case texts and predicting outcomes. *Artif. Intell. Law* 17(2), 125–165 (2009)
2. Brüninghaus, S., Ashley, K.D.: Improving the representation of legal case texts with information extraction methods. In: ICAIL 2001: Proceedings of the 8th International Conference on Artificial Intelligence and Law, pp. 42–51. ACM, New York (2001)

3. Ceylan, H., Mihalcea, R., Özertem, U., Lloret, E., Palomar, M.: Quantifying the limits and success of extractive summarization systems across domains. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010, pp. 903–911. Association for Computational Linguistics, Stroudsburg (2010)
4. Elkiss, A., Shen, S., Fader, A., Erkan, G., States, D., Radev, D.: Blind men and elephants: What do citation summaries tell us about a research article? *J. Am. Soc. Inf. Sci. Technol.* 59(1), 51–62 (2008)
5. Farzindar, A., Lapalme, G.: Letsum, an automatic legal text summarizing system. In: The Seventeenth Annual Conference on Legal Knowledge and Information Systems, JURIX 2004, p. 11. IOS Pr. Inc. (2004)
6. Farzindar, A., Lapalme, G.: Machine translation of legal information and its evaluation. In: Advances in Artificial Intelligence, pp. 64–73 (2009)
7. Galgani, F., Hoffmann, A.: Lexa: Towards Automatic Legal Citation Classification. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 445–454. Springer, Heidelberg (2010)
8. Greenleaf, G., Mowbray, A., King, G., Van Dijk, P.: Public Access to Law via Internet: The Australian Legal Information Institute. *Journal of Law and Information Science* 6, 49 (1995)
9. Hachey, B., Grover, C.: Extractive summarisation of legal texts. *Artif. Intell. Law* 14(4), 305–345 (2006)
10. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Moens, M.-F., Szpakowicz, S. (eds.) Text Summarization Branches Out: Proceedings of the ACL 2004 Workshop, pp. 74–81. Association for Computational Linguistics, Barcelona (2004)
11. Moens, M.-F.: Innovative techniques for legal text retrieval. *Artificial Intelligence and Law* 9(1), 29–57 (2001)
12. Moens, M.-F.: Summarizing court decisions. *Inf. Process. Manage.* 43(6), 1748–1764 (2007)
13. Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishnan, P., Qazvinian, V., Radev, D., Zajic, D.: Using citations to generate surveys of scientific paradigms. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, pp. 584–592 (June 2009)
14. Olsson, J.L.T.: Guide To Uniform Production of Judgments, 2nd edn. Australian Institute of Judicial Administration, Carlton South (1999)
15. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: ICAIL 2009: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107. ACM, New York (2009)
16. Uyttendaele, C., Moens, M., Dumortier, J.: Salomon: automatic abstracting of legal cases for effective access to court decisions. *Artificial Intelligence and Law* 6(1), 59–79 (1998)
17. Zhang, P., Koppaka, L.: Semantics-based legal citation network. In: ICAIL 2007: Proceedings of the 11th International Conference on Artificial Intelligence and Law, pp. 123–130. ACM, New York (2007)

# A Dataset for the Evaluation of Lexical Simplification

Jan De Belder and Marie-Francine Moens

Katholieke Universiteit Leuven  
Department of Computer Science  
Celestijnenlaan 200A, B-3001 Heverlee, Belgium  
[{jan.debelder,sien.moens}@cs.kuleuven.be](mailto:{jan.debelder,sien.moens}@cs.kuleuven.be)

**Abstract.** Lexical Simplification is the task of replacing individual words of a text with words that are easier to understand, so that the text as a whole becomes easier to comprehend, e.g. by people with learning disabilities or by children who learn to read.

Although this seems like a straightforward task, evaluating algorithms for this task is not so. The problem is how to build a dataset that provides an exhaustive list of easier to understand words in different contexts, and to obtain an absolute ordering on this list of synonymous expressions.

In this paper we reuse existing resources for a similar problem, that of Lexical Substitution, and transform this dataset into a dataset for Lexical Simplification. This new dataset contains 430 sentences, with in each sentence one word marked. For that word, a list of words that can replace it, sorted by their difficulty, is provided. The paper reports on how this dataset was created based on the annotations of different persons, and their agreement. In addition we provide several metrics for computing the similarity between ranked lexical substitutions, which are used to assess the value of the different annotations, but which can also be used to compare the lexical simplifications suggested by an algorithm with the ground truth model.

## 1 Introduction

The Lexical Simplification (LS) problem can be defined as substituting words with easier alternatives, so that the text becomes easier to comprehend. Important is that the meaning of the original text is not altered, and that it remains fluent.

There are several reasons why we would want to make text easier to understand. Different groups of readers are confronted with the difficulty of texts: adults who suffered a brain injury, deaf persons [13], young readers, non-native speakers [12] and readers with low literacy skills [15,1]. Although these different groups find texts difficult for different reasons, the causes that make texts hard to comprehend overlap to a large degree. What makes a sentence difficult to understand can usually be attributed to one of two factors or both: the lexical difficulty (i.e. difficult words and phrases) and/or the syntactic difficulty

(i.e. complex grammatical constructs). After more than a decade since its first appearance in the literature [5], Lexical Simplification is receiving a renewed interest [18,17,1]. However, the evaluation still remains problematic.

In this paper we discuss how to create ground truth models used in the evaluation of the Lexical Simplification task. Previous research usually only performed a partial evaluation, e.g. by determining whether a replacement is simpler without taking the context into account, thereby bypassing the difficult Word Sense Disambiguation aspect. Furthermore, it is difficult to compare between methods, since it requires human judgments, which are hard to reproduce. Evaluating a different parameter means running a whole set of evaluations again, which is a tedious and expensive process.

We aim to overcome these problems by developing a corpus on which Lexical Simplification methods can be evaluated. We start from an existing corpus for a related task: that of Lexical Substitution. More specifically, we started from the LexSub dataset from the SemEval 2007 Lexical Substitution task [11]. For a given word, annotators provided alternative words that could replace the original word, without changing the meaning of the sentence (too much). This solves the problem of generating words that fit in the context. We extend this dataset by ordering the alternative words by difficulty. With this dataset, we can evaluate different methods.

This paper reports on how this dataset was created based on the annotations of different persons, and their agreement. We also show how we combine the different annotations to a single list of sorted words. In addition we provide several metrics for computing the similarity between ranked lexical substitutions, which are used to assess the value of the different annotations, but which can also be used to compare the lexical simplifications suggested by the machine with a ground truth model.

In the next section, give an overview of the evaluation methodologies in previous research. Section 3 provides the details of the origin of the dataset and the Lexical Substitution task. In section 4, we provide details on our annotation process. In section 5, we analyze the results of the annotation process, and determine the quality. After assessing how reliable it is, we suggest some metrics of evaluating algorithms with it, presented in section 6. We end with our conclusions in section 7.

## 2 Previous Evaluations

In previous work, the focus was mainly on the methods for Lexical Simplification. In this section, we will not discuss the different methods and their results, but instead concentrate on the evaluation methodologies.

[17] extracts simplification from the edit history in simple Wikipedia. The method is a probabilistic model, so that a distinction can be made between edits that remove spam, edits that correct spelling errors, and actual simplifications. The evaluation was done by selecting 200 edits from each of the models (100 random, and the 100 most probable), e.g. “*annually*” → “*every year*”, and letting 3 native speakers rate these. The possible answers were “simpler”, “more

complex”, “equal”, “unrelated”, and “?” (hard to judge). By collapsing these to “simplification”, “not a simplification”, and “?”, an inter annotator agreement of  $\kappa = 0.69$  is obtained. However, these simplification are evaluated out of their original context, whereas previous research has shown that this is important [8].

In [18] the authors focused on context-specific lexical paraphrases. The method uses the Web to find and validate alternative wordings. The evaluation is done on a set of 257 news article headlines, taken from Chinese online newspapers. The measures used are precision and recall, although the authors recognize the difficulty of evaluating the recall. The latter is approximated by grouping all the correct answers from all of the methods they evaluated, and assuming this set is an exhaustive set of all the answers. The precision is based on manual judgments, but the authors do not specify by whom this was judged.

The method in [2] makes a distinction between finding pairs of synonyms, and a context aware approach that decides when to substitute, so it can be used in conjunction with e.g. the method from [17]. The dataset consisted of 65 sentences from Wikipedia, for which their method simplified exactly one word, and the baseline (the method from [5]) was also able to simplify that word (but to a different one). The evaluation was done in a thorough way, rating the degree of simplification (simpler or not), meaning preservation (preserves meaning or not) and grammaticality preservation (bad, ok, good) of the substitutions. The annotations were divided among three native English speakers, and a small portion was annotated multiple times to calculate the pairwise inter annotator agreement, which was moderate for all categories ( $\kappa$  between .35 and .53).

[19] describe a more complex method, that also performs syntactic operations, by treating the problem of text simplification as a monolingual machine translation problem, in which sentences from English Wikipedia are translated to a sentences from the Simple Wikipedia. The evaluation was done by simplifying 100 sentences from the English Wikipedia, held out from the training data, and using machine translation measures (BLEU and NIST scores) to compare the generated sentences with the gold standard, i.e. the aligned sentences from Simple Wikipedia.

[16] also perform text simplification, and use the same dataset as [19] for evaluating their method. Next to the machine translation measures, they also engage humans in the evaluation on 64 of the 100 sentences. 45 unpaid volunteers rated the simplifications in three separate experiments: one that decided whether or not the simplified sentence was simpler than the original, a second experiment to rate the grammaticality of the simplified sentences, and a third to indicate how well meaning was preserved. All ratings were on a five point Likert scale.

### 3 Selecting a Dataset

We create ground truth data by building further on another dataset, constructed for a similar task. With this latter we refer to the SemEval 2007 Lexical Substitution task [10]. This task had a similar objective: given a sentence with one marked word, replace this word with another word, so that it still fits the context. The idea behind this task was Word Sense Disambiguation in a practical

setting. Lexical Substitution (LEXSUB) is a more general problem than the one we are faced with here. For substitution, any replacement that fits the context is a valid solution, whereas for simplification we want valid replacements that are also easier to understand.

The dataset used for the Lexical Substitution task [10] consists of 201 words, which were chosen at random. For each of the words, 10 sentences were retrieved that contained the word or a conjugated form of the word. The sentences were selected from the English Internet Corpus of English produced by Sharoff [14], obtained by sampling data from the Web<sup>1</sup>.

The LEXSUB dataset thus consists of 2010 sentences in total. For each of these sentences, five annotators provided up to three words that could replace the indicated word in each sentence. The annotators also had the possibility of indicating they couldn't think of a better replacement.

To transform this to a Lexical Simplification dataset, we first remove those of the 201 words that are on a list of 'easy words'. We take this list of easy words to be the union of the 'Basic English combined word list' from Simple Wikipedia<sup>2</sup>, and the 3000 words from the Dale-Chall readability measure<sup>3</sup>. It is unlikely that those easy words will have to be simplified, or even can be simplified, so we do not include them in the annotation process. After removal there were 43 words, or 430 sentences, remaining. Later we show that these words are almost always ranked highest in the list, and therefore refrain from annotating them.

This dataset offers a great starting point, as it provides an exhaustive list of alternative words that can replace a given word, based on the context (i.e. the sentence).

## 4 Annotating the Dataset

We started from the same dataset, using the alternative words generated by the annotators as a set of valid alternatives. To convert this dataset from a Lexical Substitution problem to a Lexical Simplification problem, we have to sort the words by their difficulty.

### 4.1 Methodology

We ask the annotators to rank the different alternatives according to how easy to understand they are in the given sentence. We also include the original word in this list of words to be sorted, so we know which alternatives are easier, equally hard, or harder to understand compared to the original. Furthermore, we allow the annotators to rank different words at the same position, for the cases where they think two words are equally difficult.

<sup>1</sup> <http://corpus.leeds.ac.uk/internet.html>

<sup>2</sup> [http://simple.wikipedia.org/wiki/Wikipedia:Basic\\_English\\_combined\\_wordlist](http://simple.wikipedia.org/wiki/Wikipedia:Basic_English_combined_wordlist)

<sup>3</sup> The percentage of words in a text and not on this list is used as an indicator of difficulty.

## 4.2 Annotators

We used two different groups for the annotations. The first is Amazon Mechanical Turk<sup>4</sup>. The advantage is that it is cheap, in comparison to hiring professionals, and the results can be obtained very fast since multiple people can work on it. We requested five annotators for each sentence, located in the U.S. and that completed at least 95% of their previous assignments correctly.

However, there are also people on Mechanical Turk who are keen to finish their assignments as quickly as possible, and this might have a negative effect on the quality. Therefore we also had part of the dataset annotated again, by PhD students<sup>5</sup>. Two more annotations for roughly half the sentences were obtained this way. Using these annotations, we can test the quality of the Mechanical Turk annotations.

In total 46 different Turkers participated, each providing on average 29.5 annotations. The other annotations came from 9 different PhD students, with on average 85.9 annotations.

## 5 Analysis of the Dataset and the Annotations

### 5.1 Measuring Annotator Agreement

To get an idea of the quality of the annotations, we look into methods of calculating the inter annotator agreement. This is not an easy task, since the chance that two rankings are completely identical is very small.

In what follows, let us define  $ann_i$  to be the  $i$ -th annotator,  $n_{ann}$  the number of annotators,  $w_j$  the  $j$ -th word in the list of alternatives, and  $rank_i(w_j)$  the rank given by  $ann_i$  to word  $w_j$ . All the equations below are based on the replacement of a single word in one sentence.

**Fleiss' Kappa.** A typical measure is Cohen's kappa [3], but unfortunately this works only for binary classification problems, with two annotators. To solve the problem of multiple annotators, a solution is to compare each annotator to the majority vote of the other annotators. This is still difficult, since the majority of a ranking is hard to define.

An extension of this measure is Fleiss' kappa [7], that extends to multiple annotators and multiple classes. Although we don't have multiple classes, we can convert our ranking problem into a suitable form. We can do so by taking each two words  $(w_i, w_j)$  in the list, and put them in one of three categories:  $w_i$  and  $w_j$  are ranked equally difficult,  $w_i$  is ranked easier than  $w_j$ , and  $w_i$  is ranked more difficult than  $w_j$ . By doing so, we are able to use the Fleiss' kappa measure. Like for Cohen's kappa, a Fleiss' kappa value of 1 means perfect agreement between the annotators.

<sup>4</sup> <http://www.mturk.com>

<sup>5</sup> Although their native language isn't always English, they have a more than average understanding.

**Rank Correlation.** A more appropriate measure would be the Spearman rank correlation coefficient. This takes into account the natural ranking of the words provided by the annotators, rather than having to convert it to a set of pairwise comparisons. The Spearman rank correlation coefficient is defined as

$$\rho = \frac{\sum_j (rank'_i(w_j) - \overline{rank'_i})(rank'_k(w_j) - \overline{rank'_k})}{\sqrt{\sum_j (rank'_i(w_j) - \overline{rank'_i})^2 \sum_j (rank'_k(w_j) - \overline{rank'_k})^2}} \quad (1)$$

where  $\overline{rank'_i}$  is the average rank of the words given by annotator  $i$ . Often words are ranked at the same position by the annotators, and ties here are solved by assigning them the average of their rank. So a ranking of  $((w_1), (w_2, w_3, w_4), (w_5))$  will assign a rank 1 to  $w_1$ , and rank 3 ( $\frac{2+3+4}{3}$ ) to  $w_2, w_3$  and  $w_4$ . This is indicated by the use of  $rank'$  instead of  $rank$  in equation 1.

To extend this to a one annotator versus majority case, we define the rank assigned by the second annotator to be the average of the ranks given by the other annotators. The correlation coefficient is a number between  $-1$  and  $+1$ , with  $0$  indicating that there is no dependence.

**Penalty Based Agreement.** A third measure we can use to evaluate the agreement, is based on penalties. For each word that is ranked at a different position by two annotators, a penalty is given, proportional to the difference in distance. For each word, we can calculate the following score:

$$\text{score}(w_j) = 1 - \frac{|rank_i(w_j) - rank_k(w_j)|}{\max_l rank_k(w_l)} \quad (2)$$

This is similar to the measure used in [4], for comparing rankings. The items ranked there were names, and they were ordered according to the importance of them in a picture.

However, in [4], this was used to compare a generated ranking with an expert ranking. To extend this to our case, where we compare one annotator  $ann_i$  against the remainder of the annotators, we give a penalty for each annotator:

$$\text{score}(w_j) = 1 - \frac{1}{n_{ann} - 1} \sum_{k=1, k \neq i}^{n_{ann}} \frac{|rank_i(w_j) - rank_k(w_j)|}{\max_l rank_k(w_l)} \quad (3)$$

## 5.2 Outlier Removal

In table 1 we provide the inter annotator agreement measures, discussed above, for the initial annotations retrieved from Mechanical Turk. Although the Fleiss' kappa measure looks low, agreement seems reasonable.

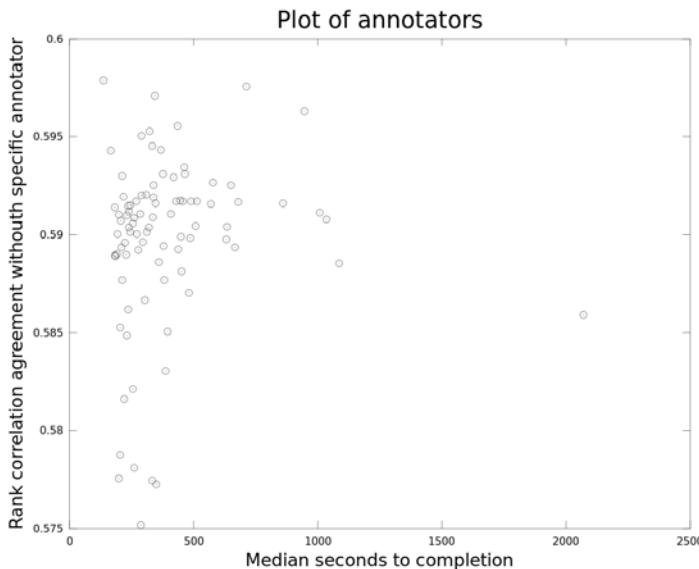
In order to improve the quality of the dataset, we will filter out some of the less accurate annotators. To illustrate we plotted the annotators on a graph, as can be seen in figure 1. On the y-axis there is the inter annotator agreement, as measured with the rank correlation agreement, that would be obtained by

**Table 1.** Agreement of the annotators, initial dataset

Measure	Score
Fleiss' Kappa	0.486
Rank Correlation	0.592
Penalty Based	0.716

**Table 2.** Agreement of the annotators, after filtering the dataset

Measure	Score
Fleiss' Kappa	0.488
Rank Correlation	0.602
Penalty Based	0.724

**Fig. 1.** Graphical representation of the annotators, with on the x-axis the average seconds to completion of 10 sentences, and on the y-axis the change in agreement by removing the annotator

removing a specific annotator. On the x-axis, there is the median submit time for a set of 10 sentences. It is interesting to note that there seems to be little correlation between the average submit time and the quality of the work.

We removed the four annotators that would result in a maximal increase in agreement between the annotators, and had their annotations redone. This resulted in the agreement scores as can be seen in table 2. Although the agreement scores are higher, the change does not seem to be very remarkable.

### 5.3 Evaluation of Quality

With a Fleiss' kappa score of 0.488, we can assume we have only a moderate agreement [9]. This measure takes into account agreement by chance. However,

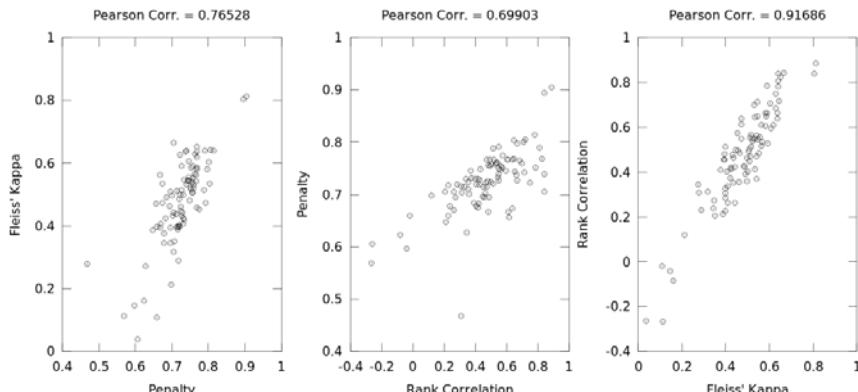
as noted often in the literature [6], it can be misleading. One factor that reduces the agreement, is that the measure as we use it is very strict: if annotator 1 ranks two words as being equally hard, and annotator 2 ranks them directly below each other, this is a disagreement, although in reality the two answers are closely related.

The Spearman rank correlation  $\rho$  of 0.602 indicates there is certainly a correlation between the annotations; if not  $\rho$  would be 0.

As a check for testing the quality of the annotations, we compare the agreement between the annotations between Turkers and the agreement between the annotations done by the students. This is only for a subset of the data (200 of the 430 sentences), and only two annotations were provided for each sentence. The results are in table 3. It can be seen that the annotators are in larger disagreement than the Turkers, illustrating the difficulty of this task, although the smaller number of annotators has to be taken into consideration.

**Table 3.** Agreement of the student annotators

Measure	Score
Fleiss' Kappa	0.393
Rank Correlation	0.451
Penalty Based	0.691



**Fig. 2.** Correlation between the inter annotator agreement metrics

In figure 2 we created a graphical representation of the correlation between the different measures we used for calculating the inter annotator agreement. Each point on the graphs is an annotator, positioned according to the agreement with the rest of the annotators.

## 5.4 Merging Annotations

In this section we convert the multiple rankings from the annotators to a single gold-standard ranking. One way of doing this, is by taking all the pairwise comparisons from all the annotators, and using the most frequent<sup>6</sup> ordering between each two words. For example, if annotator 1 and 2 rank word  $w_1$  higher than  $w_2$ , and annotator 3 ranks them equal, then the most frequent pairwise ranking is  $w_1 > w_2$ . A problem with this approach is that it can cause inconsistencies: the total ordering is not guaranteed anymore. For example, suppose annotator 1 answered  $((w_2), (w_3), (w_1))$ , annotator 2 answered  $((w_1), (w_2, w_3))$ , and annotator 3  $((w_1, w_2, w_3))$ . Then the most frequent orderings are  $(w_1 = w_2)$ ,  $(w_1 = w_3)$ ,  $(w_2 > w_3)$ , which leads to  $w_1 = w_2 > w_3 = w_1$ , or  $w_1 > w_1$ .

The previous way of merging the annotations to a single ordering neglects two factors. First, the distance between two words is not taken into account. In the example above, the first annotator ranked  $w_2$  and  $w_1$  further apart than the other annotators, but this is not reflected. A second factor is that the quality of the annotators is not taken into account: the opinion of each annotator weighs equally, but some are more accurate than others.

With these two considerations in mind, we resort to a noisy channel model method of finding the optimal ordering. With this framework, we can define the source model to be the real ordering of the words, that is at this point unknown. The channel through which we observe this real ordering, is in the form of the annotators, that generate their annotations based on the real ordering, but with additional noise (errors).

We can then calculate the optimal real ordering for the alternative words of a sentence as:

$$\max_h \prod_{i=1}^{n_{ann}} \text{rel}(ann_i) \text{sim}(h, \text{annotation}_i) \quad (4)$$

in which  $\text{rel}$  is the reliability of an annotator, and  $\text{sim}$  is the similarity of the hypothesis real ordering  $h$  and the annotation  $\text{annotation}_i$  provided by the  $i$ -th annotator. We can calculate the similarity by simply reusing equation 1 or 3. For the reliability of an annotator, we also use these equations, in the form of the agreement with the combined annotation of all the other annotators, averaged over all the sentences he/she annotated. In the remainder of this section, we report on the orderings obtained by using the penalty based method from section 5.1.

## 5.5 Properties of the Dataset

After combining the annotations into a single ordering, we can calculate its properties. In 70.5% of the sentences the word can be replaced by one or more simpler words. In 75.6% of the cases, there is also one or more word that is equally hard. Finally, in 71.6% of the cases there are words that are harder.

---

<sup>6</sup> When  $w_1 > w_2$ ,  $w_1 = w_2$ , and  $w_1 < w_2$ , we assume  $w_1 = w_2$  is the most frequent.

The average number of alternative words is 5.04. Since we allowed annotators to rank words on the same level of difficulty, there are on average 3.03 levels.

To illustrate what the dataset looks like, in table 4 there are number of example sentences and alternative words, sorted by difficulty. The two last examples are for the same word, severely, showing the dependency on the context.

Finally, to prove our hypothesis that words that are on the list of easy words are already the easiest word, and can't be simplified further, we also had sentences for five of those words annotated, yielding 50 sentences. For those words, there was only a simpler word in 10% of the cases, illustrating that it is probably better to select new words and sentences altogether.

**Table 4.** Examples sentences with alternative words

- Rabbits often feed on young, *tender* perennial growth as it emerges in spring, or on young transplants. [[soft], [tender, delicate]]
- Performance test for a system coupled with a locally manufactured station engine model MWM will start *shortly*. [[shortly, soon], [before long], [presently]]
- Perhaps the effect of West Nile Virus is sufficient to extinguish endemic birds already *severely* stressed by habitat losses. [[highly], [seriously, severely, extremely], [gravely], [critically]]
- Mutual Funds are so *severely* conflicted that they will not avail themselves of the alleged benefits of the proposed rule. [[badly], [seriously, severely, heavily], [extremely, gravely]]

## 6 Metrics

Now that we have merged the different annotations into a single dataset, we can use it for the evaluation of Lexical Simplification methods. In this section, we will define three metrics to do so, but each time with a different goal.

### 6.1 Binary Metric

When practically using the Lexical Simplification algorithms to simplify text, only a single solution can be used (i.e. a word can only be replaced by one other word). Because we included the original word each time in the list of words to be sorted, we can position the other words relative to the original word. We then define a scoring function as follows:

$$score_{bin}(w_j) = \begin{cases} +1, & \text{if } w_j \text{ is easier.} \\ 0, & \text{if } w_j \text{ is equally hard, harder,} \\ & \text{or not in the list of alternatives.} \end{cases} \quad (5)$$

### 6.2 Rank Evaluation

For a more extensive evaluation, multiple words can be generated in a sorted list. This brings us back to calculating the similarity between two rankings, a

topic that we investigated in detail in section 5.1. Our penalty based method is based on a method for comparing a generated ranking with an expert ranking, so we can use this in its original form:

$$score_{penalty}(w_j) = 1 - \frac{|rank(w) - rank_{gold}(w_j)|}{\max_l rank_{gold}(w_l)} \quad (6)$$

### 6.3 Precision and Recall

Similar to the evaluation in [18], we can calculate the precision and recall, and use these to compute the F-measure. To determine recall, we define the number of easier alternatives as  $n_{easier}$  as  $\#\{w_j | rank_{gold}(w_j) < rank_{gold}(w_{orig})\}$

$$P = \frac{\sum_j score_{bin}(w_j)}{\max_l rank(w_l)} \quad R = \frac{\sum_j score_{bin}(w_j)}{n_{easier}} \quad (7)$$

## 7 Conclusions

The Lexical Simplification of text entails replacing difficult words with words that are easier to understand. But this is a problem that is hard to evaluate, e.g. because the simplifications are context-dependent, and an exhaustive list of simplifications is hard to generate. In this paper, we have shown how we created a dataset<sup>7</sup> for this problem. By reusing an existing dataset for Lexical Substitution, with an exhaustive enumeration of all possible words that can replace a word, we solve the problem of not being able to measure the recall.

Starting from the Lexical Substitution dataset, we have first filtered out words that were too easy to simplify. Next we had annotators sort the different alternative words according to their simplicity, taking into account the context of the original word in the sentence. For these annotations we calculated several inter annotator agreement measures. The main source of the annotations comes from Mechanical Turkers, and we have shown that their agreement is similar to that of less ‘time biased’ annotators. After removing a number of outliers, we merged the annotations into a single gold standard, by interpreting it as a noisy channel problem. Finally, we suggested a number of scoring metrics that can be used with this gold standard.

In the future, we will use this dataset to evaluate Lexical Simplification algorithms. A weakness is that the original dataset replaced words mostly by other single words, i.e. multi word expressions are not very common.

**Acknowledgments.** This research is funded by the EU project *PuppyIR*<sup>8</sup> (EU FP7 231507) and the EU project *TERENCE*<sup>9</sup> (EU FP7 257410).

<sup>7</sup> Available at <http://people.cs.kuleuven.be/~jan.debelder/lseval.zip>.

<sup>8</sup> <http://www.puppyir.eu>

<sup>9</sup> <http://www.terenceproject.eu/>

## References

1. Aluísio, S., Gasperin, C.: Fostering digital inclusion and accessibility: the porsimples project for simplification of Portuguese texts. In: Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pp. 46–53 (2010)
2. Biran, O., Brody, S., Elhadad, N.: Putting it simply: a context-aware approach to lexical simplification. In: Proc. of the 49th Annual Meeting of the ACL: HLT, pp. 496–501. Association for Computational Linguistics (2011)
3. Cohen, J., et al.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
4. Deschacht, K., Moens, M., Robeyns, W.: Cross-media entity recognition in nearly parallel visual and textual documents. In: Large Scale Semantic Access to Content (Text, Image, Video, and Sound), pp. 133–144. Le Centre De Hautes Etudes Internationales D'informatique Documentaire (2007)
5. Devlin, S., Tait, J.: The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, 161–173 (1998)
6. Eugenio, B., Glass, M.: The kappa statistic: A second look. *Computational Linguistics* 30(1), 95–101 (2004)
7. Fleiss, J.: Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5), 378 (1971)
8. Lal, P., Ruger, S.: Extract-based summarization with simplification. In: DUC 2002: Workshop on Text Summarization, Philadelphia, PA, USA, July 11-12 (2002)
9. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. *Biometrics* 33(1), 159 (1977)
10. McCarthy, D., Navigli, R.: Semeval-2007 task 10: English lexical substitution task. In: Proc. of the 4th International Workshop on Semantic Evaluations (SemEval 2007), pp. 48–53 (2007)
11. McCarthy, D., Navigli, R.: The English lexical substitution task. *Language Resources and Evaluation* 43(2), 139–159 (2009)
12. Petersen, S.: Natural language processing tools for reading level assessment and text simplification for bilingual education. Ph.D. thesis, University of Washington (2007)
13. Quigley, S., Paul, P.: Language and deafness. College Hill Books (1984)
14. Sharoff, S.: Open-source corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics* 11(4), 435–462 (2006)
15. Shewan, C., Canter, G.: Effects of vocabulary, syntax, and sentence length on auditory comprehension in aphasic patients. *Cortex: A Journal Devoted to the Study of the Nervous System and Behavior* (1971)
16. Woodsend, K., Lapata, M.: Learning to simplify sentences with quasi-synchronous grammar and integer programming. In: Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 409–420 (2011)
17. Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., Lee, L.: For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In: Human Language Technologies: The 2010 Annual Conference of the NAACL, pp. 365–368 (2010)
18. Zhao, S., Liu, T., Yuan, X., Li, S., Zhang, Y.: Automatic acquisition of context-specific lexical paraphrases. In: Proc. of the IJCAI, pp. 1789–1794 (2007)
19. Zhu, Z., Bernhard, D., Gurevych, I.: A monolingual tree-based translation model for sentence simplification. In: Proc. of the 23rd International Conference on Computational Linguistics (2010)

# Text Content Reliability Estimation in Web Documents: A New Proposal

Luis Sanz, Héctor Allende, and Marcelo Mendoza

Department of Informatics

Universidad Técnica Federico Santa María, Chile

[luis.sanz@postgrado.usm.cl](mailto:luis.sanz@postgrado.usm.cl), [{hallende,mmendoza}@inf.utfsm.cl](mailto:{hallende,mmendoza}@inf.utfsm.cl)

**Abstract.** This paper illustrates how a combination of information retrieval, machine learning, and NLP corpus annotation techniques was applied to a problem of text content reliability estimation in Web documents. Our proposal for text content reliability estimation is based on a model in which reliability is a similarity measure between the content of the documents and a knowledge corpus. The proposal includes a new representation of text which uses entailment-based graphs. Then we use the graph-based representations as training instances for a machine learning algorithm allowing to build a reliability model. Experimental results illustrate the feasibility of our proposal by performing a comparison with a state-of-the-art method.

**Keywords:** Text reliability, content-based trust, textual entailment.

## 1 Introduction

Text content reliability can be defined as *the degree in which the text content is perceived to be true* [18]. Reliability content is a criterion that, following topic relevance, is one of the most influencing aspects that should be considered for assessing the relevance of a Web publication [14]. However, it is very difficult to measure it because is related to a qualitative property of the information.

In this article we introduce an approach for content reliability estimation that can be applied to Web documents. The techniques applied in this article provide jointly an effective method to automatically obtain a text reliability measure that can be used to assess a variety of Web publications. These techniques include a text segmentation strategy based on its syntactic-grammatical structure, a new text representation based on its sentences and the estimation of a distance measure between its content fragments and a knowledge corpus. A key element of our approach is the use of an entailment structure of each document to build a reliability model. We use these entailment structures, that we call entailment-based graphs, to represent how reliable is the content of a document with respect to a knowledge corpus. Then, by considering gold standard reliability scores and by using each graph as a training instance, we build a training dataset which

allow us to learn a reliability model. To illustrate the feasibility of our proposal, we conduct an evaluation of our methods by using assessments provided in the *Automatically Evaluating Summaries of Peers Tasks*, AESOP Task [4] as gold standard scores. Then, we apply learning strategies to build our reliability models. We conduct a comparison against ROUGE-SU4 [9], a state-of-the-art summarization method which exhibits similar properties for the problem of reliability estimation.

The remainder of this article is organized as follows. A discussion about the related work can be found in the next section, where we discuss credibility, reputation and the relation of reliability with summarization. Section 3 includes a formal approach to the problem, a general view of the proposal, and an illustrative example of our strategy. Section 4 presents experimental results obtained from a comparison between our approach and an alternative method. Finally, the implications and findings of this article are discussed in section 5.

## 2 Related Work

### 2.1 Credibility, Reputation, and Summarization

Some approaches that emerged from Information Retrieval (IR) and Natural Language Processing (NLP) have dealt with the problem of assessing text reliability. In the IR field, the most common approach is called credibility. Most of the credibility studies have focused on the analysis of user behavior and the way in which they evaluate the veracity of publications [10,11,16]. Credibility analysis has been slightly focused on analysis of text content. There are also specific attempts in the credibility analysis field, aimed to reliability assessment of blogs content [13,7,17,1]. In summary, credibility is applied indiscriminately to multiple concepts besides to content reliability measure, so the last one can be seen as a subarea of the first one.

Other remarkable attempts in IR field related to reliability measure are focused on Information Quality and Cognitive Authority frame [2,14]. The most common strategies of reliability measures in IR are inclined to the analysis of reputation, from votes of users or authors and, occasionally, to content comparison. A major inconvenient of this approach is that many publications are written by anonymous or unknown authors and, moreover, the contents of the publications for a given author have variable reliability levels. Note that a good reputation does not necessarily imply a high level of reliability.

In the NLP field, the issue of the reliability measure has been handled in very specific cases being most of the efforts related to summarization [8,3,12,9,6]. We need to explain how summarization is related to text reliability focusing on particular on the relation with the AESOP Task. We address this issue in the following section.

## 2.2 AESOP Task, Legibility, Responsiveness, and Pyramid Score

For summarization, the *Automatically Evaluating Summaries of Peers Task* (AESOP Task, [4]) has concentrated attempts from more than 30 universities from different countries to evaluate text content quality related concepts, becoming a major international endeavor dedicated to this topic.

The challenge proposed in the AESOP Task and its benchmark has the following characteristics: Over 44 topics a set of summaries has been made. Every topic was formed by 20 documents, where every text corresponds to an article published by an international News Agency. The articles were extracted from the AQUAINT-2 collection, a LDC English Gigaword subset which collects approximately 2.5 GB of text, with around of 907,000 documents corresponding to the period between October 2004 and March 2006. The articles were written in English and were obtained from a variety of agencies, including France Presse, the Central News Agency of Taiwan, the Xinhua News Agency, and the New York Times, among others.

The documents were selected by experts of the National Institute of Standards and Technology (NIST). The selection was based on the name and a brief description of the topic. For each topic 118 summaries were made, from which 8 were made manually and 110 were made automatically by using different automatic summarization techniques. Then 3 measures related to the content quality of each summary were assessed: *Legibility*, *responsiveness* and *pyramid score*. To assess legibility, expert evaluators assigned to each summary a numeric value from 1 to 5, related to how fluent and readable the summary turns to be, without having into account its content. To measure responsiveness, the evaluators assigned to each summary a numeric value from 1 to 5 based on the perception of how the summary fulfills the topic.

The pyramid score [12] was assessed for each summary based on the level of concordance between the summary content and the descriptive text of each topic. A summary set was defined for each topic, compounded by the set of *Summary Content Units* (SCUs) which describes each topic. These content fragments were manually identified by a group of experts. A weight was assigned to every SCU, depending on the number of model summaries it matches. Thus, the pyramid score for a summary was calculated as the total weight divided by the maximum weight obtained from a summary with an average extent (where the average extent was determined by the SCUs count measure in the model summaries corresponding to the assessed topic).

Our approach for reliability estimation takes advantage of the existence of a benchmark dataset for summarization. We will use these scores for the construction of reliability training datasets, allowing to learn reliability models. We will compare our results with a state-of-the-art summarization method, ROUGE-SU4 [9], to illustrate the feasibility of our approach. As we will explain in Section 3, the reliability of a document regarding a knowledge corpus can be modeled as a summarization process: A document is a good match of a corpus if the corpus is likely to generate the document.

### 3 A New Proposal for Content Reliability Measuring

#### 3.1 Reliability Definition

Now we introduce a formal definition of reliability, which allow us to discuss how we can estimate it. Let  $H$  be a set of possible hypotheses. A hypothesis  $h \in H$  is a statement where a truth value can be assigned. It is assumed that  $h$  is expressed by using a text. A set of truth assignments  $w$  for every possible proposition can be considered;  $w$  represents a mapping from  $H$  to  $\{0 = \text{false}, 1 = \text{true}\}$ , i.e.  $w : H \rightarrow \{0, 1\}$ .

Let  $T$  be a space of possible texts and let  $t \in T$  be a specific text. A set of hypotheses can be extracted from  $t$  and regarding  $w$ , we can build a set of pairs  $H_t = \{(h_{t1}, w_{t1}) \dots (h_{ti}, w_{ti}) \dots (h_{tn}, w_{tn})\}$ , where each  $w_{ti}$  represents the truth value for  $h_{ti}$ . Notice that in our approach we assume the principle of the excluding third or *principium tertium exclusum*.

Now we can assume that a body of knowledge is consolidated in a specific knowledge corpus  $C$ , and their hypotheses can be perceived as reliable. Thus, we can build a set of pairs  $CC = \{(h_{CC1}, w_{CC1}) \dots (h_{CCi}, w_{CCi}) \dots (h_{CCz}, w_{CCz})\}$ , where each  $w_{CCi}$  represents the truth value for a hypothesis  $h_{CCi}$  extracted from the corpus.

The reliability content of a text  $t$  regarding  $CC$  can be represented by the joint distribution  $p(H_t, CC|H_1)$ , where  $H_1$  is the hypothesis which states that  $H_t$  and  $CC$  were generated by the same truth assignment function  $w$ .

The reliability of  $t$  regarding  $C$  can be estimated by the amount of information that  $t$  represents with reference to  $C$ . This amount of information can be measured in terms of coding length  $cl(t)$ , that is the negative logarithm of the probability of  $t$ . Reliability, then, is defined as the gain (in terms of compression) in coding length obtained for codifying  $t$  when  $C$  is known.

$$\text{reliability}(t, c) = \log p(t, c | H_1) - \log p(t). \quad (1)$$

This measure can also be seen as a log-likelihood statistic:

$$\text{reliability}(t, c) = \frac{\log(p(H_t, CC|H_1))}{p(H_t, CC|H_0)}, \quad (2)$$

where  $H_0$  denotes the independence hypothesis.

#### 3.2 A Document-Corpus Reliability Representation

The proposed method for content reliability measuring has the following characteristics:

- This method uses a knowledge corpus as a point of reference, built manually from texts that exhibit a high reliability level.
- The reliability of a text is seen as a measure of similarity between text and a knowledge corpus.

- Our method uses a representation, in which the text to be assessed and the knowledge corpus are decomposed into content fragments, based on its syntactical - grammatical structure. We use a content fragment decomposition strategy based on a discourse commitment extraction algorithm[5].
- Our text representation is based on entailment-based graphs, which represents textual entailment (TE) relationships among the content fragments which compounds each document and the corpus.
- We use the entailment-based graphs and a set of human experts scores for the construction of a training dataset. Then, we apply support vector regression to build reliability models.
- Our reliability model considers the entailment structure of each document instead of a standard text-based representation.

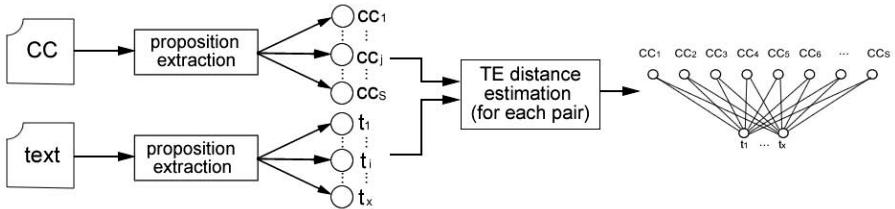
A key component of our approach is the discourse commitment extraction algorithm, which allow us to represent each document by its content fragments elements. This method considers the following steps:

1. Text enrichment: The text is processed by conducting part-of-speech tagging, named entity recognition, pronominal and nominal coreference identification, lexical dependency parsing, and probabilistic context-free grammar extraction.
2. Decomposition: The text is decomposed into content fragments by detecting sentence connectors. These connectors are inferred from the representations obtained in the text enrichment step by applying heuristics.

For further details of the content fragment decomposition strategy please see [5]. Notice that we can measure a textual entailment distance among content fragments. To do this, we propose to use the edit distance function for textual entailment defined by Negri et al. [15] and implemented in EDITS (Edit Distance Textual Entailment Suite). Then, for each document - knowledge corpus pair, we build a bipartite entailment-based graph, with the nodes on one side corresponding to content fragments extracted from a knowledge corpus and on the other side to content fragments extracted from the document to be assessed. The arcs among them represent textual entailment relationships weighted by using the edit distance entailment function. In Figure 1 we illustrate this process.

We use each entailment-based graph as a representation of the reliability that each document exhibits regarding the corpus. According to Equation 1, the  $reliability(t, c)$  function is modeled as the gain in coding length obtained for  $t$  from  $C$ .

Now we propose to use these graphs for reliability estimation. To do this we use these graphs as training instances of a machine learning algorithm in order to build a reliability model. Then, to assess the reliability of an unseen document, we will construct its entailment-based graph and by using the reliability model we will estimate its reliability.



**Fig. 1.** Bipartite entailment-based graph for document reliability assessment

### 3.3 Learning to Estimate Reliability

In this article we explore how we can use our entailment-based graphs as training instances of a reliability model. To follow a supervised approach we need to label each training instance according to its perceived reliability. Many approaches can be explored to conduct this task such as crowdsourcing or expert labeling. In this article we will consider the last course of action.

We assume that for each document - corpus pair, a reliability score is provided by, for example, expert labeling. A key element of our approach is how to use our entailment-based graphs as entries of a support vector regression model. We propose to transform each graph into a vector representation as follows. For each content fragment of the corpus  $CC_j$ , we calculate a document - corpus weight  $\eta_{CC_j}$  given by:

$$\eta_{CC_j} = \sum_{i=1}^x w_{ij}, \quad (3)$$

where each  $w_{ij}$  is the edit distance function for textual entailment between  $CC_j$  and a content fragment  $t_i$  extracted from the document. Then, a vector representation for the document is built over the corpus space, where each dimension of the corpus represents a corpus fragment  $CC_j$ , and the  $j$ -th component of the vector corresponds to  $\eta_{CC_j}$ . Then, each document - corpus pair can be represented by a vector, constructed from its entailment-based graph. Then, each training instance is compounded by the document - corpus vector (the feature vector) and its score (the label). Finally, using this dataset it is possible to conduct a machine learning process for reliability model estimation.

### 3.4 An Illustrative Example

Now we illustrate our proposal. Let  $CC$  be a knowledge corpus compounded by the following texts:

- $text_1$ : *All men are mortals and they fear death, so they study medicine to heal their body and not die*
- $text_2$ : *Medicine has made great strides over the past 100 years. Its advances have allowed the extention of life*

Let  $t$  be a text to be analyzed: *Men no longer fear death and are no longer dedicated to medical school.*

**Text Enrichment.** The text  $txt_1$  is processed according to the following steps:

1) Part-Of-Speech tagging (POS tagging): A POS tagging process is conducted over the text. The tags for  $txt_1$  are shown in Table 1.

**Table 1.** POS  $txt_1$  tags

Id	Word	Lemma	POS	Id	Word	Lemma	POS
1	All	all	DT	12	study	study	VBP
2	men	man	NNS	13	medicine	medicine	NN
3	are	be	VBP	14	to	to	TO
4	mortals	mortal	NNS	15	heal	heal	VB
5	and	and	CC	16	their	they	PRP\$
6	they	they	PRP	17	body	body	NN
7	fear	fear	VBP	18	and	and	CC
8	death	death	NN	19	not	not	RB
9	,	,	,	20	die	die	VB
10	so	so	IN	21	.	.	.
11	they	they	PRP				

2) Named Entity Recognition (NER): A NER process is conducted over each text. In the case of  $txt_1$  none entity was detected.

3) Detection of nominal and pronominal coreferences: Nominal and pronominal terms are identified. In the case of  $txt_1$  we detect the following coreferences:

- coreferent: *mortals*.
- corefered: *they*.

4) Syntactic dependency analysis: A syntactic analysis is conducted over each text. The dependency analysis corresponding to  $txt_1$  is presented in Table 2.

**Table 2.** Syntactic dependency analysis of  $txt_1$

1. det ( men-2 , All-1 )	10. dobj ( study-12 , medicine-13 )
2. nsubj ( mortals-4 , men-2 )	11. aux ( heal-15 , to-14 )
3. cop ( mortals-4 , are-3 )	12. xcomp ( study-12 , heal-15 )
4. nsubj ( fear-7 , they-6 )	13. poss ( body-17 , their-16 )
5. conj_and ( mortals-4 , fear-7 )	14. dobj ( heal-15 , body-17 )
6. dobj ( fear-7 , death-8 )	15. xcomp ( study-12 , not-19 )
7. dep ( mortals-4 , so-10 )	16. conj_and ( heal-15 , not-19 )
8. nsubj ( study-12 , they-11 )	17. dep ( heal-15 , die-20 )
9. ccomp ( mortals-4 , study-12 )	

5) Dependency parsing of probabilistic context-free grammar (Tree PCFG):  
 $(\text{ROOT} (\text{S} (\text{S} (\text{S} (\text{NP} (\text{DT All}) (\text{NNS men})) (\text{VP} (\text{VBP are}) (\text{NP} (\text{NNS mortals})))) (\text{CC and}) (\text{S} (\text{NP} (\text{PRP they})) (\text{VP} (\text{VBP fear}) (\text{NP} (\text{NN death})))))) (\text{, ,}) (\text{IN so}) (\text{S} (\text{NP} (\text{PRP they})) (\text{VP} (\text{VBP study}) (\text{NP} (\text{NN medicine}))) (\text{S} (\text{VP} (\text{TO to}) (\text{VP} (\text{VP} (\text{VB heal}) (\text{NP} (\text{PRP their}) (\text{NN body})))) (\text{CC and}) (\text{RB not}) (\text{VP} (\text{VB die})))))) (\text{. ,}))$ .

The same text enrichment pre-process is applied to  $txt_2$  and  $t$ .

**Decomposition.** To decompose each text we detect sentence connectors. Punctuation, coreferences, POS tags and the PCFG structure are considered to conduct this process according to the heuristics proposed by Hickl & Bensley [5]. These heuristics decompose  $txt_1$ ,  $txt_2$  and  $t$  into the fragments showed in Table 3.

**Table 3.** The content fragment decomposition process. Corpus segments are denoted by  $CC$  and text fragments by  $t_i$ .

<b>id</b>	<b>Proposition</b>
$CC_1$	All men are mortals
$CC_2$	mortals fear death
$CC_3$	mortals study medicine
$CC_4$	mortals study medicine to heal their body
$CC_5$	mortals study medicine to not die
$CC_6$	Medicine has made great strides
$CC_7$	Medicine has made great strides over the past 100 years
$CC_8$	Medicine advances have allowed the extention of life
$t_1$	Men no longer fear death
$t_2$	Men are no longer dedicated to medical school

Then we built a bipartite entailment-based graph with the nodes on one side corresponding to content fragments extracted from the corpus (CCs) and on the other side to content fragments extracted from the document to be assessed ( $t_1$  and  $t_2$ ). The arcs among them are weighthed by using the edit distance entailment function. Table 4 shows the distance values obtained.

**Table 4.** The edit distance values used for weighting the entailment-based graph

Arc	$d_{ij}$										
$A_{11}$	0.089	$A_{14}$	0.162	$A_{17}$	0.232	$A_{22}$	0.203	$A_{25}$	0.137	$A_{28}$	0.160
$A_{12}$	0.532	$A_{15}$	0.165	$A_{18}$	0.171	$A_{23}$	0.147	$A_{26}$	0.218		
$A_{13}$	0.199	$A_{16}$	0.238	$A_{21}$	0.056	$A_{24}$	0.157	$A_{27}$	0.243		

Finally, the vector representation obtained for  $t$  by applying Equation 3 is the following:

$$t \rightarrow CC_1 \ CC_2 \ CC_3 \ CC_4 \ CC_5 \ CC_6 \ CC_7 \ CC_8 \\ 0.145 \ 0.735 \ 0.346 \ 0.319 \ 0.302 \ 0.456 \ 0.475 \ 0.331$$

## 4 Experimental Results

### 4.1 Data Preparation

We built a dataset for reliability model estimation by considering the data instances of the AESOP Task. In particular, we considered each topic as a particular knowledge field, and each summary produced by the group of NIST experts as the corpus for each topic. Then, for each topic (we considered 44 topics) we have 8 expert summaries which are considered as its corpus.

The set of automatically generated summaries was divided into two parts, one to be used as a training set and the other to be used for testing. For each topic, the AESOP Task provides 110 summaries. We used 55 for training, reserving the others 55 for testing. This division was randomly conducted.

We evaluated our approach by the pyramid and the responsiveness scores. We discard the use of the legibility score in this article. We consider two learning strategies, one based on point-wise learning, i.e. each training instance is considered as a vector - label pair, and a list-wise learning, where each training instance is considered a sorted list of vector - label pairs.

For the list-wise learning approach, each list was randomly generated. For each topic, 100 lists were randomly generated, with 10 vector - label pairs in each one. Then, we obtained a total of 8,800 lists (100 for pyramid and 100 for responsiveness for each topic).

For each summary in the testing set we built its entailment-based graph, measuring the outcome of the reliability model obtained by using point-wise learning. For the evaluation of the list-wise approach, we generated 8,800 testing lists, following the same process considered in the training phase. We explored the use of Support Vector Regression (SVR) for model estimation. We used the Kernel Methods Matlab Toolbox implementation of this algorithm.

### 4.2 Results

We used as a performance measure a loss function. This function was calculated as follows. Let  $x$  be a set of testing documents;  $y()$ , the gold standard function, and  $g()$  our reliability function. The loss function that assess  $g()$  is given by:

$$f_{lost} = -\log(P(y|x, g)),$$

where

$$P(y|x, g) = \prod_{i=1}^n \frac{\exp(g(x_{y(i)}))}{\sum_{k=i}^n \exp(g(x_{y(k)}))}$$

and  $y_k(i)$  is the index of the document at position  $i$ . Finally, a global loss value given by the sum of each particular loss value was calculated.

Table 5 shows the results obtained for the proposed approach in this article, which is called RTE Graph, as well as for the state-of-the-art summarization

**Table 5.** Global loss values for RTE Graph and ROUGE-SU4 approaches

Approach	Loss pyramid score	Loss responsiveness score
RTE Graph (list-wise)	9,008	13,223
RTE Graph (point-wise)	9,307	15,334
ROUGE-SU4	12,913	16,572

method ROUGE-SU4. These results were obtained by using SVR, where a tuning process was conducted for parameter optimization.

Table 5 shows that our method achieves a better global performance than ROUGE-SU4, using list-wise and point-wise. In particular, the list-wise approach outperforms the point-wise approach, being this difference more significant when we consider the responsiveness model.

We evaluated also the specific contribution of each element considered in our entailment-graph representation. We started this analysis by evaluating the impact of the use of the edit distance textual entailment function. To do this we calculated the mean of the distance value for each document in the dataset and we performed a comparison against its pyramid score. Table 6 shows these results.

**Table 6.** Comparing pyramid scores and edit distance RTE mean scores

mean edit distance (RTE graph)	
5% highest scores	0,1506
5% lowest scores	0,1892

As Table 6 shows, we find that for the summaries that exhibit the 5% higher pyramid score, its mean distance is smaller than the mean distance of the summaries in the 5% lower pyramid score. This fact indicates to us that the use of the edit distance textual entailment based function allowed us to provide an entailment representation which correlates well with the gold standard.

Regarding the decomposition method considered for content fragment detection, significant improvements were obtained by using coreference identification. This can be observed in Table 7.

**Table 7.** loss with and with out use of coreferences in disaggregation process

	loss pyramid score	loss responsiveness
with use of coreferences	9.008,56	13.223,39
without use of coreferences	9.846,34	14.324,06

Table 7 shows that the use of coreferences is a very effective strategy for the decomposition phase. Its impact in the performance of the method illustrates that in particular the decomposition method is a critical step of our approach.

## 5 Concluding Remarks

In this article we introduced a new document representation which allow us to decide if a corpus is likely to generate a given document. By using textual entailment relationships among content fragments extracted from a corpus and documents to be assessed, we have built entailment-based graphs. We use these graphs and a set of human experts scores to train a machine learning model. By comparing the outcomes of the model over a set of testing documents, we conclude that our approach is feasible, outperforming in information loss a state-of-the-art summarization method.

We introduced a text representation which models if a corpus is likely to generate the text. To obtain this representation we used several NLP techniques and its computational costs were very significant compared to standard term-based representations. This fact limits the use of our approach for on-line document ranking. However, our approach can be considered as a semantic indexing strategy, being these computational costs incorporated to off-line indexing processes.

Our approach for text reliability is close to summarization but they are different concepts. Notice that a good summary is a reliable text, but the opposite is not necessarily true. We take advantage of the first fact (a good summary is a reliable text) to explore the feasibility of our entailment representation.

In this article our main matter of interest was the construction of a reliability estimation model. The merit of this article is to illustrate that this approach is feasible. However there are many open issues for the near future. We are exploring the use of language models for text reliability estimation, trying to address the dependence of our approach to the existence of gold standard scores. Currently we are exploring also how to use our graphs to extract reliability measures, without using supervised learning. Finally, another important issue is the construction of benchmarks for the evaluation of these strategies.

**Acknowledgments.** Mr. Sanz was supported by a Mecesup postgraduate fellowship grant Nr. FSM-0707, Mr. Allende was supported by projects Fondecyt 1110854 and Basal FB0821 CCTVal FB/13HA10, and Mr. Mendoza was supported by projects UTFSM-DGIP 24.11.19 and Fondef DO9I1185.

## References

1. Al-Eidan, R.M.B., Al-Khalifa, H.S., Al-Salman, A.S.: Towards the measurement of arabic weblogs credibility automatically. In: Proceedings of the 11th iiWAS Conference, pp. 618–622. ACM, New York (2009)
2. Cusinato, A., Della Mea, V., Di Salvatore, F., Mizzaro, S.: Quwi: quality control in wikipedia. In: Proceedings of the 3rd Workshop on Information Credibility on the Web, WICOW 2009, pp. 27–34. ACM, New York (2009)
3. Dang, H.T.: Overview of DUC 2005. In: Proceedings of the 2005 Document Understanding Workshop, Vancouver, B.C., Canada (2005)
4. Dang, H.T., Owczarzak, K.: Overview of the tac 2008 update summarization task. In: Proceedings of the First Text Analysis Conference (TAC 2008), Gaithersburg, Maryland, USA, pp. 1–16 (2008)

5. Hickl, A., Bensley, J.: A discourse commitment-based framework for recognizing textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 171–176. ACL, Prague (2007)
6. Hovy, E., Lin, C.Y., Zhou, L.: Evaluating duc 2005 using basic elements. In: Proceedings of DUC 2005, Vancouver, B.C., Canada, pp. 1–6 (2005)
7. Juffinger, A., Granitzer, M., Lex, E.: Blog credibility ranking by exploiting verified content. In: Proceedings of the 3rd Workshop on Information Credibility on the Web, WICOW 2009, pp. 51–58. ACM, New York (2009)
8. Kolluru, B., Gotoh, Y.: On the subjectivity of human authored short summaries. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation, pp. 12–18. ACL, Michigan (2005)
9. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL workshop on Text Summarization Branches Out, Barcelona, Spain, pp. 9–17 (2004)
10. Metzger, M.J.: Making sense of credibility on the web: Models for evaluating online information and recommendations for future research. *Journal of the American Society for Information Science and Technology* 58(13), 2078–2091 (2007)
11. Metzger, M.J., Flanagin, A.J., Medders, R.B.: Social and heuristic approaches to credibility evaluation online. *Journal of Communication* 60, 413–439 (2010)
12. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: The pyramid method. In: Proceedings of the HLT-NAACL Conference, p. 8. Association for Computational Linguistics, New York (2004)
13. Nichols, E., Murakami, K., Inui, K., Matsumoto, Y.: Constructing a scientific blog corpus for information credibility analysis. In: Proceedings of the PACLING Conference, pp. 1–6. ACM, Sapporo (2009)
14. Rieh, S.Y.: Judgment of information quality and cognitive authority in the Web. *J. Am. Soc. Inf. Sci.* 53(2), 145–161 (2002)
15. Negri, M., Kouylekov, M., Magnini, B., Mehdad, Y., Cabrio, E.: Towards Extensible Textual Entailment Engines: The EDITS Package. In: Proceedings of AIIA, Emergent Perspectives in Artificial Intelligence, Reggio Emilia, Italy (2009)
16. Vega, L.C., Sun, Y.T., McCrickard, D.S., Harrison, S.: Time: a method of detecting the dynamic variances of trust. In: Proceedings of the 4th Workshop on Information Credibility, WICOW 2010, pp. 43–50. ACM, New York (2010)
17. Weerkamp, W., Rijke, M.D.: Credibility improves topical blog post retrieval. In: ACL 2008: HLT, pp. 923–931. ACL, Columbus (2008)
18. Xu, Y.C., Chen, Z.: Relevance judgment: What do information users consider beyond topicality? *J. Am. Soc. Inf. Sci. Technol.* 57(7), 961–973 (2006)

# Fine-Grained Certainty Level Annotations Used for Coarser-Grained E-Health Scenarios

## Certainty Classification of Diagnostic Statements in Swedish Clinical Text

Sumithra Velupillai<sup>1</sup> and Maria Kvist<sup>1,2</sup>

<sup>1</sup> Dept. of Computer and Systems Sciences (DSV)  
Stockholm University, Forum 100, SE-164 40 Kista, Sweden

<sup>2</sup> Dept. of Clinical Immunology and Transfusion Medicine  
Karolinska University Hospital, SE-171 76 Stockholm, Sweden  
[sumithra@dsv.su.se](mailto:sumithra@dsv.su.se), [maria.kvist@karolinska.se](mailto:maria.kvist@karolinska.se)

**Abstract.** An important task in information access methods is distinguishing factual information from speculative or negated information. Fine-grained certainty levels of diagnostic statements in Swedish clinical text are annotated in a corpus from a medical university hospital. The annotation model has two polarities (positive and negative) and three certainty levels. However, there are many e-health scenarios where such fine-grained certainty levels are not practical for information extraction. Instead, more coarse-grained groups are needed. We present three scenarios: *adverse event surveillance*, *decision support alerts* and *automatic summaries* and collapse the fine-grained certainty level classifications into coarser-grained groups. We build automatic classifiers for each scenario and analyze the results quantitatively. Annotation discrepancies are analyzed qualitatively through manual corpus analysis. Our main findings are that it is feasible to use a corpus of fine-grained certainty level annotations to build classifiers for coarser-grained real-world scenarios: 0.89, 0.91 and 0.8 F-score (overall average).

**Keywords:** Clinical documentation, Certainty level classification, Annotation granularity, Automatic Summary, Decision Support Alerts, Adverse Event Surveillance, E-health.

## 1 Introduction

A challenging Natural Language Processing (NLP) task is to accurately extract relevant facts from clinical documentation. Speculative and negated information need to be distinguished from asserted information. Electronic health records are rich in factual and speculative opinions about a patient's clinical conditions, often expressed in free-text. This information is valuable for many e-health information access situations.

Certainty level classification in corpora is a growing research area in the domain of computational linguistics and information access, in particular for domain-specific purposes.

## 1.1 Related Work

In the interdisciplinary area of clinical natural language processing, several studies have targeted the issue of accurate information extraction by including negations and speculations in the information extraction model. In [1], assertion classification (present, absent or uncertain) is performed on medical problems. Rule-based and machine-learning techniques are used and compared. The machine-learning method, using features in a window of  $\pm 4$ , outperforms the rule-based method. Contextual features, including negation, are used for classifying clinical conditions in [2]. In this study, uncertainties are, however, not modeled. The BioScope corpus contains annotations for negation and uncertainty [3] on a sentence level, with a subset of clinical radiology reports (the remaining corpus contains biomedical research articles and abstracts). The 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text [4] included a subtask for classifying assertion levels of medical problems. The top performing system on the assertion task obtained an F-score of 0.94 [5]. However, certainty levels are not modeled on a fine-grained level in these studies. In other domains, more fine-grained certainty levels are proposed, e.g. [6], [7] and [8]. The above-mentioned studies are performed on English.

## 1.2 Aim and Objective

In this work, we use a Swedish clinical corpus with diagnostic statements annotated at a fine-grained certainty level [9] to build coarser-grained classifications reflecting three e-health scenarios where this distinction differs for each scenario: *adverse event surveillance*, *decision support alerts* and *automatic summaries*. Creating annotation models is costly. Using fine-grained models for several purposes might be an efficient approach. Our aim is to study whether an existing corpus with fine-grained certainty level annotations can be used for creating multiple scenario-specific certainty level groups, and to study whether limitations in the existing corpus are transferred as limitations in the chosen scenarios. We build automatic classifiers for each scenario, and analyze the results quantitatively. Annotation discrepancies in the corpus are scrutinized and analyzed qualitatively. To our knowledge, no previous research has used fine-grained certainty level annotations for building several use cases with coarse-grained certainty level groups, nor has this been performed on Swedish clinical text.

## 2 Method

A Swedish clinical corpus annotated for fine-grained certainty levels on a diagnostic statement level was used<sup>1</sup>. The fine-grained classification was collapsed into groups for three different coarse-grained e-health scenarios. Automatic classifiers for each scenario were built, using Conditional Random Fields and simple

<sup>1</sup> Approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm) permission number 2009/1742-31/5.

local context features. Results were evaluated quantitatively through precision, recall and F-score. Annotation discrepancies were analyzed qualitatively through manual corpus analysis.

## 2.1 Corpus Characteristics

The corpus consists of assessment entries from a medical emergency ward in the Stockholm area. In these entries, reasoning about the patient's status and diseases is documented. Diagnostic statements were automatically tagged in the clinical notes and the annotators judged their certainty levels [9]. An example entry is shown in Figure 1.

Oklart vad pats symtom kan komma av. Ingen säker <D>infektion</D>. Inga tecken till inflammatorisk sjukdom eller <D>allergi</D>. Reflux med irritation av lufrör och således hosta? Dock har pat ej haft några symtom på <D>refluxesofagit</D>. Ingen ytterligare akut utredning är befogad. Hänvisar till pats husläkare för fortsatt utredning.

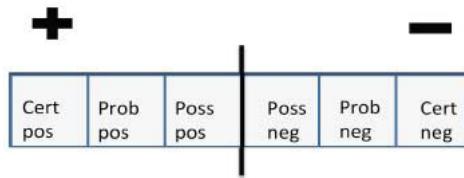
*Unclear what patient's (abbr.) symptoms arise from. No certain <D>infection</D>. No signs of inflammatory disease or <D>allergy</D>. Reflux with irritation of airways and therefore cough? But pat has not had any symptoms of <D>refluxoesophagitis</D>. No further urgent investigation required. Refer to pats GP for continued investigation..*

**Fig. 1.** Example assessment entry. D = Diagnostic statement. Each marked diagnostic statement was judged for certainty levels. In this case, the diagnostic statements *infektion* (infection), *allergi* (allergy) and *refluxesofagit* (refluxoesophagitis) were to be assigned one of the six certainty level annotation classes.

The annotators were shown the entire assessment entry and were asked to annotate each marked diagnostic statement into one of the six certainty level annotation classes<sup>2</sup>. The certainty levels are modeled in two polarities: *positive* and *negative*, as well as certainty level: *certain*, *probable* or *possible*, see Figure 2. Overall Inter- and Intra Annotator (IAA) results, measured on a subset of the total amount of annotations, were 0.7/0.58 and 0.73/0.6 F-measure/Cohens  $\kappa$ , respectively. This subset was used for the qualitative error analysis. The corpus along with guidelines and further analysis are presented in [9]<sup>3</sup>. The full corpus consists of 5 473 assessment entries, 6 186 annotated diagnostic statements and 64 832 tokens (7 464 types) annotated by one annotator. Common error types in the annotations are shown in Table 1. We see similarities in both inter- and intra-annotator discrepancies, the most common error type is *1-step* (66% and 69%).

<sup>2</sup> Other classes were also included, but are not analyzed in this work.

<sup>3</sup> The annotators were two senior physicians, accustomed to reading and writing medical records.



**Fig. 2.** Fine-grained certainty level classification of diagnostic statements into two polarities and three levels of certainty, in total six classes

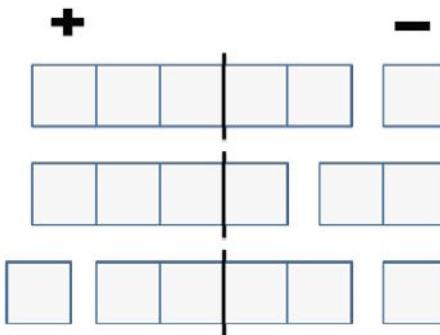
**Table 1.** The most common error types in the annotated corpus. 1-step = discrepancy in one step, e.g. *certainly negative* vs *probably negative*. Certain/Uncertain = discrepancy between the highest level of certainty and intermediate certainty level classes (*probably* or *possibly*). Polarity = discrepancy in *positive* vs *negative*.  $n_{inter}$  = inter-annotator analysis.  $n_{intra}$  = intra-annotator analysis.

Type	$n_{inter}$	%	$n_{intra}$	%
1-step	408	66	284	69
Certain/Uncertain	270	44	191	46
Polarity	99	16	58	14
Total	614	100	411	100

## 2.2 E-Health Scenarios

We define three tentative e-health scenarios: *adverse event surveillance*, *decision support alerts* and *automatic summaries*. These scenarios reflect different needs when it comes to distinguishing and defining the boundaries between certainty levels. The different coarse-grained certainty level groups for the chosen scenarios relate to the original fine-grained classification model as shown in Figure 3. The fine-grained classes *certainly positive*, *probably positive*, *possibly positive*, *possibly negative*, *probably negative* and *certainly negative* are included and excluded in different ways for each scenario. The scenarios are further described below.

**Adverse Event Surveillance.** One instrument used for surveillance of adverse events in hospital care is the Global Trigger Tool [10]. Here, a number of triggers are defined and used for extraction of records which are subsequently manually scrutinized for adverse events. Automation of the trigger identification procedure and extraction of records saves manual labor, and is presently employed at Karolinska University Hospital for triggers in the structured parts of medical records. Further development of this system would be automatic identification of some of these triggers found in the free-text part of health records, and to this add trigger negation detection. Only cases that are negated with the highest possible level of certainty should be excluded in a potential trigger extraction system. Accurate exclusion of negated cases would lower the overall manual work



**Fig. 3.** Modeling e-health use cases by utilizing fine-grained certainty level annotations for coarser-grained classifications, reflecting scenario-specific needs. Top: *adverse event surveillance*. Middle: *decision support alerts*. Bottom: *automatic summaries*.

load. Hence, in this scenario, we get a binary grading: *existence* (at some level of certainty) or *no existence* (at the most certain level). All five annotation classes except *certainly negative* are collapsed into the *existence* grade.

**Decision Support Alerts.** In this scenario, the important distinction in an information access setting, is to flag whenever there is a plausible diagnosis [11]. An example of an automated application would be a decision support: if a plausible case is identified, guidelines or other similar recommendations are automatically shown to the clinician in order to take suitable action. Another potential application would be alerting the clinician who is medically responsible for a patient: a nurse documenting a plausible condition produces an automatic alert to the responsible clinician to take action. Separating positive (or near positive) cases from negative cases is important here. Using the fine-grained certainty level annotation classes, we collapse all positive classes as well as *possibly negative*<sup>4</sup> to one group: *plausible existence*. At the negative polarity *probably negative* and *certainly negative* are collapsed into: *no plausible existence*.

**Automatic Summaries.** When presented with a new patient, an overview, e.g. textual summary, would help the clinician to get an overall impression of earlier diagnoses and health history. A presentation of diagnoses that have been affirmed, excluded, or discussed as a possibility need to be processed by an automatic information extraction system that can distinguish such cases [12]. Moreover, from a different perspective, patients might be interested in obtaining an overview of their own health records in a similar manner, in order to understand and participate in her or his clinical situation. In this scenario, we use *affirmed* and *negated* as two separate groups, and the remaining intermediate, speculative classes are collapsed into one *speculated* group. Hence, we get a multi-class classification problem with three class labels.

<sup>4</sup> The two classes *possibly positive* and *possibly negative* are in this case judged together as a joint middle class.

### 2.3 Automatic Classification and Evaluation

We have used Conditional Random Fields [13], as implemented in CRF++<sup>5</sup> with default parameter settings for building token level classifiers. All sentences containing diagnostic statements annotated for certainty levels were tokenized<sup>6</sup>, and local context features (word, lemma and Part-of-Speech (PoS) tags<sup>7</sup>) with a window of  $\pm 4$  were used for each token, as this setting produces best results [15]. Each diagnostic statement token was assigned exactly one certainty level class, all other tokens were assigned the class *NONE*.

The corpus was divided into a training set (80%, 4 367 sentences, 4 929 diagnostic statements, 51 523 tokens) and a test set (20%, 1 106 sentences, 1 257 diagnostic statements, 13 309 tokens), with a stratified distribution of annotation class labels, see Table 2.

**Table 2.** Coarser-grained certainty level annotation class labels, training and test set: number of class instances and percentages in parentheses. S-1 = *adverse event surveillance*. S-2 = *decision support alerts*. S-3 = *automatic summaries*.

Scenario	Group	Training set			Test set		
		S-1 (%)	S-2 (%)	S-3 (%)	S-1 (%)	S-2 (%)	S-3 (%)
S-1	existence	4 372 (89)			1 103 (88)		
	no existence	557 (11)			154 (12)		
S-2	plausible existence		3 934 (80)			995 (80)	
	no plausible existence		995 (20)			262 (20)	
S-3	affirmed			2 463 (50)			625 (50)
	speculated			1 909 (39)			478 (38)
	negated			557 (11)			154 (12)
Total		4 929 (100)	4 929 (100)	4 929 (100)	1 257 (100)	1 257 (100)	1 257 (100)

Results were measured with precision, recall and F-measure, using the CoNLL 2010 Shared task evaluation script `conlleval.pl`<sup>8</sup>. 95% confidence intervals were calculated for precision and recall. Two baselines were used: majority class baseline and a classifier with no local context features, i.e. the diagnostic statement itself is used as the only feature.

## 3 Results

In this section we present automatic classification results for each e-health scenario, as well as a qualitative error analysis based on the annotated corpus. In the error analysis, we find that difficulties in the distinction between the fine-grained classes *probably negative* and *certainly negative* seem to be the source

<sup>5</sup> <http://crfpp.sourceforge.net/#source>

<sup>6</sup> Multi-word diagnostic statements such as *heart attack* were concatenated and treated as one token.

<sup>7</sup> Using a general Swedish tagger [14].

<sup>8</sup> <http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt>

of most errors in the corpus, and Inter- and Intra-Annotator Agreement (IAA) problems are therefore reflected differently in the three scenarios. We also find that results in the error analysis for the coarse-grained grades are correlated with the distribution of diagnostic statements along the scale of the fine-grained certainty levels. Some diagnostic statements are evenly distributed along this scale, while others are more frequent in the positive polarity (e.g. hypertension, different types of arrhythmias, hyperventilation, allergies, different skin diseases) or negative polarity (e.g. thrombosis and ischemia), as shown in [9]. This reflects the clinical need to negate certain disorders in the documentation, but not others. The discrepancies reflect difficulties in judging certainty for different types of diagnostic statements at the respective polarities, with different types of linguistic and clinical assessment problems arising at the respective polarities accordingly.

### 3.1 Adverse Event Surveillance

In this scenario, we have a binary classification problem: *existence* and *no existence*. This could also be considered similar as a negation detection task.

**Classification Results.** In Table 3, results for the baseline (without context features) and for the classifier using a local context window of  $\pm 4$  is shown. A majority class baseline is 88%. In general, using local context features improves results compared to both baselines (0.89 F-score), but compared to the majority class baseline only a slight improvement is seen. For the minority class *no existence*, context features increase results considerably, in particular for precision (from 0.54 to 0.83), although recall is low (0.51).

**Table 3.** Classification results for the scenario *adverse event surveillance*. Binary classification: *existence* and *no existence*. P = Precision, R = Recall, F = F-score. 95% confidence intervals are given ( $\pm$ ). Majority class baseline = 88%. Baseline = no context features, Local context = word, lemma and PoS-tag, window  $\pm 4$ .

Class label	Baseline			Local context		
	P	R	F	P	R	F
existence	0.53 $\pm$ 0.03	0.98 $\pm$ 0.01	0.68	0.93 $\pm$ 0.01	0.91 $\pm$ 0.02	0.92
no existence	0.54 $\pm$ 0.08	0.14 $\pm$ 0.05	0.23	0.83 $\pm$ 0.06	0.51 $\pm$ 0.08	0.63
Total	0.53 $\pm$ 0.03	0.88 $\pm$ 0.02	0.66	0.92 $\pm$ 0.01	0.86 $\pm$ 0.02	0.89

**Error Analysis.** The lower results for *no existence* in the automatic classification for this scenario appears to be connected to known difficulties in the distinction between *probably negative* and *certainly negative* in the annotated corpus. There are not many errors in assigning polarity (see Table 1), i.e. the diagnostic statements are clearly in the negative polarity, but the strength of the negation has been judged differently in many cases. Part of the errors are due to the lexical context surrounding the diagnostic statement. For instance,

the phrase *inga hållpunkter för* (no indicators of), has been inconsistently interpreted. These cases are also a source of many errors in the automatic classification. Moreover, these inconsistencies are often related to diagnostic statements belonging to diagnosis types that are difficult to exclude, such as *DVT* (deep venous thrombosis), where complete exclusion is clinically difficult. Speculations arise around these diagnosis types because of important severe consequences if missed or misjudged. There are also inconsistencies that depend on whether the annotator(s) have judged the local or global context (i.e. the whole assessment entry, or only the current sentence). Modifiers such as *liten*, e.g. *liten misstanke* (small suspicion), are an interesting source of errors: these can be interpreted differently depending on whether emphasis is put on *misstanke* (suspicion), or *liten* (small), and would need to be defined further in the guidelines.

### 3.2 Decision Support Alerts

In this scenario we need two groups. The classification task is hence modeled with binary class labels: *plausible existence* and *no plausible existence*.

**Classification Results.** In Table 4, results are shown for the classification baseline as well as for using local context features. A majority class assignment is 80%. Overall results are improved using local context features (from 0.61 F-score to 0.91), and are also improved compared to the majority class baseline. For the minority class *no plausible existence*, results are considerably improved both for precision (from 0.72 to 0.92) and recall (from 0.22 to 0.79).

**Table 4.** Classification results for the scenario *alerts for decision support*. Binary classification: *plausible existence* and *no plausible existence*. P = Precision, R = Recall, F = F-score. 95% confidence intervals are given ( $\pm$ ). Majority class baseline = 80%. Baseline = no context features, Local context = word, lemma and PoS-tag, window  $\pm 4$ .

Class label	Baseline			Local context		
	P	R	F	P	R	F
plausible existence	0.48 $\pm$ 0.03	0.97 $\pm$ 0.01	0.64	0.95 $\pm$ 0.01	0.90 $\pm$ 0.02	0.92
no plausible existence	0.72 $\pm$ 0.05	0.22 $\pm$ 0.05	0.34	0.92 $\pm$ 0.03	0.79 $\pm$ 0.05	0.85
Total	0.49 $\pm$ 0.03	0.82 $\pm$ 0.02	0.61	0.94 $\pm$ 0.01	0.88 $\pm$ 0.02	0.91

**Error Analysis.** The boundary in the fine-grained classification model is shifted towards the positive polarity, as compared to the *adverse event surveillance* scenario. The main source of errors lies in cases where certain clinical exclusion is very difficult, due to the nature of the diagnosis itself (e.g. *DVT*). Another source of errors lies in cases where tests have been performed in order to exclude a specific diagnosis. These cases are difficult since performing a test in itself is an indication that there is a risk of this diagnosis, but from the surrounding context it can be evident that the diagnosis is highly unlikely.

### 3.3 Automatic Summaries

In this scenario, we need three grades, resulting in a multi-class classification problem: *affirmed*, *speculated*, and *negated*.

**Classification Results.** A majority class assignment (*affirmed*) is 50%. In Table 5 results for the classifiers (baseline, and context window  $\pm 4$ ) are shown. Using local context features result in a considerable improvement for all classes (0.8 F-score, overall average, compared to 0.5, both baselines). Recall for *negated* is, however, relatively low (0.55).

**Table 5.** Classification results for the scenario *automatic summary*. Multi-class classification: *affirmed*, *speculated* and *negated*. P = Precision, R = Recall, F = F-score. 95% confidence intervals are given ( $\pm$ ). Majority class baseline = 50%. Baseline = no context features, Local context = word, lemma and PoS-tag, window  $\pm 4$ .

Class label	Baseline			Local context		
	P	R	F	P	R	F
affirmed	0.79 $\pm$ 0.03	0.72 $\pm$ 0.03	0.75	0.87 $\pm$ 0.03	0.81 $\pm$ 0.03	0.84
speculated	0.25 $\pm$ 0.02	0.77 $\pm$ 0.02	0.38	0.81 $\pm$ 0.02	0.77 $\pm$ 0.02	0.79
negated	0.50 $\pm$ 0.08	0.18 $\pm$ 0.08	0.27	0.81 $\pm$ 0.06	0.55 $\pm$ 0.08	0.66
Total	0.40 $\pm$ 0.03	0.67 $\pm$ 0.03	0.50	0.84 $\pm$ 0.02	0.76 $\pm$ 0.02	0.80

**Error Analysis.** In this scenario, we focus on an error analysis in the positive polarity, which is not covered in the other two scenarios. These errors mostly reflect difficulties in distinguishing between *probably positive* and *certainly positive* in the annotated corpus. A majority of the cases are due to linguistic markers such as *misstänkt*  $\langle D \rangle x \langle /D \rangle$  (suspected  $\langle D \rangle x \langle /D \rangle$ ) or *kliniska tecken på*  $\langle D \rangle x \langle /D \rangle$  (clinical signs of  $\langle D \rangle x \langle /D \rangle$ ). We see more discrepancies in the annotations concerning diagnosis types determined by subjective judgement, e.g. *hyperventilering* (hyperventilation) and *panikångest* (panic disorder) than diagnosis types that are measured objectively, e.g. *hypertoni* (hypertension). A difference in the judgments made by the human annotators lies in whether they have based their judgments on clinical knowledge or linguistic markers, e.g. *Ur-inprov pos. därfor troligen urinvägsinf.* (Urine sample pos. thus probably urinary tract inf.) We observe some difficult cases for chronic diseases. For instance, the example *troligen stressutlöst astma* (probably stress triggered asthma), could be interpreted as *certainly positive* in the sense that the patient is diagnosed with asthma, or as *probably positive* in the sense that this particular event of an asthma attack is probably triggered by stress.

## 4 Analysis and Discussion

In this study we present work using a corpus annotated with fine-grained certainty classes on a diagnostic statement level, for coarser-grained e-health scenarios. We present three scenarios: *adverse event surveillance*, *decision support*

*alerts* and *automatic summaries*. These scenarios are real-world situations where computerized support is beneficial [12], and where Natural Language Processing techniques involving negation handling may be useful [11]. Each scenario requires different certainty level models, and we collapse classes from the fine-grained classification model into three different coarser-grained groups. We build classifiers using local context features for each scenario. A qualitative analysis on annotation errors deepens the understanding of problems in the boundaries between certainty level classes. We observe promising results by the automatic classifiers for all three scenarios (0.89 F-score (*adverse event surveillance*), 0.91 F-score (*decision support alerts*) and 0.8 F-score (*summaries*), overall average). Our main findings are that it is feasible to use a fine-grained certainty level classification model of diagnostic statements for building coarser-grained e-health scenarios. Although overall IAA is relatively low for the fine-grained model [9], most errors are found in the 1-step borders between the fine-grained levels, thus yielding higher IAA for coarser-grained situations. Annotation discrepancies in intermediate certainty level classes do not pose problems when classes are collapsed into coarser-grained certainty level groups. However, there are some problematic issues, in particular in the distinction between *probably negative* and *certainly negative* in the fine-grained classification model, which need to be further defined in the annotation guidelines. This problem becomes evident when looking at the results for the automatic classifier for the scenario *adverse event surveillance*, where recall in the minority class *no existence* is 0.51. Whether the fine-grained model is considered a sliding scale, or a two-step decision (polarity followed by certainty level) by the annotators is also a factor that should be studied further and need to be clarified when creating fine-grained certainty level annotation tasks.

Previous work (e.g. [1], [2], [4], [5]), on similar tasks are difficult to compare for several reasons. For instance, the certainty level models, annotation tasks, corpora and classification approaches are different to those employed in this work. However, some general trends are observed, such as the problem of skewed class distributions and ambiguity of context cues. Interestingly, local context features in a window of  $\pm 4$  are shown to be useful also for English [1], as well as for Swedish [15]. Cross-lingual studies would be a very interesting continuation of this work. Moreover, the fine-grained certainty levels might also be useful as features for other (higher-level) classification tasks.

Qualitative studies on terminologies used for expressing diagnostic certainties reveal that intermediate probabilities are more often difficult to agree on among human (clinical) evaluators ([16] and [17]), which is in line with our observations. This is an inherently subjective task, and it is not trivial to define what upper performance bounds would be for classifiers.

#### 4.1 Limitations

The automatic classifiers have been built on annotations by one annotator only, not on a consensus set by several annotators. Overall results are also affected by skewed class distributions, results for minority classes need to be further analyzed. Moreover, other classification algorithms should be tested. We treat this

task as a token level classification problem, using Conditional Random Fields for classification. Other classification algorithms or representations might be better suited for this task, this should be studied further and compared. More detailed feature analysis is also needed, as well as under- or oversampling data for dealing with the problem of skewed class distributions. For instance, no global context features have been used, nor any clinical domain-knowledge based features, such as test results.

Moreover, the qualitative error analysis is performed on annotations by two annotators, and only on a subset of the original corpus. A correlation between inter-annotator discrepancies and the errors resulting from the classifiers should be analyzed in future studies.

## 4.2 Significance of Study

Our results are valuable for further work on creating accurate information extraction methods for clinical real-world cases. In health care, there is a constant need for quick decisions based on earlier documentation. This is often complicated by the accumulating mass of text surrounding every patient case. Automatic text processing for applications such as decision support and summaries or overviews, adapted to natural language, would facilitate the clinical workday. Also, automation of surveillance tools for adverse events can assist in improvement of hospital care. This study indicates that it is possible to use a general resource for specific scenario solutions. Instead of creating, in this case, three coarse-grained annotation tasks and subsequent corpora, one fine-grained model can be used for several purposes successfully. To our knowledge, no previous research has used fine-grained certainty level annotations for building several coarse-grained use cases, nor has this been studied on Swedish clinical text.

**Acknowledgments.** We would like to express our appreciations to the anonymous and known reviewers for invaluable comments and suggestions for this paper.

## References

1. Uzuner, Ö., Zhang, X., Sibanda, T.: Machine Learning and Rule-based Approaches to Assertion Classification. *JAMIA* 16, 109–115 (2009)
2. Harkema, H., Dowling, J.N., Thornblade, T., Chapman, W.W.: ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics* 42, 839–851 (2009)
3. Vincze, V., Szarvas, G., Farkas, R., Móra, G., Csirik, J.: The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics* 9 (2008)
4. Uzuner, Ö., South, B.R., Shen, S., DuVall, S.L.: 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *JAMIA* 18, 552–556 (2011)
5. de Brujin, B., Cherry, C., Kiritchenko, S., Martin, J., Zhu, X.: Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *JAMIA* 18, 557–562 (2011)

6. Wilbur, J.W., Rzhetsky, A., Shatkay, H.: New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC Bioinformatics* 7, 356+ (2006)
7. Rubin, V.L., Liddy, E.D., Kando, N.: Certainty identification in texts: Categorization model and manual tagging results. In: *Computing Affect and Attitude in Text: Theory and Applications*. Springer, Heidelberg (2006)
8. Saurí, R.: A Factuality Profiler for Eventualities in Text. PhD thesis, Brandeis University (2008)
9. Velupillai, S., Dalianis, H., Kvist, M.: Factuality Levels of Diagnoses in Swedish Clinical Text. In: Moen, A., Andersen, S.K., Aarts, J., Hurlen, P. (eds.) *Proc. XXIII Intl. Conf. of the European Federation for Medical Informatics*, pp. 559–563. IOS Press, Oslo (2011)
10. Griffin, F.A., Resar, R.: IHI Global Trigger Tool for Measuring Adverse Events, 2nd edn. IHI Innovation Series white paper. Institute for Healthcare Improvement, Cambridge (2009)
11. Denny, J.C., Miller, R.A., Waitman, L.R., Arrieta, M.A., Peterson, J.F.: Identifying QT prolongation from ECG impressions using a general-purpose Natural Language Processor. *IJMI* 78(suppl.1), 34–42 (2009)
12. Kvist, M., Skeppstedt, M., Velupillai, S., Dalianis, H.: Modeling human comprehension of Swedish medical records for intelligent access and summarization systems, a physician's perspective. In: *Proc. 9th Scandinavian Conf. on Health Informatics*, SHI, Oslo (2011)
13. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *ICML*, pp. 282–289 (2001)
14. Knutsson, O., Bigert, J., Kann, V.: A robust shallow parser for Swedish. In: *Proceedings of Nodalida 2003*, Reykavik, Iceland (2003)
15. Velupillai, S.: Automatic Classification of Factuality Levels – A Case Study on Swedish Diagnoses and the Impact of Local Context. In: *Proc. 4th Intl. Symp. on Languages in Biology and Medicine (LBM 2011)*, Singapore (2011)
16. Khorasani, R., Bates, D.W., Teeger, S., Rothschild, J.M., Adams, D.F., Seltzer, S.E.: Is terminology used effectively to convey diagnostic certainty in radiology reports? *Academic Radiology* 10, 685–688 (2003)
17. Hobby, J.L., Tom, B.D.M., Todd, C., Bearcroft, P.W.P., Dixon, A.K.: Communication of doubt and certainty in radiological reports. *The British Journal of Radiology* 73, 999–1001 (2000)

# Combining Confidence Score and Mal-rule Filters for Automatic Creation of Bangla Error Corpus: Grammar Checker Perspective

Bibekananda Kundu<sup>1,2</sup>, Sutanu Chakraborti<sup>2</sup>, and Sanjay Kumar Choudhury<sup>1</sup>

<sup>1</sup> Language Technology, Centre for Development of Advance Computing,  
Kolkata-700091, India

<sup>2</sup> Department of Computer Science and Engineering, Indian Institution of Technology,  
Chennai-600036, India  
`{bibekananda.kundu, sanjay.choudhury}@cdac.in,`  
`sutanuc@cse.iitm.ac.in`

**Abstract.** This paper describes a novel approach for automatic creation of Bangla error corpus for training and evaluation of grammar checker systems. The procedure begins with automatic creation of large number of erroneous sentences from a set of grammatically correct sentences. A statistical Confidence Score Filter has been implemented to select proper samples from the generated erroneous sentences such that sentences with less probable word sequences get lower confidence score and vice versa. Rule based Mal-rule filter with HMM based semi-supervised POS tagger has been used to collect the sentences having improper tag sequences. Combination of these two filters ensures the robustness of the proposed approach such that no valid construction is getting selected within the synthetically generated error corpus. Though the present work focuses on the most frequent grammatical errors in Bangla written text, detail taxonomy of grammatical errors in Bangla is also presented here, with an aim to increase the coverage of the error corpus in future. The proposed approach is language independent and could be easily applied for creating similar corpora in other languages.

**Keywords:** Automatic Error Corpora Creation, Confidence Score, Mal-rule, Grammar Checking.

## 1 Introduction

Socrates's famous dictum was “Correct language is the prerequisite for correct living”. In the context of our everyday use of editing environments, the need of automatic grammatical error detection and correction cannot be overemphasized. The system plays a pivotal role in Computer-Assisted Language Learning (CALL) for second language learners. Its function can be also encapsulated as a post processor component of Machine Translation (MT) and Optical Character Recognition (OCR) system. One of the major limitations of using rule-based parser is the knowledge

acquisition bottleneck and the inability to reliably capture the syntactic structure of free word order language like Bangla using Context Free Grammar rules. To the best of our knowledge, till now there is no robust rule-based parser is available for Bangla language. This observation has motivated elegant probabilistic and statistical interpretation of free word order languages. It also inspired a great deal of attention towards learning syntax from completely unannotated text. But most of the existing empirical error detection models have been hampered by unavailability of sufficiently large annotated learner's error corpora. There is a dearth of annotated error learner corpora of Bangla text depending on learner's age variation and social and educational influences. One of the major problem of building error corpus from learners' data is that the process is very time consuming and required linguistic knowledge to examine each sentence of learners' text to determine nature and density of errors. To overcome this problem, a corpus of ungrammatical Bangla sentences has been created automatically considering performance errors and language learning errors that occur frequently. This paper is more closely aligned to the task of automatic error corpora creation and does not focus on the methodology of an actual grammar checking system that can be built using the corpus. Before starting our discussion on automated error corpus creation methodology, we provide a background on the origin and linguistic aspects of Bangla language and illustrate types of text error of Bangla Second Language Learners at the time of writing text.

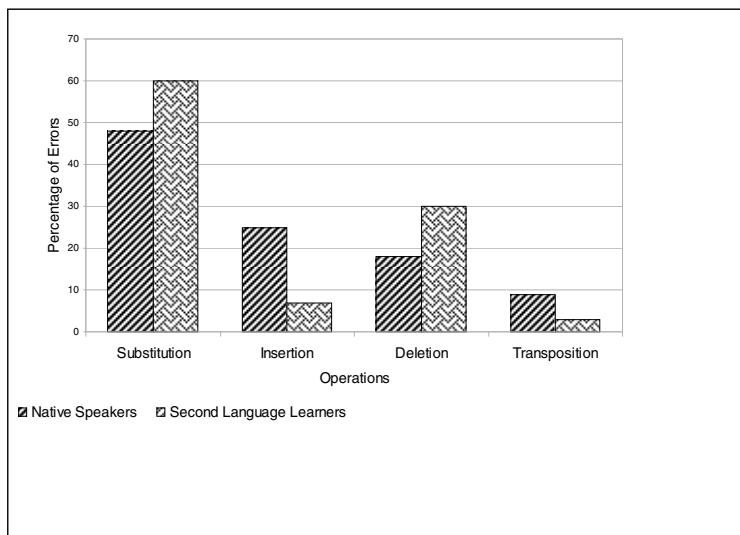
## 2 Background

Bangla is the fifth popular language in the world and the second in India. It is the national language of Bangladesh. This language belongs to the Indo-Aryan family and originated from Prakrit which is a sister language of Sanskrit. Sister languages of Bangla are Oriya, Magahi and Maithili in the west and Assamese in the north east of India. Bengali and Assamese are the eastern most languages of the Indo-European family of languages. When compared to languages like English, Bangla is largely free from words orders with some specific limitations. Like other Asian languages it follows a Subject-Object-Verb (S-O-V) pattern but orientation of these three atoms is flexible, i.e. S-V-O is allowable but not popularly used. Inspite of these free movements there is an invisible bonding between words having a mutual attraction towards each other which is governed by the property "Valency".

### 2.1 Errors in Text

It has been seen that many people are fluent in speaking Bangla language but their writing skill is appalling because of their lack of grammatical knowledge of the language and oversight in the time of writing. Even professional writers occasionally succumb to such errors. Bangla Second Language Learners often commit grammatical mistakes while writing text because of their lack of language knowledge (Language Learning Error) and due to oversight, carelessness or tiredness (performance error). Performance errors can occur mainly due to four operations: insertion, deletion,

transposition and substitution. When an error involves more than one operation, it is known as Composite Error. There are two primary concerns at the time of automatic error corpus creation, first one being linguistically realistic and the second one is to mimic the error scenarios that happen normally. To analyse the kind of naturally produced error scenario we have collected 1500 sentences from 10 standard native students' exam papers of Bangla and also have collected second language learners' data from students whose first language is either Hindi or Oriya or Telegu. Performance errors and language learning errors occurred in their text are then carefully analysed. Exam papers are collected with the assumption that students make more mistakes in the time of examination as they are usually in a hurry to complete their answers within the limited time period. In the course of studying Second Language Learners text, it has been found that the proportion of errors occurred by substitution operation is much more than any other operations. Figure 1 shows the proportion of performance errors caused by each of the four operations.



**Fig. 1.** Proportion of Errors in Native Speakers and Second Language Learners Corpus

The Native Speakers and the Second Language Learners make same kinds of mistakes such as misuse of punctuation and cohort/homophones [12]. But study shows that Second Language Learners make much more mistakes than native speakers. Most frequent error types produced by native speakers may not be produced by second language learners. For example, errors generated while writing complex sentences are infrequent for language learners, as most of the time language learners avoid writing complex sentences. They write complex sentences only when they have enough confidence in their ability to construct them correctly. Second Language Learners can be of two types viz. L1 and L2. Kind of errors produced by L1 Language Learners are influenced by their native language. When native languages are similar but not identical, L1 produces errors due to negative transfers. They fail to

find exact equivalence between these two languages. On the other hand, L2 Language Learners produce errors because of their incomplete knowledge of syntactic and/or morphological irregularities. They face trouble due to the novelty of the new language [12]. After analyzing the collected Bangla second language learners' data we came to know that the above statements (quoted in [12]) are also true for Bangla language. Therefore, learners who learn Bangla language having the background of Oriya, Assamese or Hindi as native language produces different kinds of errors than learners having native languages like Malayalam, Tamil, Telegu or English. We have classified the types of errors according to the operations involved in performance error and also depending on language learning errors. We shall now elaborate below different kind of errors depicted by second language learners.

### 1. Transposition Operation:

Incorrect Sentence:

Bangla: *theke gaachha phala pa.De*

English: *from tree fruit falls.*

Here the Post position *theke (from)* is placed before noun *gaachha (tree)*.

Correct Sentence:

Bangla: *gaachha theke phala pa.De.*

English: *Fruit falls from tree.*

### 2. Addition Operations:

#### a) Repeated words:

Bangla: *aami ekati \*bhaala bhaala Chele*

English: *I am a \*good good boy*

#### b) Unnecessary words:

Bangla: *paramaaNu anu apekShaa \*adhika kShudratara*

English: *atom is \*more smaller than molecule.*

### 3. Deletion Operations:

#### a) Implicit Subject:

Bangla: *\*[ ] tomaara maÑgala karuna* (Subject *iishbara* is missing here)

English: *May \*[ ] bless you.* (Subject: *God* is missing here)

#### b) Implicit Verb:

Bangla: *tumi ki maadhyamika pariikShaa \*[ ]* (Verb: *debe* is missing here)

English: *Will you \*[ ] matriculation exam?* (Verb: *give* is missing here)

### 4. Substitution Operations:

#### a) Similar word or Cohort replacement:

Incorrect Sentence:

Bangla: *\*bale baagha thaake<sup>1</sup>*

English: *\*tell tiger lives*

<sup>1</sup> All Bangla examples are given in ITRANS format.

\* Indicates error word in the sentence.

Correct Sentence:

Bangla: *bane baagha thaake*

English: *Tiger lives in forest*

Here *bale* (tell) and *bane* (forest) are cohorts of each other but *bale* is verb and *bane* is noun. In literature this type of error is also known as real word spelling error.

## Types of Grammatical Errors

### 1. Tense Error:

Example 1:

Bangla: *aami prashnapatra pa.Daba o uttara diYechhilaama.*

English: *I will read the question paper and I gave the answer.*

Example 2:

Bangla: *gatakaala aami sinemaa Jaaba*

English: *Yesterday I will go to Cinema.*

Example 3:

Bangla: *Jakhaana aami darajaa khulachhilaama takhana se ghare Dhuke pa.Dechhila*

English: *When I was opening the door then he entered the room.*

### 2. Person Error:

Example:

Bangla: *chhaatraraa nishchaYa bidyaalaYa Jaabe Jadi \*se pariikShaa dite chaaYa.*

English: *student must goes to school if \*he wants to appear in the exam.*

Plural sense of student has been lost by the singular representation of 'he'.

### 3. Case Error: case marker associated with pronoun and noun may be replaced. For example in the sentence *eTaa \*kaakaaraa ba\_i* (English: *This is uncle's book*) the suffix *raa* of the noun *kaakaa* (uncle) is changed from genitive case '*ra*'.

### 4. Adjectival Suffix Error: In the sentence *\*daYaamaYii shikShaka aasachhena* (English: *The kind-hearted teacher is coming*) the female suffix *maYii* of the word *daYaa* (kindness) is changed from male suffix *maYa* which goes with *shikShaka* (male teacher).

### 5. Improper use of punctuation:

Example 1:

Bangla: *tomaara naama ki |*

English: *What is your name.*

Here the punctuation | is used instead of '?' symbol.

Example 2:

Bangla: *aami\*, dekhalam se aasachhe |*

English: *I, see he is coming.*

6. Sentence Fragment:

Example:

Bangla: *aami gaana gaa\_iba \*| jadi tumi naacha |*

English: *I will sing. if you dance.*

7. Invalid Subject-Verb agreement:

Subject and Verb have to agree with respect to number and person. *aami bhaata \*khaabena* (English: *I eat rice*) is an incorrect sentence because the subject *aami* (*I*) is the first person non honorific but the person information of the verb *khaabena* (*eat*) is third person honorific.

8. Count Error:

Example:

Bangla: *aamaara tinajana bandhu aachhe : jaYanta, raajiiba, debaaruna o saurabha |*

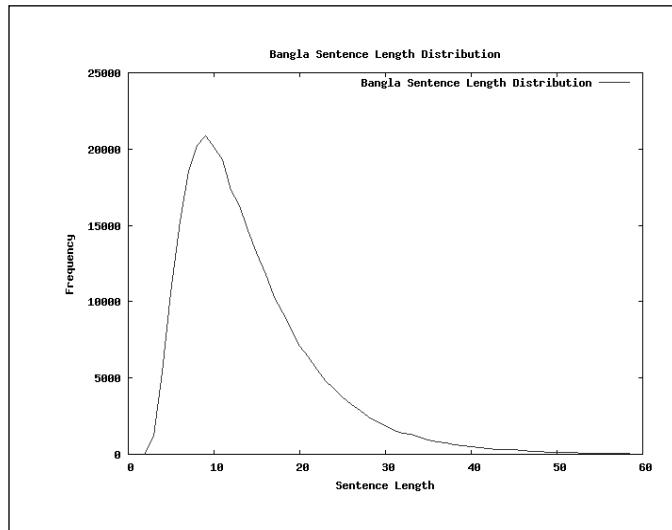
English: *I have three friends: Joyanta, Rajib, Debarun and Saurabh.*

## 2.2 Previous Work

Stemberger [4] introspects the performance error of native speaker spoken language and reports proportion of the four types of error as follows: substitution (48%) > insertion (24%) > deletion (17%) > combination (11%). Foster [3] has manually created an error corpus for English and has classified missing word errors based on Part of Speech tag of this missing word. According to her “98% of the missing parts-of-speech come from the following list (the frequency distribution in the error corpus is given in brackets): det (28%) >verb (23%) > prep (21%) > pro (10%) > noun (7%) > to (7%) > conj (2%)”. But manually creation of such corpus is very time consuming and non trivial task. Brockett et al. [15] created an artificial error corpus by introducing mass/count noun errors. They treated the error correction task in the machine translation point of view. Their aim was to apply Statistical Machine Translation (SMT) technique for converting ungrammatical sentences containing mass/count noun errors to grammatical sentences. Wagner, Foster, and Genabith [2] have suggested a novel approach of automated error corpus creation. They have carried out a detailed analysis of Missing Word Errors, Extra Word Errors, Agreement Errors and Covert Errors. Lee and Seneff [14] created artificial error corpora by introducing verb form errors. To mimic the real life errors, Foster and Anderson [16] designed the GenERRate tool. Their algorithm generates error corpus by introducing error along the line of the previously specified real life error templates.

## 3 Experimental Data Set

For our analysis, Bangla well-formed unicode sentences were collected from the web of various domains including literature, science, sports, music and news wire (2005-2010). We assumed that the syntax and semantics of the collected sentences are correct as they are mostly collected from different news wires which are normally edited and proof-read. Corpora from multiple domains have been collected to avoid



**Fig. 2.** Bangla Sentence Length Distribution

the skewed distribution of data. From this set of collected Bangla sentences (approx 4 lakh 80 thousand), sentence length distribution has been measured. It is found that sentences containing 11 words are the most frequent in this corpus. Figure 2 shows the Bangla Sentence length distribution.

## 4 Methodology

Now we will discuss our novel approach for error corpus generation. The procedure is as follows:

### Step-1

If a grammatical sentence contains  $n$  words then transposition between two consecutive words can generate  $(n-1)$  sentences with assumption that only one transposition done in each sentence. Table 1 shows 3 sentences generated from a sentence containing 4 words. Though the last two examples in the table are grammatically correct, but transposition-2 is semantically weird and transposition-3 is relatively uncommon.

**Table 1.** Examples of Transposition Operation

Operation	Example
Source	<i>gaachha theke phala pa.De</i> <sup>2</sup>
Transposition -1	<i>thek gaachha phala pa.De</i>
Transposition -2	<i>gaachha phala theke pa.De</i>
Transposition -3	<i>gaachha theke pa.De phala</i>

**Step-2**

Transposition of highly collocated sequences surely induces noise in a grammatical sentence. Erroneous sentences have been automatically generated by changing the word order of different types of Bangla collocated words sequences collected from the corpus. We distinguish between the following three categories: echo words (if  $w_1w_2$  is a word sequence and  $w_2$  has no meaning), hyphenated words ( $w_1$  and  $w_2$  are connected by hyphen) and highly collocated words. Extraction of echo words and hyphenated words is simple. One can use a simple regular expression  $[a-zA-Z]+ \backslash-[a-zA-Z]+$  for collecting hyphenated words from corpus and  $[\s[a-zA-Z][a-zA-Z]+[\s[a-zA-Z]\2[\s[a-zA-Z]]^3$  for collecting echo words. For collecting collocated and co-occurred word sequences from corpus, a statistical approach [17] has been used. Variance( $\sigma^2$ ) of the number of words separating word  $w_2$  from word  $w_1$  have been estimated and low variance word sequences have been filtered using a statistical significance test (t-test) with 99.5% confidence level. The null hypothesis  $H_0$  is that the word sequences ( $w_1w_2$ ) appear independently in the corpus. These filtered word sequences are cross verified with Mutual Information (MI) values between  $w_i$  and  $w_j$ . The word sequences having higher Mutual Information and lower variances and having t-value greater than 2.57 (considering  $\alpha = 0.005$ ) have been considered as collocated words. MI between words  $w_1$  and  $w_2$  has been estimated as follows:

$$MI(w_1, w_2) = \log_2 \frac{p(w_1, w_2)}{p(w_1).p(w_2)} \quad (1)$$

$$\text{where } p(w_1, w_2) = \frac{Count(w_1, w_2)}{N}$$

and  $Count(w_1, w_2)$  is the number of sentences in which  $w_1$  and  $w_2$  co-occur and  $N$  is the number of sentences in the training corpus. Accordingly the probability of the denominator of Equation (1) is calculated.

<sup>2</sup> Bangla Sentence: *gaachha theke phala pa.De*  
 English Word Meaning: Tree from fruit fall  
 English Translation: Fruit falls from tree

<sup>3</sup> Python regex notation has been used here.

**Step-3**

Another way of generating erroneous sentences is by replacing a word with its cohorts and homophones. Cohorts are generated using regular expression by adding, deleting or substituting a single character or moving character sequences in a word. These generated words are then verified with spelling dictionary to ensure that the generated words are correctly spelled. In this process, if we assume that  $k$  number of words/cohorts can be generated on an average from a single word then  $k \times n$  sentences can be generated from a sentence containing  $n$  words. Instead of  $k^n$  sentences,  $k \times n$  sentences are generated as we are considering just replacement of one word at a time. We can reduce the value of  $k$  by considering only the nearest neighbor 4 keys (UP, DOWN, LEFT, and RIGHT) of the keyboard position for a particular character of a word in the time of generating cohort. Levenshtein Distance [18] (Edit Distance) also can be used to prune the over generated cohort words. Words having minimum edit distance with the original word are selected for the cohort list.

**Step-4**

By deleting a particular word from a sentence containing  $n$  words we can generate  $n$  sentences where each sentence containing  $(n-1)$  words. Table 2 shows 4 sentences generated from a sentence containing 4 words where each sentence containing 3 words.

**Table 2.** Examples of Deletion Operation

Operation	Example
Source	gaachha theke phala pa.De
Deletion - 1	theke phala pa.De
Deletion - 2	gaachha phala pa.De
Deletion - 3	gaachha theke pa.De
Deletion- 4	gaachha theke phala

**Step-5**

By addition a word from a vector  $\vec{W} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_v \end{bmatrix}$  in  $(n+1)$  possible position of a sentence containing  $n$  words, we can generate  $V \times (n+1)$  sentences where  $V$  is the length of the vector. Here we are considering one word is inserted at a time. Table 3 shows number of sentences generated by addition operation. Thus applying step-1 to step-5 we can generate approximately  $(n-1)+ k \times n+ n + V \times (n+1)$  sentences from a sentence containing  $n$  words.

**Table 3.** Examples of Addition Operation

Operation	Example
Source	<i>gaachha theke phala pa.De</i>
Addition 1	$\vec{W}$ <i>gaachha theke phala pa.De</i>
Addition 2	<i>gaachha</i> $\vec{W}$ <i>thek</i> <i>phala pa.De</i>
Addition 3	<i>gaachha theke</i> $\vec{W}$ <i>phala pa.De</i>
Addition 4	<i>gaachha theke phala</i> $\vec{W}$ <i>pa.De</i>
Addition 5	<i>gaachha theke phala pa.De</i> $\vec{W}$

**Step-6**

Figure 3 shows a  $N \times N$  tag association matrix which is generated after analyzing 5000 manually parts-of-speech (POS) tagged Bangla sentences having different syntactic categories. Every possible combination of two POS tag sequence is searched programmatically from this tagged corpus. On successful match, each cell of the matrix corresponding to the tag sequence is filled with 1, otherwise the cell contains 0. The cell with zero value indicates an invalid relationship i.e. POS tag of column  $N_i$  can not occur after tag of row  $N_j$ . In other words POS tag of  $N_i$  does not follow tag  $N_j$  row. For example Post position (PPS) cannot appear after intensifier (INT). Consulting this matrix, mal-rule can be generated which can be used for transposition of the word sequence of a sentence after being annotated by an automatic POS tagger.

	PN	CN	VB	PPS	PUNC	PR	CC	JJ	RB	DGT	INT	END
START	1	1	1	0	1	1	1	1	1	1	1	0
PN	1	1	1	1	1	1	1	1	1	1	1	0
CN	1	1	1	1	1	1	1	1	1	1	1	0
VB	1	1	1	0	1	1	1	1	1	1	1	0
PPS	1	1	1	0	1	1	1	1	1	1	1	0
PUNC	1	1	1	0	0	1	1	1	1	1	1	1
PR	1	1	1	1	1	1	1	1	1	1	1	0
CC	1	1	1	0	1	1	1	1	1	1	1	0
JJ	1	1	1	0	1	1	1	1	1	1	1	0
RB	1	1	1	0	1	1	1	1	1	1	1	0
DGT	1	1	1	1	1	1	1	1	1	1	1	0
INT	0	0	0	0	0	0	0	1	1	1	0	0
END	0	0	0	0	0	0	0	0	0	0	0	0
<hr/>												
<b>PN</b> PROPER NOUN												
<b>CN</b> COMMON NOUN												
<b>PR</b> PRONOUN												
<b>VB</b> VERB												
<b>RB</b> ADVERB												
<b>JJ</b> ADJECTIVE												
<b>INT</b> INTENSIFIER												
<b>PPS</b> POST POSITION												
<b>CC</b> CONJUNCT												
<b>PUNC</b> PUNCTUATION												

**Fig. 3.** POS tag association matrix

#### 4.1 Confidence Score and Mal-rule Filters

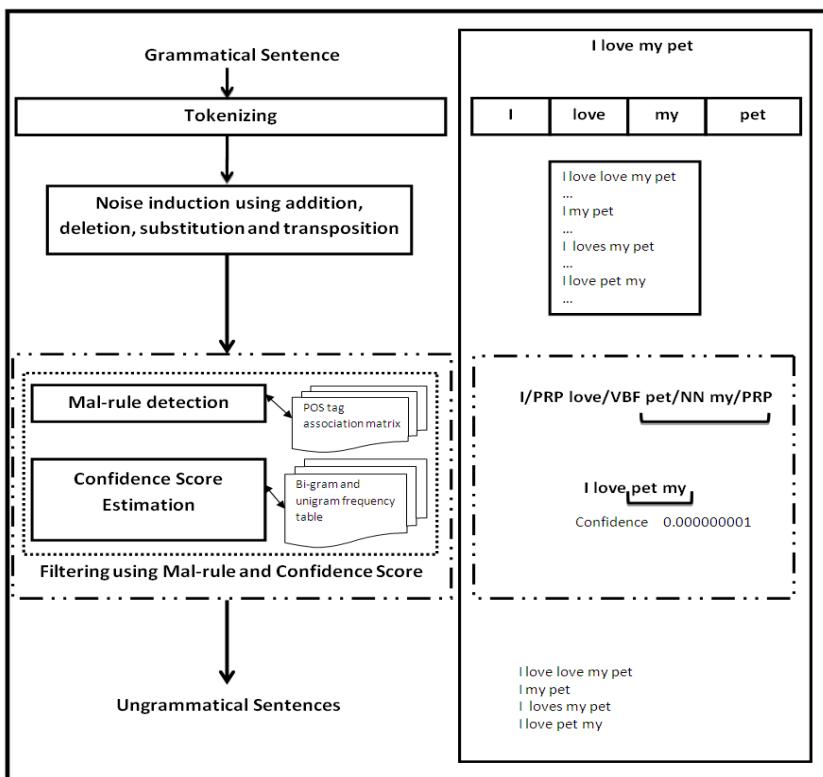
Following the above mention procedure, we can generate erroneous sentences from a corpus of grammatical sentences. Our procedure generates approximately  $\{(n-1)+k \times n + n + V \times (n+1)\}$  sentences from a sentence containing  $n$  words. Therefore, the number of generated sentences using this method increases with the number of words in a grammatical sentence. We have seen that the mode of the sentence length distribution of our collected Bangla corpora is 11. This implies that the upper bound of the number of sentences generated by our procedure is  $10 + k \times 10 + 10 + V \times 11$ . Those many sentences can be generated from a single sentence having 11 words. If we have 22000 11-word sentences in our corpus of approximately 480000 grammatical sentences, then  $22000 * \{10 + k \times 10 + 10 + V \times 11\}$  sentences can be generated using our method. Some Bangla sentences may have as many as 57 words but we are not considering such cases as such sentences are very infrequent (See Figure 2). Therefore filtering ungrammatical sentences from this set of  $\{(n-1)+k \times n + n + V \times (n+1)\}$  sentences is not a trivial task. In this stage proper sampling is required so that sentences indicative of more frequently made errors have higher probability of getting selected. Therefore we have applied both rule-based and statistical based approach for collecting significant sample from this population. Initially we pass the sentences through our HMM based semi-supervised POS tagger and then generated tag sequences are pass through mal-rule detector which collect the sentences containing improper pos tag sequences. We also have calculated the confidence score of each sentence by calculating bigram, Mutual Information (MI) and Relative Position Score [10]. A numeric score is assigned to determine the quality of the sentence. The sentence-level confidence measure is based on the score of each and every individual word in the sentence. Confidence score estimation using N-gram, measures the grammatical soundness of the sentence and MI based confidence score, measures the lexical consistency [19]. MI is used to detect presence of which word reduces the uncertainty of appearance of another word in the same sentence. Confidence score of a sentence using MI has been calculated as follows:

$$\begin{aligned}
 Sore(S) &= Score(w_1, w_2, w_3 \dots w_n) \\
 &= \frac{1}{n} \sum_{i=1}^n Score(w_i) = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1, j \neq i}^n MI(w_j, w_i)}{n-1}
 \end{aligned} \tag{2}$$

Here  $MI(w_j, w_i)$  is calculated using equation (1). MI based confidence measure do not take word order into account. It focuses on long range lexical relationships. For this reason, we have also estimated the relative position based confidence score. Confidence score of a sentence using Relative Position Score [10] has been calculated as follows:

$$\begin{aligned}
 RP_{Score}(S) &= RP_{Score}(w_1, w_2, w_3 \dots w_n) \\
 &= \sum_{j=2}^n \sum_{i=1}^{j-1} \frac{\left( \frac{freq_{Dep}(w_i, w_j)}{freq_{Ind}(w_i, w_j)} \right)}{j-1} \\
 &= \frac{n-1}{n-1}
 \end{aligned} \tag{3}$$

where  $freq_{Dep}(w_i, w_j)$  is the number of sentences in which  $w_i$  and  $w_j$  co-occur with a constraint that  $w_j$  appear after  $w_i$  in a sentence and  $freq_{Ind}(w_i, w_j)$  is the number of sentences in which  $w_i$  and  $w_j$  co-occur without any positional constraint. Mutual Information has been used for proper selection of the erroneous sentences generated by substitute operation. Low Mutual Information ensures that a word in the sentence is wrongly placed in the context of the other words. Bigram and Relative position scores have been used to select the erroneous sentences generated by transposition operations. The error corpora creation procedure with an English example is shown in Figure 4.



**Fig. 4.** Simplified functional diagram of automatic error corpora creation

## 5 Result and Discussion

Following the experimental procedure described in Section 4 we have generated erroneous sentences from randomly selected 1000 sentences from a corpus of grammatical sentences. Then these generated ill-formed sentences are filtered using mal-rule detector and depending on the confidence score (see sub section 4.1). After manually analysing the random sample of generated ill-formed sentences, we found that 87% of generated sentences are really ungrammatical. Most of these generated sentences have invalid POS tag sequences. Though some of the generated sentences have valid POS tag sequences but the word sequences in these sentences are infrequent. Experimental result also shows that 13% of that generated sentences are grammatical because insertion, deletion and substitution operation some time generates another grammatical construction. Figure 5 shows sample of Bangla erroneous sentences generated by our method from a grammatical sentence with their aforementioned confidence score. In this figure, the first sentence is a correct sentence and the remaining erroneous sentences are generated automatically. In this figure R\_S indicate the relative position score of a sentence.

Bangla Sentences	Confidence Score of the Sentence Using		
	B-gram	MI	R_S
<b>Correct Sentence</b>			
gaachha theke phala pa.De	7.40E-026	0.6502461741	0.4810439560
<b>Error Sentences for Transposition operation</b>			
theke gaachha phala pa.De	3.02E-033	0.6502461741	0.4334249084
gaachha theke pa.De phala	1.85E-025	0.6502461741	0.43477564103
gaachha phala theke pa.De	2.64E-029	0.6502461741	0.4641941392
<b>Error Sentences for Addition operation</b>			
gaachha theke phala phala pa.De	6.59E-033	0.8127406288	0.5180275743
gaachha gaachha theke phala pa.De	6.65E-033	1.05182834701	0.49725020350
gaachha theke theke phala pa.De	7.50E-029	0.7025908583	0.5030321530
<b>Error Sentences for Substitution operation</b>			
gaachha Theke phala pa.De	6.61E-033	-5.5447936457	0.3600000002
gaana theke phala pa.De	7.53E-030	-1.74079366467	0.39056776562
gaachha theke tala pa.De	3.76E-029	-3.3069949612	0.3964285715
maachha theke phala pa.De	7.58E-030	0.55386208974	0.40056776557
<b>Error Sentences for Deletion operation</b>			
gaachha phala pa.De	7.30E-026	0.59991544233	0.43750000000
theke phala pa.De	6.71E-023	0.23883813519	0.4367845696
gaachha theke pa.De	2.08E-018	0.64066086710	0.408854166667

**Fig. 5.** Erroneous sentences generated from a single sentence and selected according to the confidence score

Using echo words, hyphenated words and collocation collection methodology as discussed in the step 2 of section 4, we have collected desired results. Table 4 shows Bangla Echo words and Hyphenated words collected from the corpus.

**Table 4.** Bangla Echo words and Hyphenated words

Echo Words	Hyphenated Words
<i>oShudha TaShudha</i>	<i>aNu-paramaaNu</i>
<i>kha_i Ta_i</i>	<i>adala-badal</i>
<i>goYendaa ToYendaa</i>	<i>anumata-abhimata</i>
<i>chakaara bakaara</i>	<i>asukha-bisukha</i>
<i>chaNDaala phaNDaala</i>	<i>aaina-aadaalata</i>
<i>jaata paata</i>	<i>kaapa.Da-chopa.Da</i>
<i>nardamaa Tardamaa</i>	<i>Kaamanaa-baasanaa</i>

Transposition between them might cause error to be induced in a sentence. Transpositions of echo words are not allowable but transpositions of hyphenated words are allowed sometime. For example we may sometimes use “*baasanaa-Kaamanaa*” in place of “*Kaamanaa-baasanaa*”, though these appearances are very infrequent. Figure 6 shows some automatically collected collocated and co-occurred word sequences along with their relative position, mean and variance of relative positions, t-value and Mutual Information between these word sequences. Transposition of automatically collected echo words, hyphenated words and collocated words induce noise in a grammatical sentence and this procedure of automatic induction of noise gives a very good result.

W1	W2	Relative Positions	MEAN	SD	TVAL	MI
<i>jiijnjaasaa</i>	<i>karala</i>	1	1	0	5.99	0.02028
<i>chautrisha</i>	<i>nambara</i>	1	1	0	4	0.0106
<i>ghaad.a</i>	<i>naad.ala</i>	1	1	0	3.16	0.008667
<i>kamyunista</i>	<i>paatira</i>	1	1	0	2.65	0.005921
<i>chamake</i>	<i>uthala</i>	1	1	0	2.64	0.003883
<i>satyi</i>	<i>kathaa</i>	1	1	0	2.7	0.002006
<i>khrii</i>	<i>puu</i>	1,8,10	1.83	2.56	5.48	0.0295

**Fig. 6.** Erroneous sentences generated from a single sentence and selected according to the confidence score

## 6 Conclusion

In this paper, we discussed practical issues pertaining to automatically creating an error corpus by combining statistical and linguistic knowledge. Types of errors in the time of writing text are analysed in detail. Then a methodology of automatic error corpus creation with appropriate manual intervention has been discussed. Issues pertaining to creating erroneous sentences resulting from pronoun referencing error,

state error, time error, and other semantic errors fall outside the scope of this paper. Though the present work focuses on the most frequent grammatical errors in Bangla written text, detail taxonomy of grammatical errors in Bangla is also presented here, with an aim to increase the coverage of the error corpus in future.

As part of future work, we plan to devise a more principled approach to sampling the auto generated error corpus in the boundary cases and also to ensure that automatically generated error sentences will mimic the naturally occurring learners' errors. A statistical classifier can make use of active learning to bootstrap the corpus creation process. We hope that the research reported in this paper encourages other researchers in Indian Languages to build robust grammar checkers using the error corpus we built and also contribute further to the growth of the corpus. A similar approach combining linguistic and statistical approach can also be tried for developing error corpora in other Indian Languages where such resources are not available as of now.

## References

1. Kamp, H., Reyle, U.: *From Discourse to Logic:Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representatio*. Studies in Linguistics and Philosophy. Kluwer Academic Publishers (1993)
2. Wagner, J., Foster, J., van Genabith, J.: A Comparative Evaluation of Deep and Shallow Approach to the Automatic Detection of Common Grammatical Error. In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*, pp. 112–121 (2007)
3. Foster, J.: Good Reasons for Noting Bad Grammar: Empirical Investigations into the Parsing of Ungrammatical Written English, Phd. Thesis, University of Dublin, Trinity College, Dublin, Ireland (2005)
4. Stemberger: Syntactic errors in speech. *Journal of Psycholinguistic Research*, 313–345 (1982)
5. Thurmail, G.: Parsing for Grammar and Style Checking. In: *Proceedings of the 13th International Conference on Computational Linguistics*, pp. 365–370 (1990)
6. Bustamante, F.R., Leon, F.S.: GramCheck: A grammar and style checker. In: *Proceedings of COLING*, pp. 175–181 (1996)
7. Stanley, Goodman: An empirical study of smoothing techniques for language modeling. In: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* (1996)
8. Dagan, I., Karov, Y., Roth, D.: Mistake-Driven Learning in Text Categorization. In: *The Second Conference on Empirical Methods in Natural Language Processing*, pp. 55–63 (1997)
9. Powers, D.M.W.: Learning and Application of Differential Grammars. In: *Proceedings Meeting of the ACL Special Interest Group in Natural Language Learning*, pp. 88–96 (1996)
10. Liu, C., Wu, C., Harris, M.: Word Order Correction for Language Transfer Using Relative Position Language Modeling. In: *Proceedings of 6th ISCSLP*, pp. 1–4 (2008)
11. Michaud, L.N., Mccoy, K.F.: An intelligent tutoring system for deaf learners of written English. In: *Proceedings of the Fourth International ACM SIGCAPH Conference on Assistive Technologies*, pp. 13–15

12. Leacock, Chodorow, Gamon, Tetreault: Automated Grammatical Error Detection for Language Learners. Morgan & Claypool Publishers (2010)
13. Sjobergh, Knutsson: Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In: Proceeding of the International Conference on Recent Advances in Natural Language Processing, pp. 506–512 (2005)
14. Lee, Seneff: Correcting misuse of verb forms. In: Proceeding of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technology, pp. 174–182 (2008)
15. Brockett, Dolan, Gamon: Correcting ESL errors using phrasal SMT techniques. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 249–256 (2006)
16. Foster, Andersen: GenERRate: Generating errors for use in grammatical error detection. In: Proceedings of the Fourth Workshop on Building Educational Applications Using NLP, pp. 82–90 (2009)
17. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
18. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966)
19. Raybaud, S., Langlois, D., Smaili, K.: Efficient combination of confidence measures for machine translation. In: Proc. INTERSPEECH, pp. 424–427 (2009)

# Predictive Text Entry for Agglutinative Languages Using Unsupervised Morphological Segmentation

Miikka Silfverberg, Krister Lindén, and Mirka Hyvärinen

University of Helsinki  
Department of Modern Languages  
Helsinki, Finland  
`{miikka.silfverberg,krister.linden,mirka.hyvarinen}@helsinki.fi`

**Abstract.** Systems for predictive text entry on ambiguous keyboards typically rely on dictionaries with word frequencies which are used to suggest the most likely words matching user input. This approach is insufficient for agglutinative languages, where morphological phenomena increase the rate of out-of-vocabulary words. We propose a method for text entry, which circumvents the problem of out-of-vocabulary words, by replacing the dictionary with a Markov chain on morph sequences combined with a third order hidden Markov model (HMM) mapping key sequences to letter sequences and phonological constraints for pruning suggestion lists. We evaluate our method by constructing text entry systems for Finnish and Turkish and comparing our systems with published text entry systems and the text entry systems of three commercially available mobile phones. Measured using the keystrokes per character ratio (KPC) [8], we achieve superior results. For training, we use corpora, which are segmented using unsupervised morphological segmentation.

## 1 Introduction

Mobile phone text messages are a hugely popular means of communication, but mobile phones are not especially well-suited for inputting text because of their small size and often limited keyboard. There exist several technological solutions for text entry on mobile phones and other limited keyboard devices. This paper is concerned with a technology called *predictive text entry*, which utilizes redundancy in natural language in order to enable efficient text entry using limited keyboards (typically having 12 keys).

The subject of predictive text entry has been extensively studied, but the studies have mainly concentrated on predictive text entry of English. Because of the limited morphological complexity of English, these approaches have usually been able to rely on an extensive dictionary along with word frequencies, since a sufficiently large English dictionary almost eliminates the problem of out-of-vocabulary (OOV) words. E.g. [5] reports low OOV word rates of 1.42% for a training set containing the 40,000 most frequent words in the North American

Business News Corpus and a test set consisting of 54,265 sentences from the same corpus.

For morphologically complex languages like Finnish and Turkish, productive inflection, derivation and compounding raise the number of OOV words regardless of the size of the dictionary, i.e. the vocabulary growth rate does not converge [2]. This means that OOV words present a serious problem for dictionary based approaches to predictive text entry of languages like Finnish and Turkish.

In this paper we present an approach to predictive text entry based upon a morphologically segmented training corpus, which is used to construct a probabilistic model of morphotax. We additionally use a probabilistic model on letter sequences and two phonological constraints, which constrain the results of the probabilistic models. We show that this combination delivers superior results compared with a system based on a colloquial dictionary and a morphological analyzer [11] for text entry of Finnish, when evaluated on actual text-message data using the keystroke per character ratio (KPC) [8]. Thus we achieve superior results to [11] without using labour intensive linguistic resources such as morphological analyzers. Additionally, we compare our method to the predictive text entry in three commercially available mobile phones and show that our approach gives superior KPC.

Apart from two phonological rules, our approach is entirely unsupervised and data-driven, since we use the unsupervised morphological segmentation system Morfessor [3] for segmenting the training corpus and the tools for constructing POS-taggers from the HFST interface [7]. We show that our method can also be applied to another agglutinative language<sup>1</sup> besides Finnish, namely Turkish. We compare the Turkish text entry system with an existing text entry system, which is based on a Markov model on letter sequences and show that our approach gives a substantial improvement in KPC.

The paper is structured as follows. In Section 2 we present some earlier approaches to predictive text entry. In Section 3, we present the components of our model for text entry and explain how these models are combined into a system for predictive text entry. In Section 4 we describe the training and test corpora used in constructing and testing predictive text entry systems for Finnish and Turkish together with the phonological rules which are used to realize Finnish vowel harmony. Evaluation of the systems is presented in Section 5 and the results are discussed in Section 6. Finally we present some concluding remarks and future work directions in Section 7.

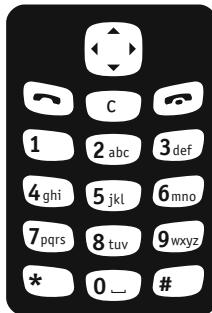
## 2 Related Approaches to Text Entry

The mobile phone keypad is a so called *clustered keyboard*, where each key can be used to enter several letters. E.g. on the Finnish mobile phone keypad in Figure 1 key “2” is used to enter the letters “a”, “b”, “c”, “ä” and “å”.

<sup>1</sup> Agglutinative languages are characterized by extensive use of inflectional and derivational affixes as well as compounding.

The original method for text entry is the so called *multitap-method*. When entering text in multitap mode, each key is pressed multiple times to scroll through the list of letters that are associated with the key. As text-messages have gained popularity, other faster methods for text entry have been devised. These can broadly be classified into *movement minimization techniques*, which concentrate on keypad layout, and *language prediction techniques*, which use linguistic models to disambiguate ambiguous user input [10].

The most widely used language prediction techniques are based on a dictionary. They disambiguate suggestions based on word frequencies. The best known example of a dictionary based system is the commercially successful T9-system [4]. There are many variants of dictionary based methods. E.g. some methods try to guess the word before all characters have been typed. Some approaches also include information on the probability of word sequences [10].



**Fig. 1.** The 12-key keypad of a typical Finnish mobile phone. There are three letters in the Finnish alphabet “ä”, “å” and “ö”, which are not shown on the keypad. The letters “ä” and “å” are entered pressing key “2” four times and five times respectively. The letter “ö” is entered by pressing key “6” four times.

As we noted in the introduction, dictionary based methods are not optimal for agglutinative languages, where the OOV rate remains high even with large dictionaries. Two alternative approaches better suited for agglutinative languages are known to the authors: prefix-based disambiguation [9] and disambiguation of output using a probabilistic model on letter sequences [13]. The methods resemble each other. Both methods use the previous letter context to guess the next letter, but in the prefix-based approach, an incorrectly guessed letter is corrected immediately after it has been entered. Conversely, when using a probabilistic model on letter sequences, the user first inputs all letters in the word and then scrolls through a list of suggestion words matching the input.

Our own method utilizes a similar probabilistic model on letter sequences as [13]. The novel aspects of our method are (1) utilizing a morphologically segmented training corpus in order to construct a probabilistic model of words as morph sequences and (2) using phonological constraints for filtering impossible suggestions. To the best of our knowledge, this has not been tried before in

the domain of text-entry. In the related domain of speech recognition, similar approaches have yielded good results [2] for agglutinative languages.

### 3 A Probabilistic Model of Word Structure

Predictive text entry can be seen as a labeling task, where every key in a sequence of keys is assigned its most likely letter. The usual approach to such tasks is using stochastic models with hidden variables e.g. HMMs.

Though predictive text entry can be implemented fairly well using n-gram models (such as HMMs) on letter sequences, as exemplified by [13], there are problems with this approach. An HMM cannot encode very long dependencies inside words, which leads to difficulties since it is not possible to adequately separate stems from affixes or to handle long phonological dependencies like vowel harmony. Higher order HMMs are not useful in practice because of efficiency problems [13].

In order to construct a general prediction model, which still represents word structure at a higher level than at the level of single letters, we represent words as morph sequences, which are extracted from an automatically segmented training corpus.

To illustrate the usefulness of our approach we look at some Finnish word forms. Consider the word form “taloa” (sg. partitive case of the word house). Automatic segmentation of the training corpus might give the segmentation “talo+a”, into the stem “talo” and the ending “a”. If the word form “taloakin” (sg. partitive case of “talo” with the clitic “kin”) does not occur in the training data, we can still estimate its probability by utilizing the frequencies of the morph combinations “talo + a” and “a + kin”,

Our model for word structure is a Markov chain of morph sequences. Data sparseness is likely to be a serious problem, since there are tens of thousands of morphs, many of which only occur once. We therefore combine the Markov chain with an HMM which maps key sequences to letter sequences. The HMM does not utilize morph boundaries, so it gives some estimate for the probability of a word form like “a-l-a-t-a-l-o” (a common Finnish surname), even though the combination of morphs “ala + talo” would never have been observed in the training data and the morph sequence model would therefore be unable to give a good estimate for the probability of the compound word.

Finally many agglutinative languages like Finnish and Turkish incorporate phonological phenomena, such as vowel harmony, which can span over arbitrarily long distances in word forms. These phenomena cannot be adequately handled using n-gram models of morphs or letters, which has prompted us to include phonological constraints in our system.

The statistical models and phonological rules are implemented as weighted finite-state transducers, which allows us to combine them using the algebraic operations for finite-state transducers. Transducers are a natural choice for coding arbitrarily long dependencies such as vowel harmony.

### 3.1 A Hidden Markov Model for Predicting Letter Sequences from Key Sequences

We denote a sequence of mobile phone keys  $k_i$  of length  $n$  by  $K = (k_i)_{i=1}^n$ . Correspondingly, we denote a sequence of letters  $l_i$  of length  $n$  by  $L = (l_i)_{i=1}^n$ . For key  $k_i$ , we denote the corresponding set of letters by  $M(k_i)$ . E.g.  $M(2) = \{a, b, c, ä, å\}$  on a typical Finnish mobile phone keyboard. For key sequence  $K$ , we denote the set of corresponding letter sequences by  $M(K)$ .

The task of the letter model is to give the probability of a letter sequence  $L$  given a sequence of keys  $K$ . Naturally  $P(L|K) > 0$ , iff  $L \in M(K)$ . We give the standard third order HMM approximation for  $P(L|K)$  in equation (1). The second equality follows by noting that  $P(k_i|l_i) = 1$  for all  $i$ , since every letter corresponds to exactly one key. This effectively makes our HMM equivalent to a Markov chain. Three special letters  $l_{-2}$ ,  $l_{-1}$ ,  $l_0$  are required to make the approximation work. These buffer symbols are added both to the training data and the suggestions. To counteract data sparseness, we smooth probabilities using lower order HMMs as explained in the following subsection.

$$P(L|K) = \prod_{i=1}^n P(k_i|l_i)P(l_i|l_{i-3}, l_{i-2}, l_{i-1}) = \prod_{i=1}^n P(l_i|l_{i-3}, l_{i-2}, l_{i-1}) \quad (1)$$

### 3.2 A Markov Chain of Morphs

A morph of  $n$  letters in the training data is simply a sequence of  $n$  letters, so we denote it by  $L = (l_i)_{i=1}^n$ . A key sequence  $K = (k_i)_{i=1}^m$  corresponds to a sequence of morphs  $L_1 \dots L_s$ , where each  $L_j = (l_{j_i})_{i=1}^{n_j}$ , iff  $\sum_{j=1}^s n_j = m$  and  $l_{j_i} \in M(k_{n_1+...+n_{j-1}+i})$  for all  $l_{j_i}$ . We denote the set of morph sequences that correspond to a key sequence  $K$  by  $M(K)$ .

The task of the morph model is to assign a probability for each sequence of morphs in  $M(K)$  for the key sequence  $K$ . The probability of a sequence of  $s$  morphs  $(L_1, \dots, L_s) \in M(K)$  is given by the chain rule of probabilities in equation (2).

$$P(L_1, \dots, L_s) = P(L_1)P(L_2|L_1) \dots P(L_s|L_1, \dots, L_{s-1}) \quad (2)$$

We make the standard assumptions for a first order Markov model, namely that  $P(L_i|L_1, \dots, L_{i-1}) = P(L_i|L_{i-1})$ , which means that we assume that the probability of a morph occurring depends only on its left neighboring morph and the morph itself. Thus we can approximate equation (2) by equation (3).

$$P(L_1, \dots, L_s) = P(L_1)P(L_2|L_1)P(L_3|L_2) \dots P(L_s|L_{s-1}) \quad (3)$$

In practice we use a training corpus for estimating the probability  $P(L_i|L_{i-1})$ . For the morphs  $L_i$  and  $L_{i-1}$  we use the estimate in equation (4), where  $C(L_{i-1}, L_i)$  is the number of times the morph  $L_i$  followed the morph  $L_{i-1}$  in the training corpus and  $C(L_{i-1})$  is the count of the morph  $L_{i-1}$  in the training corpus.

$$\hat{P}(L_i|L_{i-1}) = C(L_{i-1}, L_i)/C(L_{i-1}) \quad (4)$$

Since many morphs  $L_i$  and  $L_{i-1}$  do not occur adjacently anywhere in the training corpus, we also utilize the unigram estimates  $\hat{P}(L_i) = C(L_i)/S$  when estimating the probabilities  $P(L_i|L_{i-1})$ . Here  $S$  is the size of the training corpus. The actual estimate for the probability  $P(L_i|L_{i-1})$  is given in equation (5). The coefficient  $a$  is determined by deleted interpolation (see [1]).

$$P(L_i|L_{i-1}) = \hat{P}(L_i|L_{i-1})^a \hat{P}(L_i)^{1-a}, \text{ where } 0 \leq a \leq 1. \quad (5)$$

### 3.3 Phonological Constraints

We use phonological constraints to filter the results given by the statistical components of the system. The result given by the system is thus the most probable string, which satisfies the phonological constraints. Formally they are two-level constraints, which can be implemented using the two-level compiler `hfst-twolc`<sup>2</sup>.

### 3.4 Combining Models Using Weighted Finite-State Calculus

Both the HMM on letter sequences and the morph sequence model are implemented as sets of weighted finite-state transducers. The models are compiled using the POS tagger tools, [12], in the `hfst-interface`<sup>3</sup>. We simply replace words and tags by keys, letters and morphs.

The input key sequence entered by the user is compiled into a finite state transducer, which codes all possible realizations of the key sequence as letter sequences. The realizations are weighted using the HMM model on letter sequences and the weighted letter sequences are coded into morph sequences. These morph sequences are then re-scored using the morph sequence model. Finally those morph sequences which do not satisfy the phonological constraints are filtered out.

In a last processing step, the morpheme boundaries are removed and the ten most likely letter sequences are extracted.

## 4 Data and Linguistic Resources

We trained predictive text entry systems for Finnish and Turkish to evaluate our method. We compare our results with two existing text entry systems by [11] and [13]. There are no standardized test materials for predictive text entry for Finnish or Turkish, but we were able to obtain the training materials and test materials used in the previous systems.

The training materials and test materials for both Finnish and Turkish were processed in the same way. All uppercase letters were transformed into lowercase letters and all words that included non-alphabetical characters were removed. This included among other characters such as numbers and punctuation except apostrophes in Turkish, which are used to signify the boundary between the stem and affix in some word forms.

<sup>2</sup> <https://kitwiki.csc.fi/twiki/bin/view/KitWiki/HfstTwolC>

<sup>3</sup> <http://hfst.sf.net>

## 4.1 Finnish

For training and testing the Finnish text entry system, we use the same data as [11], though in addition to the training data they use a morphological analyzer, which we do not utilize. The training material is extracted from Finnish IRC logs and contains some 350,000 words. The test material consists of 6,663 words of actual text message data<sup>4</sup>.

**Phonological Constraints for Finnish.** In Finnish a word form, which is not a compound word, cannot contain both back-vowels (“a”, “o”, “u”) and front-vowels (“ä”, “ö”, “y”). We implemented two two-level rules [6], which realize this constraint on a morphologically segmented word form.

Figure 2 shows one of the rules. The rule disallows an affix with front-vowels, together with a stem with back-vowels. The named regular expressions `Affix` and `FrontVowelAffix` are sets of known inflectional and derivational affixes in Finnish. The expression `BackVowelStem` denotes sequences of four or more characters, where all vowels are back-vowels.

```
"Front Vowel Harmony"
<[ FrontVowelAffix ]> /<== BackVowelStem Affix* _ ;
```

**Fig. 2.** Rule for Finnish front vowel harmony using the rule-syntax of hfst-twolc for rules whose center is a regular expression

## 4.2 Turkish

For training and testing the Turkish text entry system, we use the same material as [13]. It is a corpus of news paper text containing some 20 million words. The material is divided into a test corpus containing 2,597 words and a training corpus which includes the rest of the words in the material. Thus the training data and test data are disjoint.

With Turkish we do not use phonological constraints.

## 5 Evaluation

In this section, we present the results of experiments using the Finnish and Turkish training data and test data presented in the previous section. For Finnish we examine the impact of varying the amount of training data on the performance of the predictive text entry system. For Turkish we present results on the whole training material.

<sup>4</sup> The original test data contains 10,851 words, but it turned out that the latter part of the test data file is actually a unqualified list of words, which skews test results, so we decided to only use the earlier half of the material.

### 5.1 The Keystrokes Per Characters Ratio

In this paper we use the keystrokes per character (KPC) ratio for measuring the efficiency of text entry. The KPC ratio for a text entry method is computed as the average number of keystrokes required to input one letter in a test corpus. Following [13], we do not consider space characters as a part of the test data.

By examining the schematic picture of a mobile phone keypad in Figure 1, it can be seen that the key sequence needed to input the word “kukka” (flower) on a mobile phone with Finnish keypad and using the multitap input method is 5-5-8-8-5-5-<NEXT>-5-5-2. The <NEXT>-key is required after entering the first “k” in order to tell the text entry that the next press of key 5 starts a new symbol. This increases the number of keystrokes from 9 to 10. On test data consisting solely of the word “kukka”, the KPC ratio would thus be  $10/5 = 2.0$ .

When computing the KPC ratio for predictive text entry methods, we assume that multitap is used as a fallback method when entering OOV words, i.e. words that are not found among the suggestions given by the system. In detail, entering an OOV word requires:

1. Entering the keys for the letters used to write the word (one keystroke per letter).
2. Scrolling through the suggestions (9 keystrokes in our system, since 10 suggestions are given).
3. Deleting the last suggestion one letter at a time using a backspace key (one keystroke per letter).<sup>5</sup>
4. Switching to multitap mode using a special key (one keystroke).
5. Inputting the word in multitap mode (keystroke count varies depending on the word).
6. Switching back to predictive mode using a special key (one keystroke).

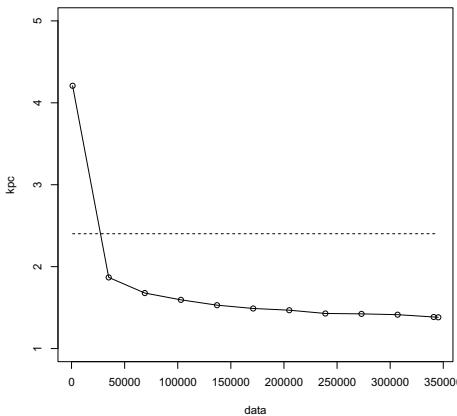
### 5.2 Results for Finnish

We constructed 12 text entry systems using different portions of the training data for Finnish presented in Section 4. We used the first 1,000, 35,000, 69,000, 103,000, 137,000, 171,000, 205,000, 239,000, 273,000, 307,000, 341,000 and 345,337 words respectively. The impact of the size of the training data is shown in Figure 3. The minimum KPC ratio 1.3748 was attained for the entire training data consisting of 345,337 words.

We also evaluated the effect of the different components on the KPC of the predictive text entry system. The results are shown in Table 1.

We compared our system to another published Finnish text-entry system by [11], which is based on a colloquial dictionary compiled from running text and a morphological analyzer. The authors do not evaluate their system using the KPC ratio, but we were able to obtain their test results and according to

<sup>5</sup> Many commercial phones make it possible to delete an entire word using one keystroke. However, deleting the word one letter at a time is consistent with the evaluation procedure used in [13].



**Fig. 3.** The effect of the amount of training data (horizontal axis) on the KPC ratio (vertical axis) of the Finnish predictive text entry system. The dashed line marks the KPC for multitap 2.4018. The minimal KPC ratio is 1.3738.

**Table 1.** KPC for Finnish Multitap and for using different components of our system. The third column shows the improvement over multitap.

Method	KPC	Improvement (%)
Multitap	2.4018	0.0
Letter n-grams	1.7368	27.7
Letter n-grams and morph sequence model	1.3751	42.7
Letter n-grams, morph sequence model and rules	1.3748	42.8

our experiments they achieve a KPC ratio of 1.6120. This means that our system achieves a 15.1% point decrease in KPC compared with their system using the same training materials and test data, but without using the morphological analyzer<sup>6</sup>.

In practice, ten suggestions for an input sequence is quite a lot. Few users are likely to scroll through ten suggestions especially if there are many non-words in the suggestion list. Therefore we also computed the KPC ratio for the entire training data as a function of the number of suggestions given by the system. The results are shown in Table 2.

Finally we wanted to compare our system to some commercially available text entry systems. To accomplish this, we took a list of thirty words chosen at random from the test data, entered the words into three commercially available

<sup>6</sup> When examining the test data used by [11], we discovered, that the latter half of the data consisted of a unqualified word list, which affected their results negatively. We have computed the KPC ratio for both our own system and the system of [11] using only the 6,663 first words in the test data.

**Table 2.** The effect of the number of suggestions on the KPC ratio of the Finnish text entry system using all of the training data

# of Sugg.	1	2	3	4	5	6	7	8	9	10
KPC	2.1478	1.7363	1.5872	1.5153	1.4602	1.4257	1.3998	1.3849	1.3798	1.3748

**Table 3.** The KPC ratio for a 30 word random sample from our test data using the multitap method, three commercial mobile phones and our text entry system. Only the first three suggestions for each input sequence were considered for the predictive text entry systems.

System	KPC
Multitap	2.3
Nokia C7	2.2
Nokia 2600	2.0
Samsung SGH M310	2.0
Our system	1.4

mobile phones and computed the KPC ratio. We also tested our own system using the words. Since the text entry system of the mobile phones did not give ten suggestions, we computed results only on the three first guesses given by each system. The results are shown in Table 3.

### 5.3 Results for Turkish

For Turkish, we trained one system using the entire 20 million word training corpus, which was presented in Section 4. We compare our results against the predictive text entry system by [13].<sup>7</sup> The results are shown in Table 4.

**Table 4.** KPC for Turkish using different input methods. The third column shows the improvement over the multitap method.

Method	KPC	Improvement (%)
Multitap	2.4386	0.0
Letter n-grams [13]	1.4382	41.0
Our method	1.1800	51.6

<sup>7</sup> For Turkish our computation gives the KPC ratio 2.4386 for the multitap method. This differs slightly from the figure 2.2014 given by [13]. Thus it is possible that our results are not entirely comparable to the results of [13]. The improvement in KPC ratio given for the method by [13] in Table 4 is computed using our figure for the KPC ratio of the multitap method. The improvement given by [13] is 35%.

## 6 Discussion

For Finnish, our system achieves a substantial 15.1% point drop in KPC ratio compared with the other system which utilizes a morphological analyzer. In their Finnish text entry system [11] give only 3 suggestions. Looking at Table 2 we see that the KPC ratio for our system is 1.5872, when only considering the three first suggestions, which is still lower than the KPC ratio 1.6120, which their system achieves. Considering, that except the phonological constraints, we use only language independent components, this is remarkable.

The phonological constraints seem to be having very little effect, as can be seen in Table 1. The decrease in KPC when using the phonological constraints is only about 0.1%. This may be a result of the unsupervised segmentation, which does not always succeed in finding the correct morpheme boundaries and therefore may prevent the rules from being triggered.

As Table 3 shows, our system outperforms three commercially available mobile phones on a thirty word test set chosen at random from our test data. This shows that our approach has great practical potential.

As can be seen in Table 4 our method achieves an additional 10% point reduction in KPC for Turkish compared with the system in [13]. Our KPC ratio 1.1800 needs to be related to the fact that our method can never achieve a KPC ratio lower than 1, since every letter in a word needs to be typed. We achieve a 52% reduction in KPC for the Turkish test data compared with the multitap method. A fast computation reveals that the maximal possible reduction is only 59%, which demonstrates that our system is nearly optimal on the Turkish data.

## 7 Conclusions and Future Work

We have demonstrated a highly accurate predictive text entry model, which can be constructed using unsupervised methods. Additionally linguistic rules can be added to improve the performance of the system.

There are several interesting future research directions. In order to reduce the KPC ratio to  $< 1$ , the system should be able to predict morphs before they are completely typed. We should also consider adding a model, which extends over word boundaries. Further, it would be interesting to examine the effect of the segmentation of the training corpus on the function of the phonological rules. A linguistically soundly segmented training corpus would probably allow the rules to act more often and thus improve the KPC ratio.

**Acknowledgments.** We wish to thank Cüneyd Tantuğ for the Turkish data, Sam Hardwick for the Finnish training data and Jarmo Wideman for drawing Figure 1. The first author is funded by the graduate school Langnet. Finally we wish to thank the anonymous reviewers for their valuable work.

## References

1. Brants, T.: Tnt - a statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing, pp. 224–231. ACL, Seattle (2000)
2. Creutz, M., Hirsimäki, T., Kurimo, M., Puurula, A., Pylkkönen, J., Siivola, V., Varjokallio, M., Arisoy, E., Saracilar, M., Stolcke, A.: Morph-based speech recognition and modeling of out-of-vocabulary words across languages. ACM Transactions on Speech and Language Processing 5(1) (2009)
3. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Transactions on Speech and Language Processing 4(1) (2007)
4. Grover, D., King, M., Kushler, C.A.: Reduced keyboard disambiguating computer. Patent US 5818437 (1998)
5. Klarlund, N.: Word n-grams for cluster keyboards. In: TextEntry 2003 Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods, pp. 51–58. ACL, Stroudsburg (2003)
6. Koskenniemi, K.: Two-level Morphology: A General Computational Model for Word-Form Recognition and Production. Ph.D. thesis, University of Helsinki (1983)
7. Lindén, K., Axelson, E., Hardwick, S., Silfverberg, M., Pirinen, T.: HFST—Framework for Compiling and Applying Morphologies. In: Mahlow, C., Piotrowski, M. (eds.) SFCM 2011. CCIS, vol. 100, pp. 67–85. Springer, Heidelberg (2011)
8. MacKenzie, I.S.: KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. In: Paternó, F. (ed.) Mobile HCI 2002. LNCS, vol. 2411, pp. 195–210. Springer, Heidelberg (2002)
9. Mackenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E.: Letterwise: Prefix-based disambiguation for mobile text input. In: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, pp. 111–120. ACM, Orlando (2001)
10. MacKenzie, I.S., William Soukoreff, R.: Text entry for mobile computing: Models and methods, theory and practice. Human-Computer Interaction 17(2), 147–198 (2002)
11. Silfverberg, M., Hyvärinen, M., Pirinen, T.: Improving predictive entry of Finnish text messages using IRC logs. In: Jassem, K., Fuglewicz, P., Piasecki, M., Przepiórkowski, A. (eds.) Proceedings of the Computational Linguistics—Applications Conference, Jachranka (2011)
12. Silfverberg, M., Lindén, K.: Combining statistical models for POS tagging using finite-state calculus. In: Pedersen, B.S., Nešpore, G., Skadina, I. (eds.) 18th Nordic Conference on Computational Linguistics, pp. 183–190 (2011)
13. Tantuğ, A.C.: A probabilistic mobile text entry system for agglutinative languages. IEEE Transactions on Consumer Electronics 56(4), 1018–1024 (2010)

# Comment Spam Classification in Blogs through Comment Analysis and Comment-Blog Post Relationships

Ashwin Rajadesingan and Anand Mahendran

VIT University

School of Computer Science and Engineering

Vellore, TN, India-632014

{ashwinr2008,manand}@vit.ac.in

**Abstract.** *Spamming* refers to the process of providing unwanted and irrelevant information to the users. It is a widespread phenomenon that is often noticed in e-mails, instant messages, blogs and forums. In our paper, we consider the problem of spamming in blogs. In blogs, spammers usually target commenting systems which are provided by the authors to facilitate interaction with the readers. Unfortunately, spammers abuse these commenting systems by posting irrelevant and unsolicited content in the form of spam comments. Thus, we propose a novel methodology to classify comments into spam and non-spam using previously-undescribed features including certain blog post-comment relationships. Experiments conducted using our methodology produced a spam detection accuracy of 94.82% with a precision of 96.50% and a recall of 95.80%.

**Keywords:** spam detection, comment spam, blog spam, text mining.

## 1 Introduction

Over the past decade, blogs have gained immense popularity in the Internet. A *blog* or *weblog* is an online journal or diary that usually allows interactions between the author and the readers through comments. According to WordPress[1], a blog publishing platform, their users alone produce an average of 500,000 new posts and 400,000 new comments everyday. Unfortunately, with such huge volumes of traffic in a largely unmoderated space, blogs have become a target for spammers. Spammers have expanded from spamming traditional e-mail and messaging systems to social networks, blogs, forums etc. Akismet[2], a plug-in for comment and trackback spam detection, has detected over 25 billion spams over the past four years in Wordpress blogs. Thus, with such high levels of spam in blogs, it is imperative that we constantly devise newer strategies to combat them.

The different types of spam in blogs include splogs, comment spam, trackback spam etc[3]. In this paper, we restrict our scope to detecting comment spam which is the most common type of blog spam. Studies indicate that 81% of blogs

have commenting systems[4] which allow the author to interact with the readers. A typical commenting system contains text fields for commenter's name, home-page url, e-mail address and a text area for typing comments. These commenting systems are exploited unethically by spammers who post advertisements, irrelevant links and malware in the text area as spam comments. These comments are generated through automated applications called *bots* which repetitively post irrelevant and often malicious content as comments.

Spam e-mail detection methodologies used in detecting spam comments have been reasonably successful[5] but cannot be viewed as a full-fledged solution to the comment spam problem. Their fallacies may be attributed to the inherent difference in the features of spam comments and spam e-mails. While the purpose of a spam e-mail is to coax the recipient into interacting with the solicited website, the purpose of a spam comment is to improve the search engine rankings of the advertised website. Also, unlike spam e-mails, spam comments (as shown in Fig. 2,3,4) are optimized according to the ranking algorithms of search engines such as Google through *Search Engine Optimization* (SEO) techniques. For example, the Google search engine employs its PageRanking algorithm to rank websites based on the weighted sum of their incoming links[6]. Thus, spammers use a SEO technique called *link building*[7] which involves repeatedly posting links in blogs, forums etc., to increase the incoming links and thereby, improving the search rankings.

Currently, blog owners and blogging platforms such as WordPress have adopted certain techniques to reduce comment spam. Some blog owners choose to manually monitor and moderate comments. While this process may be effective in removing spam completely, it is laborious and unfeasible especially if the blog attracts a large amount of traffic. Also, some blog owners disallow multiple postings of the same comments in their blogs. This approach prevents some but not all spam comments from being posted. Another approach is to prevent comment spam by distinguishing automated spamming bots from genuine commenters using CAPTCHAs[8]. CAPTCHAs are puzzles that usually involve recognizing letters or numbers from cluttered images that are difficult for bots to automatically identify. However, research has proved that this method is not foolproof and that it can be broken[9]. Yet another approach is to attach a "nofollow" link attribute to the commenting systems[10]. The "nofollow" attribute directs the search engine crawlers not to follow the links posted in comments. Thus, these links do not contribute to the page rank of the linked page during search queries. Unfortunately, spammers continue to spam even "nofollow"-attributed commenting systems as experiments conducted by SEO communities show that the links posted in such commenting systems are still followed by some crawlers[11].

The rest of the paper is organized as follows. In section 2, we discuss the past works related to comment spam. In section 3, we describe the dataset used for the validation of our methodology. In section 4, we identify and describe features required for our proposed methodology. In section 5, we provide a mathematical model that combines the features extracted in section 4. In section 6, we describe our experimental setup and the results obtained on applying our methodology.

In section 7, we conclude and explore the scope for further research on comment spam.

## 2 Related Work

Spam detection has been an active research area over the past decades with considerable work done primarily on email spam[12][13][14]. However, specific research on comment spam started only in 2005 and has yet to gain much prominence.

In 2005, Mishne et al[15] used probabilistic language models to detect spam comments. The difference in language models (calculated using a smoothed KL-Divergence) of the blog post, its comments and the pages that were linked by the comments were used in the spam detection process. A major drawback of this method is that spam classification is solely based on comparing language models. Thus, spam comments that have language models similar to that of the blog post may pass the spam filters without detection. In 2006, Han et al[16] proposed a collaborative filtering method for detecting spam links in blog comments. In their method, blog owners manually identify and share spam links through a trusted network of blogs called *trustroll*) to aid in spam detection. This approach can be applied only to user-hosted blogs (eg. Wordpress) and not developer-hosted blogs (eg. Blogger) as blog owners do not have the facility to create custom trustrolls in developer-hosted blogs. Also in 2006, Wong et al.[17] proposed a collaborative security system to detect spam comments. Their system automatically identified spam comments and constructed signatures which were distributed to a set of peers to assist in their spam detection process. In their system, for each spam comment detected, a signature is created and stored in a database before it is distributed to its peers in the network. This methodology is difficult to put into practical use because the database size and the network traffic increase with increase in spam comments. In 2007, Cormack et al[5] worked on spam filtering for short messages such as comments by analyzing and evaluating the available filtering systems such as Bogofilter, OSBF-Lua etc. They focused their analysis purely on comments and did not correlate the comments with their corresponding blog posts. In 2009, Bhattacharai et al[18] performed content analysis of spam comments to identify features such as number of word duplications, stop words ratio etc., which were used to train classifiers for spam detection. They obtained an accuracy of 86% in detecting spam comments using their approach.

Previous works on spam comment detection relied on methodologies using language models, collaborative approaches and content analysis techniques. However, these approaches did not take into extensive consideration, the meta-data in blogs such as time of posting, name of commenters etc., and focused purely on the text content of the blog post and its comments. To the best of our knowledge, our methodology is *the first to integrate features based on meta-data describing both comments and their relationship with blog posts for detecting spam comments*. In our work, we propose a novel methodology which combines the results from content analysis of comments and blog post-comment relationships to train

classifiers such as Naive Bayes, Support Vector Machines (SVM) etc., to detect spam comments. Our approach is more robust and accurate when compared to previous works as the comments are classified not only based on their properties but also based on their correlation with the blog posts.

### 3 Dataset

We use a blog corpus compiled by Mishne et al[15] for evaluating our methodology. This corpus contains 50 random blog posts with 1024 comments. The number of comments per blog post range from 3 to 96 and the average length of the comments is 41 words. The blogs and the comments are predominantly in English (over 90%). These comments were classified by human evaluators into spam and non-spam. The corpus contains 332 non-spam comments and 692 spam comments (about 67%). This is a realistic representation of the percentage of spam comments in the blogosphere and is in accordance with values obtained from recent observations[19]. All examples featured in this paper have been extracted from this corpus.

### 4 Feature Selection

Spam comments have certain defining features which distinguish them from non-spam comments. We analyzed comments and identified six such features which can be used to train classifiers in detecting spam comments. In this process, we used Beautiful Soup, a HTML parser library[20] and NLTK library[21] (Natural Language Toolkit) for extracting and evaluating blog posts along with their corresponding comments.

#### 4.1 Features Based on Comment Analysis

The following features are based purely on the the properties of comments. Here, we analyze the content and the meta-data related to the comments in order to identify features that aid in the spam detection process.

buy cheap **phendimetrazine**, phendimetrazine online, and phendimetrazine

Posted by: **phendimetrazine** at January 30, 2005 10:41 PM

**Fig. 1.** An example of a spam comment containing the commenter's name

**Presence of References to Own Name.** Commenting systems always provide a separate name field in which the commenter may input his/her name. Thus, genuine commenters generally never find the need to post their names in the comment body. However, spammers post multiple copies of keywords as names in both the name field and the comment body of commenting systems in order to increase the keyword density which improves search rankings[22]. In the corpus, it is observed that 93.17% of comments referring to the commenter's own name are spam comments. Thus, the *presence of references to own name* is used as a feature in comment spam detection. In Fig. 1, we observe such a spam comment where "phendimetrazine" is present in both the name field and the comment body.

pharmacy brooks pharmacy <http://brooks-pharmacy.buy-2005-top.com/> on line  
 pharmacy.buy-2005-top.com/ tramadol tramadol <http://www.buy-2005-top.com>

Posted by: [internet pharmacy](#) at March 24, 2005 08:27 PM

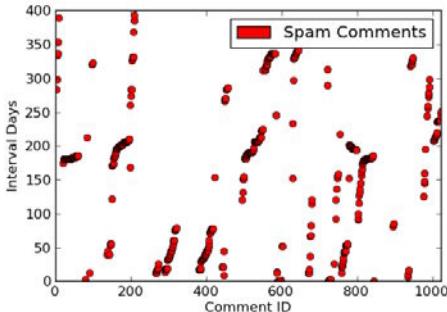
**Fig. 2.** An example of a spam comment containing homepage links

**Presence of Homepage Links.** Usually, links present in non-spam comments direct the user to a specific inner page rather than the homepage or the sub-domain homepage of a website. This is because genuine commenters provide topic-specific information which is available on these inner pages. On the other hand, spammers try to increase the search rankings of their entire website by post links directing to both the homepage and the inner pages of their website. In the corpus, we find that 91.05% of comments containing links directing to homepages are classified as spam. Hence, the *presence of homepage links* is used to distinguish spam from non-spam comments. Figure 2 shows a part of a spam comment containing links to homepages.

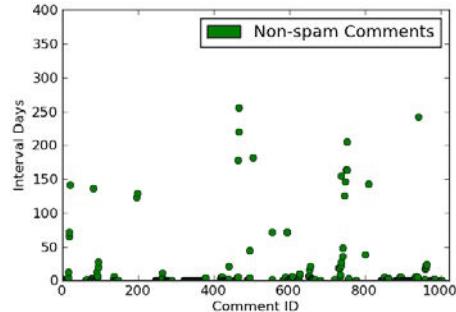
**Presence of Dictionary Words in Name Field.** Spammers often input keywords in the name field and website links in the comment body of commenting systems to improve the search ranking of their website for the inputted keywords. These keywords are usually dictionary words such as "shopping", "business" etc. Using the pyEnchant programming module[23], we observe that 84.34% of comments having dictionary words in their name field are classified as spam in the corpus. Thus, the *presence of dictionary words in name field* is an effective feature in the spam detection process. Figure 2 shows a part of a spam comment which contains "internet pharmacy" (both dictionary words) in the name field.

## 4.2 Features Based on Comment-Blog Post Relationships

The following features highlight the properties that link blog posts and comments. These features are especially effective in detecting spam comments as the correlation between blog posts and spam comments are generally very weak.



**Fig. 3.** Graph shows the distribution of spam comments with respect to the time interval between blog and comment posting



**Fig. 4.** Graph shows the distribution of non-spam comments with respect to the time interval between blog and comment posting

**Time Interval between the Dates of Blog Posting and Comment Posting.** As content in blogs is chronologically ordered, we can analyze the time interval between the dates of blog posting and comment posting. It is observed that as this time interval increases, the possibility of a comment being spam also increases. This measure is quite intuitive, for example, if a comment is posted, say, two years after the blog post was posted, the comment is most likely to be spam. We plotted two graphs (Fig. 3 and Fig. 4) showing the distribution of spam and non-spam comments in the corpus based on time interval. In both graphs (Fig. 3 and Fig. 4), “Interval Days” refers to the time interval between the dates of blog posting and comment posting (in days) and “Comment Id” refers to a unique number identifying each comment in the corpus. We observe that most non-spam comments are posted close to the blog post publishing date whereas spam comments have a wider distribution. This difference may be used in the distinguishing spam and non-spam comments and thus, *time interval between the dates of blog posting and comment posting* is a valuable feature which can be utilized in spam detection.

**Presence of References to Blog Post Author.** The “Comments” section in blogs serves as a discussion platform for commenters and blog post authors. The comments are usually directed at the author or at other commenters by

Spyware Stormer

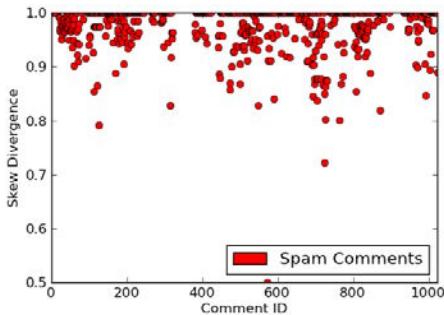
Posted by: [Spyware Stormer](#) at March 4, 2005 02:29 AM

Anti Spyware

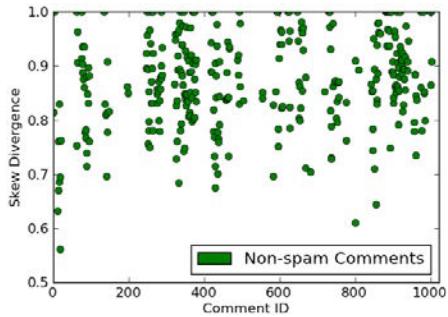
Posted by: [anti spyware](#) at March 4, 2005 03:41 AM

**Fig. 5.** An example of a spam comment containing another commenter’s name

usually referring to their names. In the corpus, it is observed that 91.67% of comments containing references to authors are non-spam comments. Hence, the *presence of references to blog post authors* is a useful feature to differentiate spam and non-spam comments. However, the presence of references to other commenters is not used as a differentiating feature because such references can be forged by posting multiple comments, with atleast one comment containing the other commenter's name. For example, in Fig. 5, the first comment is authored by "Spyware Stormer" and the following comment contains "Spyware", which is incidentally the first name of the previous commenter. Thus, an algorithm using the presence of references to other commenters as a feature would wrongly classify the two comments as non-spam.



**Fig. 6.** Graph shows the distribution of spam comments with respect to skew divergence



**Fig. 7.** Graph shows the distribution of non-spam comments with respect to skew divergence

**Skew Divergence.** As spam comments differ greatly in the language used when compared to their corresponding blog post, we compare the language model of the comment with that of the blog post. A language model is a probability distribution over the word sequences present in the text. In our approach, we calculate the language model of the blog post and the blog comments using maximum likelihood estimations and then the skew divergence[24], which is the difference between the two models, is calculated. The skew divergence is asymmetric and is a modification of the Kullback-Leibler divergence (KL-Divergence). The skew divergence  $S_\alpha$  between two language models  $l_1$  and  $l_2$ , is given by:

$$S_\alpha(l_1 \parallel l_2) = KL(l_2 \parallel \alpha l_1 + (1 - \alpha)l_2) \quad (1)$$

where

$$KL(l_1 \parallel l_2) = \sum_y l_1(y)(\log l_1(y) - \log l_2(y)) \quad (2)$$

Here,  $KL(l_1 \parallel l_2)$  is the KL-divergence of language models  $l_1$  and  $l_2$ ,  $y$  represents each word in  $l_1$  and  $\alpha$  is the skew divergence constant. It can be seen that

the skew divergence is a KL-divergence with  $l_1$  smoothed using  $l_2$  according to  $\alpha$ . It has been observed and proved by Lee et al[24] that higher values of  $\alpha$  tends to produce better results. Thus, we choose  $\alpha=0.99$  for our calculations. As skew divergence is asymmetric, we calculate both  $S(l_{post} \parallel l_{comment})$  and  $S(l_{comment} \parallel l_{post})$  and find their mean  $\bar{S}$  as follows:

$$\bar{S} = \frac{S_{0.99}(l_{post} \parallel l_{comment}) + S_{0.99}(l_{comment} \parallel l_{post})}{2} \quad (3)$$

$\bar{S}$  is normalized and plotted in two graphs (Fig. 6 and Fig. 7) which shows the distribution of the spam and non-spam comments based on their normalized mean skew divergence. In the two graphs (Fig. 6 and Fig. 7), “Skew Divergence” refers to the divergence values and “Comment Id” is a unique number identifying each comment in the corpus. From the graphs, we observe that most spam comments have a higher skew divergence when compared to non-spam comments. This observation concurs with our intuitive understanding that language models of spam comments differ greatly from that of blog posts. Hence, the difference in *skew divergence* values of spam and non-spam comments aids in detecting spam comments.

## 5 Combining Features

The features described in section 4 perform poorly in the spam detection process when taken into consideration individually, but when these features are combined to train a classifier, they perform comment spam detection accurately. Before calculating and combining feature values, appropriate preprocessing is performed on all the blog posts and comments in the dataset. Preprocessing includes stemming, removing stop words, punctuation etc., which improve the overall accuracy of our process. After preprocessing, values for all the features are calculated and are used to represent the comments.

Our approach can be mathematically defined as follows:

Let us assume that each comment instance is a point in an instance space. All comments can be described by the six features mentioned in section 4. These features have domains  $D_i$  (i=1 to i=6) as shown in table 1.

**Table 1.** Domain Details

Features	Domain Name	Domain
Presence of references to own name	$D_1$	Boolean
Presence of homepage links	$D_2$	Boolean
Presence of dictionary words in name field	$D_3$	Boolean
Time interval between the dates of blog posting and comment posting	$D_4$	Continuous
Presence pf references to blog post authors	$D_5$	Boolean
Skew Divergence	$D_6$	Continuous

As shown in table 1, time interval and skew divergence are in continuous domain, while the other features take boolean values (true represents the presence of that feature in the comment while false represents otherwise).

Thus, each comment instance  $C$  in the corpus can be represented as:

$$C = D_1 \times D_2 \times D_3 \times D_4 \times D_5 \times D_6 \quad (4)$$

These comment instances, along with their manual classifications (spam or not spam) are used to train learning algorithms to detect spam comments.

## 6 Experimental Setup and Results

The spam detection problem is essentially a binary-text classification problem (classification into spam and non-spam). In our methodology, we train classifiers such Nave Bayes, Support Vector Machines (SVM) etc., to obtain a model which is then tested for accuracy. We use different classifiers in order to analyze and evaluate the classifier that is most accurate for our classification problem.

Firstly, a dataset containing the six feature values and the manual classification for each comment in the corpus is compiled. Then, we use a ten-fold cross validation process[25] to test and evaluate the classifications made by the classifier. Here, the compiled data set is divided into 10 equal parts. The classifier is trained and tested 10 times where each time, a different part of the dataset is the testing set while the remaining parts are combined to form the training set. This process helps avoid the possibility of overfitting[26] and gives an accurate estimation of the accuracy of our classifier. The accuracy of the classifier is determined to be the total number of correct classifications divided by the total number of classifications made by the classifier. The results for different classifiers are shown in table 2:

**Table 2.** Results

Classifying algorithms	Accuracy	Precision	Recall
Naive Bayes Classifier	94.04%	95.92%	95.23%
Support Vector Machines (SVM)	92.57%	91.62%	97.97%
Logistic Regression	92.96%	94.92%	94.65%
Decision Trees (C4.5)	94.82%	96.50%	95.80%

From table 2, we observe that all learning algorithms perform very well with the extracted features values. We observe that SVM has the highest recall value but its precision and accuracy is less than that of some of the other classifying algorithms. Decision trees give the highest overall accuracy of 94.82% along with a precision of 96.50% and a recall of 95.80%. The accuracy obtained is 8.82% higher than the accuracy obtained by Bhattarai et al.[18] Also, the accuracy

obtained is much higher when compared to the 83% accuracy obtained by Mishne et al.[15] using the same spam corpus. Since, the relative cost of misclassifying a legitimate comment as spam is very high when compared to misclassifying a spam comment as legitimate, our focus has been to obtain a high precision in our system. Thus, with a precision of 96.50% obtained using decision trees, we believe that we have devised an excellent spam detection methodology with very high precision.

## 7 Conclusion and Future Work

From our results, we observe that the spam detection accuracy is vastly improved if both comment analysis and blog post-comment relationships are considered during the spam detection process. In our approach, spam comments need to closely mimic non-spam comments not only in their own properties but also in their relationships with the blog posts in order to deceive the classifier. Thus, our approach discourages spammers by making spamming more computationally expensive as spammers would need to post comments customized according to the blog post content. Also, we believe that our methodology is relatively language independent as most of the features mentioned in section 4 are not language dependent (such as date of comment posting, author's and commenter's names etc.). But, as the size of the corpus is small, we consider our results as a proof-of-concept and a base for further experimentation. In the future, we look to improve and expand the blog spam corpus. Also, we wish to include more features (such as those chosen by Bhattacharjee et al[18] and test the level of language independence of our methodology. We would also like to incorporate a collaborative spam detection module for better efficiency. Another possible extension of our work would be to use WordNet[27] to identify similar words present in comments and blog posts.

## References

1. Wordpress, <http://web.archive.org/web/20110307112536/http://en.wordpress.com/stats/> (retrieved in 2011)
2. Akismet, <http://web.archive.org/web/20110523025730/http://blog.akismet.com/2011/04/08/25-billion-pieces-of-spam/> (retrieved in 2011)
3. Thomason, A.: Blog spam: A review. In: Fourth Conference on Email and Anti-Spam (CEAS) (2007)
4. Sobel, J.: State of the blogosphere (2010), <http://web.archive.org/web/20110325150629/http://technorati.com/blogging/article/state-of-the-blogosphere-2010-introduction/> (retrieved in 2011)
5. Cormack, G.V., Hidalgo, J.M.G., Sanz, E.P.: Spam filtering for short messages. In: ACM Sixteenth Conference on Information and Knowledge Management, pp. 313–320 (2007)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999)

7. Malaga, A.R.: Search engine optimization black and white hat approaches. In: Zelkowitz, M.V. (ed.) *Advances in Computers: Improving the Web*. Advances in Computers, vol. 78, ch. 1, pp. 1–39. Elsevier (2010)
8. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: Captcha: Using Hard ai Problems for Security. In: Biham, E. (ed.) *EUROCRYPT 2003*. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003)
9. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: Breaking a visual captcha. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 134–141 (2003)
10. GoogleBlog, <http://web.archive.org/web/20110716122606/http://googleblog.blog-spot.com/2005/01/preventing-comment-spam.html> (retrieved in 2011)
11. ThoughtMechanics: Does nofollow attribute work? google says yes, studies say otherwise, <http://web.archive.org/web/20110104152458/http://www.thoughtmechanics.com/does-nofollow-attribute-work-google-says-yes-studies-say-otherwise/> (retrieved in 2011)
12. Drucker, H., Wu, D., Vapnik, V.: Support vector machines for spam categorization. *IEEE Transactions on Neural Networks* 10, 1048–1054 (1999)
13. Androutsopoulos, I., Koutsias, J., Chandrinou, K.V., Palioras, G., Spyropoulos, C.D.: An evaluation of naive bayesian anti-spam filtering. *Computing Research Repository* (2000)
14. Androutsopoulos, I., Koutsias, J., Chandrinou, K.V., Spyropoulos, C.D.: An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2000*, pp. 160–167. ACM, New York (2000)
15. Mishne, G., Carmel, D., Lempel, R.: Blocking blog spam with language model disagreement. In: *First International Workshop on Adversarial Information Retrieval on the Web*, pp. 1–6 (2005)
16. Han, S., Ahn, Y-Y., Moon, S., Jeong, H.: Collaborative blog spam filtering using adaptive percolation search. In: *World Wide Web Workshop 2006*, Edinburgh, UK (2006)
17. Wong, B., Locasto, M., Keromytis, A.: Palprotect: A collaborative security approach to comment spam. In: *Information Assurance Workshop*, pp. 170–175. IEEE (2006)
18. Bhattacharai, A., Rus, V., Dasgupta, D.: Characterizing comment spam in the blogosphere through content analysis. In: *IEEE Symposium on Computational Intelligence in Cyber Security, CICS 2009*, pp. 37–44 (2009)
19. Akismet, <http://web.archive.org/web/20110106110340/http://blog.akismet.com/-2005/10/29/rising-percentage/> (retrieved in 2011)
20. Richardson, L.: *Beautiful Soup Documentation* (2007)
21. Loper, E., Bird, S.: Nltk: The natural language toolkit. In: *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Association for Computational Linguistics, Philadelphia (2002)
22. Sedigh, A.K., Roudaki, M.: Identification of the dynamics of the google ranking algorithm. In: *13th IFAC Symposium on System Identification* (2003)
23. PyEnchant, <http://packages.python.org/pyenchant/> (retrieved in 2011)

24. Lee, L.: On the effectiveness of the skew divergence for statistical language analysis. *Artificial Intelligence and Statistics*, 65–72 (2001)
25. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *International Joint Conference on Artificial Intelligence*, pp. 1137–1143. Morgan Kaufmann (1995)
26. Hawkins, D.M.: The problem of overfitting. *Journal of Chemical Information and Computer Sciences* 44, 1–12 (2004)
27. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. Bradford Books (1998)

# Detecting Players Personality Behavior with Any Effort of Concealment

Fazel Keshtkar, Candice Burkett, Arthur Graesser, and Haiying Li

Institute for Intelligent Systems, The University of Memphis

Memphis, TN, USA

{fkshtkar, cburkett, a-graesse, hli5}@memphis.edu

**Abstract.** We introduce a novel natural language processing component using machine learning techniques for prediction of personality behaviors of players in a serious game, Land Science, where players act as interns in an urban planning firm and discuss in groups their ideas about urban planning and environmental science in written natural language. Our model learns vector space representations for various features extraction. In order to apply this framework, input excerpts must be classified into one of six possible personality classes. We applied this personality classification task using several machine learning algorithms, such as: Naïve Bayes, Support Vector Machines, and Decision Tree. Training is performed on a relatively dataset of manually annotated excerpts. By combining these features spaces from psychology and computational linguistics, we perform and evaluate our approaches to detecting personality, and eventually develop a classifier that is nearly 83% accurate on our dataset. Based on the feature analysis of our models, we add several theoretical contributions, including revealing a relationship between different personality behaviors in players' writing.

**Keywords:** Personality Detection, Classification, Conversation, Leary's Rose Framework, Natural Language Processing, Sentiment Analysis.

## 1 Introduction

Detecting personality and/or behavior in conversation is a hard task. In serious game (i.e., in chat rooms in serious games), players may have talk about different ideas than others they are chatting with during conversation. They also might be exposed to or be affected with different personalities or moods during the conversation by other players. On the other hand, players have various personality behavior, such as, helping, leading, aggressive, or dependent. Their personalities may cause them to behave varied within conversation. In our model, we aim to detect player's personality preferably without disrupting their relationship with others. We believe these findings will help human mentors manage players and interact in the right manner in conversation.

In this paper, we explore a component of natural language processing, using machine learning techniques, based on Leary's Rose framework (See Section 1.2)

for prediction of personality behaviors of players in a serious game, Land Science, where players act as interns in an urban planning firm and discuss in groups their ideas about urban planning and environmental science in written natural language. The possible dialog paths are defined by Leary Rose, a framework for interpersonal communication.

Our model learns vector space representations for various feature extraction. In order to apply this framework, input excerpts must be classified into one of six possible personality classes (See Section 1.2). We applied this personality classification task using several machine learning algorithms. More specifically, classification performance was measured using a Naïve Bayes classifier, Support Vector Machines algorithm, and Decision Tree named J48. And the raining set is performed on a relatively dataset of manually annotated excerpts. We extract a combination of features from psychology and computational linguistics. We develop and evaluate our approaches to detecting personality, and eventually develop a classifier that is nearly 80% accurate on our dataset. Based on feature analysis of our models, we add several theoretical contributions, including a relationship between different personality behavior in players writing.

## 1.1 Land Science Game

Land Science is a “serious game” created by researchers at the University of Wisconsin-Madison [1,14,3] that has been designed to simulate a regional planning practicum experience for students. During the 10 hour game, students play the role of interns at a fictitious regional planning firm (called Regional Design) where they make land use decisions in order to meet the desires of virtual stakeholders who are represented by Non-Player Characters (NPCs). Students are split into groups and progress through a total of 15 stages of the game in which they complete a variety of activities including a virtual site visit of the community of interest in which students familiarize themselves with the history and ecology of the area as well as the desires of difference stakeholder group. In addition, students get feedback from the stakeholders, and use a custom designed Geographic Information System (iPlan) to create a regional design plan. Throughout the game players communicate with other members of their planning team as well as a mentor (i.e., an adult who is representing a professional planner with the fictitious planning firm) through the use of a chat feature that is embedded in the game.

## 1.2 Leary’s Rose Framework

Leary’s Interpersonal Circumplex (also referred to as Leary’s Rose) has been used by researchers for many decades as a foundation for categorizing personality characteristics based on the statements people make [9]. Leary’s circumplex measures characteristics on two dimensions: the above-below axis represents variations from dominant (above) to submissive (below) and the opposed-together axis represents variations of cooperation from accommodating (together) to rebellious (opposed). The use of two axes allows the Rose to be easily separated into



**Fig. 1.** Leary's Rose Framework

**Table 1.** Our dataset with some examples of student's conversations that convey Leary's Rose categories

Category	Percent	Example
Leading	13.69%	Finish your task now so we can move on.
Helping	22.22%	How can I help you with that?
Competitive	24.48%	My plan is better than your plan.
Aggressive	03.04%	That idea is stupid. It will never work.
Dependent	29.98%	What should I do now?
Withdrawn	04.71%	Sorry, never mind, I'm not thinking.

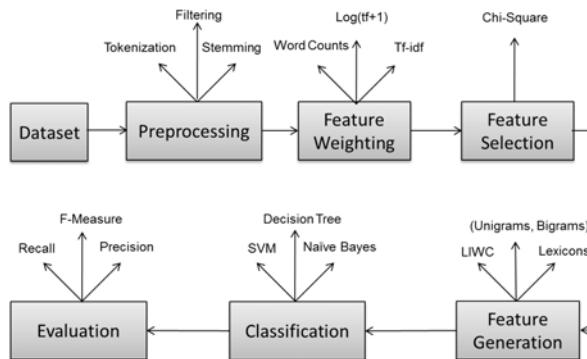
4 quadrants: above-together, above-opposed, below-together and below-opposed. Furthermore, each of these quadrants can be further split resulting in a total of eight different characteristic indices (see Figure 1).

The rest of the paper is organized as in the followings. Our model is described in Section 2 as well as dataset and human annotation performance. The Section 3 provides a summary of the experiments and results along with discussion. In Section 4, we describe related works and addressing personality detection in similar context, e.g., online chats. Finally, Section 5 is the conclusion and addresses the future work direction.

## 2 Our Model

### 2.1 Dataset Construction

Players in the epistemic game, Land Science, communicate with both other players and mentors using chat windows embedded in the game. For the purposes of these analysis, we only assessed the discourse of the players and did not analyze the discourse of the mentors. Annotation was done using the coding scheme (further discussed under Human Annotation) that was developed by the researchers based on the Timothy Leary's Interpersonal Circumplex Model [9]. The researchers selected 1,000 player excerpts (average length = 4.8 words) to



**Fig. 2.** Leary's Rose Framework

be analyzed. For our purposes, an excerpt was defined as a turn of speech that was taken by the student. On the other word, in the one excerpt occurred every time a student typed something and clicked “send” or hit “enter” in the chat function. The excerpts were selected from a larger set of 3,227 excerpts, so approximately 31% of the excerpts were used in the analyzed data set. In order to proportionally represent all stages of the game in the set that was analyzed, approximately 31% of the player excerpts were randomly selected from each stage of the game. Our model is illustrated in Figure 2 and in the following sections we describe the components of this model.

**Human Annotation.** For the purposes of this study, researchers developed a coding scheme based on Leary's Interpersonal Circumplex. This coding scheme looked specifically at all 4 quadrants of the Circumplex, but combined Helping and Co-operating into one category simply referred to as "Together" and also combined Aggressive and Defiant into one category simply referred to as "Opposed". In other words, the developed coding scheme focused on 6 categories: Competitive, Leading, Dependent, Withdrawn, Opposed and Together. Using this coding scheme, two trained researchers annotated the data set of 1,000 excerpts. The first series of training required the human annotators to independently code 200 excerpts randomly selected from the Land Science corpus. The kappa statistic was computed to assess inter-rater reliability on this set and agreement was fair (.33). Following this, the annotators discussed and refined any issues regarding the coding scheme and then annotated a new set of 1,000 excerpts that were randomly selected from the Land Science corpus. The kappa statistic was computed to assess inter-rater reliability on the second training set and agreement was substantial (.70). Results indicated increased reliability and thus completed the training of the human annotators. Once the two annotators were trained they independently annotated a set of 1,000 excerpts described in the data set portion of this paper.

**Lexicon Resources.** Sentiment-based lexical resources annotate words/concepts with polarity. To achieve greater coverage, we use four different sentiment-based lexical resources. They are described as follows.

1. SentiWordNet [4]: assigns three scores to Synsets of WordNet: positive score, negative score and objective score. When a word is looked up, the label corresponding to maximum of the three scores is returned. For multiple synsets of a word, the output label returned by majority of the Synsets becomes the prediction of the resource.
2. Subjectivity lexicon [18] is a resource that annotates words with tags like parts-of-speech, prior polarity, magnitude of prior polarity (weak/strong), etc. The prior polarity can be positive, negative or neutral. For prediction using this resource, we use this prior polarity.
3. General Inquirer [15] is a list of words marked as positive, negative and neutral. We use these labels to use Inquirer resource for our prediction.
4. Taboada [16] is a word-list that gives a count of collocations with positive and negative seed words. A word closer to a positive seed word is predicted to be positive and vice versa.

## 2.2 Feature Extraction

From this dataset we extracted a wide range of different features. The sentences were first parsed with Stanford POS Tagger, an English language parser (Kristina Toutanova and Christopher D. Manning. 2000.), which allowed us to extract linguistic information such as word tokens, lemmas, part-of-speech tags, syntactic functions and dependency structures. The actual feature vectors were then generated on the basis of this linguistic information by using a "bag of n-grams" approach, i.e. by constructing n-grams (unigrams, bigrams and trigrams) of each feature type (e.g. n-grams of word tokens, n-grams of part-of-speech tags...) and by counting for each n-gram in the training data how many times it occurs in the current instance. Additionally to these n-gram counts, we also included punctuation counts, average word length and average sentence length.

**Sentiment Score Feature.** Based on predictions of individual traits, we compute the Sentiment prediction for each trait with respect to a keyword in form of percentage of positive, negative and objective content. This is on the basis of predictions by each resource by weighting them according to their accuracies. These weights have been assigned to each resource based on experimental results. For each resource, the following scores are determined.

$$PositiveScore(s) = \sum_{i=0}^n P_i W_{P_i} \quad (1)$$

$$NegativeScore(s) = \sum_{i=0}^n N_i W_{N_i} \quad (2)$$

**Table 2.** LIWC features [13]

STANDARD COUNTS:
-Word count (WC), words per sentence (WPS), type/token ratio (Unique), words captured (Dic), words longer than 6 letters (Sixltr), negations (Negate), assents (Assent), articles (Article), prepositions (Preps), numbers (Number)
-Pronouns (Pronoun): 1st person singular (I), 1st person plural (We), total 1st person (Self), total 2nd person (You), total 3rd person (Other) PSYCHOLOGICAL PROCESSES:
-Affective or emotional processes (Affect): positive emotions (Posemo), positive feelings (Posfeel), optimism and energy (Optim), negative emotions (Negemo), anxiety or fear (Anx), anger (Anger), sadness (Sad)
-Cognitive Processes (Cogmech): causation (Cause), insight (Insight), discrepancy (Discrep), inhibition (Inhib), tentative (Tentat), certainty (Certain)
-Sensory and perceptual processes (Senses): seeing (See), hearing (Hear), feeling (Feel)
-Social processes (Social): communication (Comm), other references to people (Otherref), friends (Friends), family (Family), humans (Humans)
RELATIVITY:
-Time (Time), past tense verb (Past), present tense verb (Present), future tense verb (Future)
-Space (Space): up (Up), down (Down), inclusive (Incl), exclusive (Excl), Motion (Motion)
PERSONAL CONCERNs:
-Occupation (Occup): school (School), work and job (Job), achievement (Achieve)
-Leisure activity (Leisure): home (Home), sports (Sports), television and movies (TV), music (Music)
-Money and financial issues (Money)
-Metaphysical issues (Metaph): religion (Relig), death (Death), physical states and functions (Physcal), body states and symptoms (Body), sexuality (Sexual), eating and drinking (Eating), sleeping (Sleep), grooming (Groom)
OTHER DIMENSIONS:
-Punctuation (Allpct): period (Period), comma (Comma), colon (Colon), semi-colon (Semic), question (Qmark), exclamation (Exclam), dash (Dash), quote (Quote), apostrophe (Apostro), parenthesis (Parenth), other (Otherp)
-Swear words (Swear), nonfluencies (Nonfl), fillers (Fillers)

$$ObjectiveScore(s) = \sum_{i=0}^n O_i W_{O_i} \text{ where,} \quad (3)$$

$PositiveScore(s)$  = Positive score for each excerpts  $s$ ;  $NegativeScore(s)$  = Negative score for each excerpts  $s$ ;  $ObjectiveScore(s)$  = Objective score for each excerpts  $s$ ;  $n$  = Number of resources used for prediction;  $P_i, N_i, O_i$  = Positive, Negative, and Objective count of excerpt predicted respectively using resource  $i$ ;  $W_{P_i}, W_{N_i}, W_{O_i}$  = Weights for respective classes derived for each resource  $i$ .

**LIWC Features.** We can extract features derived from the LIWC output. In specific, LIWC counts and groups the number of instances of nearly 4,500

keywords into 80 psychologically meaningful dimensions. We create one feature for each of the 80 LIWC dimensions, LIWC, 80 dimensions summarized mostly under the following four categories:

- Linguistic processes: Functional aspects of text (e.g., the average number of words per sentence, the rate of misspelling, swearing, etc.)
- Psychological processes: Includes all social, emotional, cognitive, perceptual and biological processes, as well as anything related to time or space.
- Personal concerns: Any references to work, leisure, money, religion, etc.
- Spoken categories: Primarily filler and agreement words.

For each instance, we calculate the ratio of words in each category from the LIWC toolkit [13], as these features are correlated with the personality dimensions [13]. These features and their categories are shown in Table 2.

### 2.3 Automated Approaches to Personality Classification

We explain three automated approaches to classify detecting personality behavior, each of which utilizes classifiers trained on the dataset of Section 2.1. The features employed by each strategy are described here.

**Psycholinguistic Personality Detection.** The Linguistic Inquiry and Word Count (LIWC) software [13] is a popular automated text analysis tool used widely in the social sciences. It has been used to detect personality traits [10], to study tutoring dynamics [2], and, most relevantly, to analyze personality detection [10].

Since LIWC software does not include a text classifier, we create features derived from the LIWC output. In particular, LIWC counts and groups the number of instances of nearly 4,500 keywords into 80 psychologically meaningful dimensions. We construct one feature for each of the 80 LIWC dimensions, which can be summarized broadly under the four categories that explained in Section 2.2. Indeed, the LIWC2007 software used in our experiments subsumes most of the features introduced in other work. Thus, we focus our psycholinguistic approach to personality detection on LIWC-based features.

### 2.4 Classification

On the other hand, our classification approach to personality detection provides us to model both content and context with n-gram features. Specifically, we consider the following two n-gram feature sets, with the corresponding features lowercased and unstemmed: UNIGRAMS and BIGRAMS. Features from the our approaches just introduced are used to train Naïve Bayes, Support Vector Machine classifiers, and Decision Tree.

**Naïve Bayes (NB) Classifier.** Naïve Bayes (NB) classifier provides a simple approach and can view such a classifier as a specialized form of Bayesian network and it leans on two simple assumptions. First, it assumes that the predictive attributes are conditionally independent given the class. Then, it posits that no hidden or latent attributes influence the prediction process [6].

For a document  $X$ , with label class  $c$ , the Naïve Bayes (NB) classifier gives us the following decision rule [6]:

$$P(C = c|X = x) = \frac{p(C = c)p(X = x|C = c)}{p(X = x)}, \text{ where} \quad (4)$$

$$P(X = x|C = c) = \prod_i P(X_i = x_i|C = c) \quad (5)$$

We use John and Langley [6] Naïve Bayes classifier in Weka [5] to train our Naïve Bayes models on all three approaches and feature sets described above, namely LIWC, lexicons, UNIGRAMS, BIGRAMS. We also evaluate every combination of these features, but for brevity include only UNIGRAMS+BIGRAMS, which performs best.

**Support Vector Machine (SVM).** We also train Support Vector Machine (SVM) classifiers, which find a high-dimensional separating hyperplane between two groups of data. To simplify feature analysis in Section 5, we restrict our evaluation to linear SVMs, which learn a weight vector  $w$  and bias term  $b$ , such that a document  $x$  can be classified by:

$$y = \text{sign}(\vec{w} \cdot \vec{x} + b) \quad (6)$$

We use SMO [7] to train our SVM models on all three approaches and feature sets described above: LIWC, LEXICONS, UNIGRAMS, and BIGRAMS. We also evaluate every combination of these features, but for shortness include only LIWC+BIGRAMS, and LEXICON+BIGRAMS which performs best.

**Decision Trees.** We use J48, an open source Java implementation of the C4.5 algorithm in Weka [5] data mining tool to train our dataset for decision trees classifier. We evaluate all our approach on all combination of feature set, but we consider the features which performed best (UNIGRAMS+BIGRAMS, UNIGRAMS+LIWC). Our classification experiments are carried out with 10-fold cross validation on the corresponding dataset.

### 3 Results and Discussion

The model for classification personality strategies explained in Section 2 are performed using a 10-fold cross validation method under its default setting in Weka [5]. The parameters for model are chosen for each test fold based on standard cross validation experiments on the training dataset. All folds are chosen

**Table 3.** Automated classifier performance for three approaches based on 10-fold cross-validation experiments. Reported: Accuracy, Precision, Recall and F-measure are computed using Weka [5].

Approach	Features	Acc.	COM			DEP			LEA			WIT			COP			AGG		
			P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
LEXICAL	lexicons <sub>j48</sub>	61.95%	.67	.66	.67	.56	.66	.61	.61	.55	.58	.56	.54	.55	.66	.60	.63	.25	.21	.22
LIWC	liwc <sub>j48</sub>	59.30%	.57	.62	.60	.64	.67	.65	.52	.40	.45	.62	.42	.50	.60	.62	.61	.50	.53	.51
CLASSIFIERS	unigrams <sub>sum</sub>	60.54%	.74	.70	.72	.64	.63	.63	.50	.32	.50	.83	.39	.53	.52	.74	.61	.17	.33	.22
	bigrams <sub>sum</sub>	70.40%	.92	.65	.76	.92	.75	.82	.79	.40	.52	1	.44	.62	.50	1	.66	1	.17	.29
	liwc+bigramssvm	77.47%	.90	.75	.82	.93	.78	.85	.95	.64	.77	.93	.54	.68	.54	.95	.69	1	.2	.33
	lexicons+bigramssvm	<b>83.71%</b>	.96	.80	.87	.96	.84	.90	.98	.76	.86	1	.74	.85	.98	.76	.86	1	<b>.32</b>	<b>.48</b>
	bigrams <sub>nb</sub>	65.02%	.04	.87	.62	.87	.70	.77	.50	.21	.3	.80	.44	.57	.46	.96	.62	1	.16	.28
	unigrams <sub>b</sub> +bigram <sub>nb</sub>	60.53%	.77	.60	.67	.72	.64	.68	.50	.53	.51	1	.39	.54	.40	.74	.52	.67	.5	.57
	unigrams+bigram <sub>j48</sub>	62.78%	.83	.67	.74	.83	.62	.71	.46	.43	.45	.82	.47	.60	.42	.84	.56	.26	.22	.24
	unigrams+liwc <sub>j48</sub>	74.0%	.86	.80	.83	.81	.73	.77	.63	.64	.63	.85	.78	.81	.63	.75	.69	.50	.68	.57

so that each includes all instances from six classes; therefore, learned classifiers are always measured on dataset from unseen instances.

Table 3 shows the results of the top scores that we managed to achieve with each of the three classifiers over three approaches. We also use the combination of features and learner parameters that was determined to give the best accuracy by the classifiers. “Approach” column shows the model that have been tested, the “features” column indicates the types of features that have been used, the rest of columns indicates the results based on Accuracy, Precision, Recall, and F-measure (Acc., P, R, F) for all six classes.

We observe that our automated classifiers approaches achieve human judges (*kappa*) and baseline for most of feature sets. The statistical baseline for this six classes classification problem, considering the slight imbalances in the class distribution, is 30%. However there is an exception such as *Recall* for “aggressive” where does not performs significant. We can argue on this due to low number of instances in this class. However, this is expected given that human judges often focus on unreliable cues to aggressive utterances. If we look at the confusion matrix in Figure 3; Firstly, we note that most of the aggressive instances (8) classified as “helping” personality. Many other classes considered as “helping” as well. We figured out, this happened due to human judges evaluation, because the judges considered many small responses such as: OK, Yep, Thanks, Cool, etc as “helping” class. Secondly, as it shown in Table 1 the number of instances in “aggressive” class is low. We found out that the players are not often aggressive during chat conversation. It might be due to their work environment in that they are supervised by a human mentor during the game.

Interestingly, the psycholinguistic approach (LIWC<sub>j48</sub>) performs almost 30% more accurately than baseline rather than SVM or NB. Also j48 perform higher than SVM and NB on lexical subjective scores features. In overall, all the standard text categorization approach proposed in Section 2 performs between 9% and 53% more accurately than baseline. However, best performance overall is achieved by combining features from these two approaches. Particularly, the combined model LEXICONS+BIGRAMS<sub>SVM</sub> is 83.71% accurate at personality classification.

Surprisingly, models trained only on UNIGRAMS<sub>svm</sub> (60.54%), the simplest n-gram feature set, outperform LIWC (non text classification) approaches, and

a	b	c	d	e	f	<-- classified as
130	1	2	1	37	0	a = competitive
2	155	1	0	47	0	b = dependent
5	4	61	0	26	0	c = leading
0	0	0	14	12	0	d = withdrawn
2	1	0	0	146	0	e = helping
0	0	0	0	8	2	f = aggressive

**Fig. 3.** The Confusion Matrix performed by SVM Classifiers approach over BIGRAMS and subjective Lexicon features

**Table 4.** Top 16 highest weighted features learned by BIGRAMS+LEXICONS<sub>svm</sub> and LIWC<sub>svm</sub>. The results show for binary classification of “Helping, Aggressive” and “Leading, Dependent”.

BIGRAMS+LEXICONS <sub>svm</sub>	LIWC <sub>svm</sub>
Helping, Aggressive	Leading, Dependent
always want	six letters
didnt seem	pronoun
don t	personal pronoun
for me	i
is quite	we
it is	you
need to	she/he
no need	they
people don	impersonal pronouns
quite deadly	article
really that	verb
seem to	auxiliary verbs
slow down	past tense
speaking spanish	present tense
stop speaking	future tense

models trained on BIGRAMS<sub>nb</sub> (65.02%) perform even better. This suggests that a universal set of feature such psycholinguistic keyword personality (i.e., LIWC) can not be the best model for personality detection, and a context-sensitive approach (e.g., BIGRAMS) might be necessary to achieve state-of-the-art personality detection performance.

To better understand the models learned by these automated approaches, we report in Table 4 the top 16 highest weighted features for two pair classes (Helping, Aggressive & Leading, Dependent) as learned by BIGRAMS+LEXICONS<sub>svm</sub> and LIWC<sub>svm</sub>. From BIGRAMS+LEXICONS<sub>svm</sub> approach we have chosen classifier for classes “Helping” (with highest F-measure) and “Aggressive” (lowest F-measure), for LIWC<sub>svm</sub> approach we have chosen classifier for classes “Leading, Dependent” with similar reason. We note that player with “Helping” personality behavior tend to use some how similar language with “Aggressive” players; in particular, “need to” and “no need”, the former one can

be consider as “Helping” behavior and latest one can be regarded as “Aggressive” attitude. Accordingly, in term of global features such as psycholinguistic features (LIWC), “Leading” and “Dependent” players tend to use similar pronouns(personal or impersonal) (i.e.; i, we, you, she/he, they). Finally, when we look at Confusion Matrix (Figure 3), it turns out that all misclassified instances from “Aggressive” class fall into “Helping” class and similarly almost 75% of misclassified instances in “Leading” class are classified as “Dependent” class.

## 4 Related Work

To our knowledge, the only research has beed done specifically on the automatic classification of sentences based on Learys Rose for emotion detection is done by [17]. They described a methodology for a serious gaming project, deLearyous, which aims at developing an environment in which users can improve their communication skills by interacting with a virtual character in (Dutch) written natural language. In order to apply this framework, they classified the input sentences into one of four possible “emotion” classes (above, below, opp, tog, see Figure 1). They applied several machine learning algorithms, SVM, Naïve Bayes, Conditional Random field to obtained the calcification performance. For this, they used different features set from the their dataset (unigrams, lemma tri-grams and dependency structures). They obtained 52.5% accuracy around 25% over the baseline. In contrast, in our method we use Leary’s Rose framework to detect personality rather than emotion.

Mairesse et al [10,11] found that identification of personality (main Five in speech) by automatic analysis perform better than the baseline, and their analysis confirms previous findings linking language and personality, while revealing many new linguistic and prosodic markers. However, they had limitation for their method involving speech recognition that is recognition errors will introduce noise in all features except prosodic features, and prosodic features on their own are only effective in the extroversion model.

Another possible research that were done to let a machine learner determine the appropriate sentiment/emotion class. [12] and [8], for instance, attempt to classify LiveJournal posts according to their mood using Support Vector Machines trained with frequency features (word counts, POS-counts), length-related features (length of posts/sentences/...), semantic orientation features (using WordNet to calculate the distance of each word) and special symbols (emoticons).

## 5 Conclusion and Future Research

In this paper we have developed a dataset containing personality excerpts based on Leary’s Rose framework. By this, we have presented that the detection of personality behavior is more efficient than that of human judges. Consequently, we have presented three automated methods to personality detection, based on understanding from research in natural language processing, machine learning,

and psychology. We explore that while text classification based on n-gram (UN-IGRAMS, BIGRAMS) is the best particular detection approach, a combination method such as LIWC and Subjective Lexicons features along with n-gram features can achieve better performance.

Eventually, we have done several notable contributions. Particularly, our results indicate it is vital of take into account both the context, such as BIGRAMS, rather than precisely using a global set of personality indications (e.g., LIWC and Subjective Lexicons). We have also shown some findings based on the feature weights found by our classifiers that show the difficulties confronted by judges in annotating the dataset. Finally, we have found a possible connection between personality behavior by players, such “Helping, Aggressive” and “Dependent, Leading”, based on BIGRAMSs and LIWC similarities.

For future work, we want to include an extended experiment of the methods proposed in current research to sentiment analysis, opinion mining, as well as emotion detection in other domains. Also, we want to extend the method in this work to apply in Big-Five personality detection. It will help us to not only detect the player’s behaviors but also to detect introvert and extrovert players, and a focus on approaches with POS features might be useful.

**Acknowledgements.** This work was funded by the National Science Foundation (DRK-12-0918409). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these funding agencies, cooperating institutions, or other individuals

## References

1. Bagley, E.A.S.: Stop Talking and Type: Mentoring in a Virtual and Face-to-Face Environmental Education Environment. Ph.D. thesis, University of Wisconsin-Madison (2011)
2. Cade, W.L., Lehman, B.A., Olney, A.: An exploration of off topic conversation. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 669–672. Association for Computational Linguistics, Los Angeles (June 2010), <http://www.aclweb.org/anthology/N10-1096>
3. D’Angelo, C., Arastoopour, G., Chesler, N., Shaer, D.: Collaborating in a virtual engineering internship. In: Computer Supported Collaborative Learning Conference (CSCL), Hong Kong, SAR (2011)
4. Esuli, A., Sebastian, F.: Sentiword-net: A publicly available lexical resource for opinion mining. In: Proceedings of LREC 2006, Genova, Italy (2006)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)
6. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, pp. 338–345 (1995)
7. Keerthi, S., Shevade, S., Bhattacharyya, C., Murthy, K.: Improvements to platt’s smo algorithm for svm classifier design. Neural Computation 13(3), 637–649 (2001)

8. Keshtkar, F., Inkpen, D.: Using sentiment orientation features for mood classification in blogs, pp. 1–6 (September 2009)
9. Leary, T.: *The Interpersonal Diagnosis of Personality*. John Wiley and Sons Inc. (1957)
10. Mairesse, F., Walker, M., Mehl, M., Moore, R.: Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research* 30(1), 457–500 (2007)
11. Mairesse, F., Walker, M.: Automatic recognition of personality in conversation. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, pp. 85–88 (2006)
12. Mishne, G.: Experiments with mood classification in blog posts. *ACM SIGIR* (2005)
13. Pennebaker, J.W., Chung, C.K., Ireland, M., Gonzales, A., Booth, R.J.: *Linguistic inquiry and word count (liwc)*. Lawrence Erlbaum Associates, Mahwah (2007)
14. Shaer, D.W., D'Angelo, C., Chesler, N.C., Arastoopour, G.: Nephrotex: Teaching first year students how to think like engineers. In: *Laboratory Improvement (CCLI) PI Conference*, Washington D.C. (2011)
15. Stone, P., Dunphy, D., Smith, M., Ogilvie, D.: *The general inquirer: A computer approach to content analysis*. MIT Press (1966)
16. Taboada, M., Grieve, J.: Analyzing appraisal automatically. In: *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, CA, pp. 158–161 (2004)
17. Vaassen, F., Daelemans, W.: Emotion classification in a serious game for training communication skills. In: *Proceedings of the 20th Meeting of Computational Linguistics in the Netherlands*, Netherlands, pp. 155–168 (2010)
18. Wiebe, J., Wilson, T.: Learning to disambiguate potentially subjective expressions. In: *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pp. 112–118 (2002)

# Erratum: Neoclassical Compound Alignments from Comparable Corpora

Rima Harastani, Beatrice Daille, and Emmanuel Morin

University of Nantes  
LINA, 2 Rue de la Houssinière  
BP 92208, 44322 Nantes, France

{Rima.Harastani, Beatrice.Daille, Emmanuel.Morin}@univ-nantes.fr

---

A. Gelbukh (Ed.): CICLING 2012, Part II, LNCS 7182, pp. 72–82, 2012.  
© Springer-Verlag Berlin Heidelberg 2012

---

**DOI 10.1007/978-3-642-28601-8\_43**

The author missed to add the following acknowledgement to the paper:

**Acknowledgments.** The research leading to these results has received funding from the European Community's Seventh Framework Program (\*/FP7/2007-2013\*) under Grant Agreement no 248005.

---

The original online version for this chapter can be found at  
[http://dx.doi.org/10.1007/978-3-642-28601-8\\_7](http://dx.doi.org/10.1007/978-3-642-28601-8_7)

---

# 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  
Bandyopadhyay, 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, E1  
Dalbelo 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, Heshaam 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, Sofía 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, E1  
 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  
 Jacquin, 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  
 Makhlouta, Jad I-297  
 Malcuori, Marisa II-206  
 Marciničzuk, 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, E1  
 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  
 Reuß, 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, Ruvan 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
- Zanolí, 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